


Seiteninhalt als PDF


Oft möchte man den Inhalt einer Seite als PDF-Datei auf seinem Rechner haben. Wer nun keinen PDF-Drucker installiert hat, kann auf PDF-Dateien zugreifen, die vom Server bereitgestellt werden. Dazu muss man am  [DokuWiki](#)-Code einige kleine Änderungen vornehmen.


Ich habe mich hier für einen Weg entschieden, der aus dem Seiteninhalt eine Datei erzeugt. Dadurch, dass ich keine druckaufbereiteten Daten benutze, sieht das Ergebnis anders aus als wenn man es über den Drucken-Button des Webbrowsers ausgibt.

Zunächst habe ich mich bei der Umsetzung für [HTML2FPDF](#) entschieden. Es ist nicht so leistungsfähig wie andere Tools, aber es erfordert außer PHP keine weitere Software auf dem Webserver – ein Vorteil, wenn man für seinen Webauftritt nur Webspace hat und keinen eigenen Server.

Ich lasse das Tool eine PDF-Datei erzeugen und baue einen Link zum Herunterladen in die Fußzeile.

Anfangs habe ich in zwei Schritten den Inhalt der Seite abgefragt:

1. Zuerst muss ich dazu das verwendete Template betrachten: wo wird der Inhalt ausgegeben und wo soll der Link entstehen? Ich benutze ein leicht verändertes  ACH-Template. Dort – in der Datei `main.php` – findet sich eine Passage `<?php tpl_content() ?>`, die ich benutze, um mir den Inhalt der Seite, fertig aufbereitet, in eine Variable zu speichern: `<?php $myhtml = tpl_content(); ?>`.
2. Nun muss ich die Datei `inc/template.php` bearbeiten – sie enthält die Funktion `tpl_content()` –, damit ich wirklich den Inhalt zurück erhalte. Deshalb füge ich am Ende eine Zeile ein: `return $html_output;`.

Dann ist mir aufgefallen, dass PHP ja Methoden zum Output-Buffering bereitstellt . Damit muss ich nur noch den obigen Schritt 1. ausführen, dabei ändere ich den Text in `<?php ob_start(); tpl_content(); $myhtml = ob_get_contents(); ob_end_clean(); ?>`. Die Wirkung ist dieselbe: der Inhalt der Seite ist in der Variablen `$myhtml`.

Auf jeden Fall ist sichergestellt, dass ich den Inhalt auch wirklich erhalte. Nun muss er ausgegeben werden. Das kann nur unterhalb der Stelle im Template erfolgen, an der ich diesen Inhalt erfasst habe – in meinem Fall in der Fußzeile. Dort trage ich folgenden Code ein:

```
<?php
// Umwandlung Inhalt -> PDF-Datei
include(DOKU_INC . 'inc/html2fpdf/cPDF.php');
$destdir = DOKU_INC . 'tmp';
if (! @is_dir($destdir)) @mkdir($destdir);
$leaveTime = 60 * 60 * 24;
if (@is_dir($destdir)) {
    if ($dh = @opendir($destdir)) {
        while (($file = @readdir($dh)) !== false) {
            if (time() - @filemtime($file) > $leaveTime) @unlink($file);
        } // while (($file = @readdir($dh)) !== false)
        @closedir($dh);
    } // if ($dh = @opendir($destdir))
```

```

} // if (@is_dir($destdir))
$tempf1 = @tempnam($destdir, 'PDF_');
$tempf2 = strtolower($tempf1 . '.pdf');
$linkname = DOKU_BASE . 'tmp/' . basename($tempf2);
@link($tempf1, $tempf2);
@unlink($tempf1);
@chmod($tempf2, 0664);
$mycss = 'http://' . $_SERVER['SERVER_NAME'] . DOKU_BASE .
'lib/exe/css.php?print=1';
$myPDFsize = HTMLzuPDF($myhtml, $tempf2, $mycss);
?>

```

```

<a <?php echo $tgt; ?>
  href="<?php echo $linkname; ?>"
  title="<?php echo $lang['txt_domepdf']; ?>"
  "
    title="<?php echo $lang['txt_domepdf']; ?>" />
</a>

```

Kurze Erläuterung: zunächst Sorge ich per `include()` dafür, dass mir die Funktion `HTMLzuPDF()` zur Verfügung steht. Dann lege ich das Verzeichnis fest, in dem die PDF-Dateien abgelegt werden; falls es nicht besteht, wird es angelegt. Dann folgt ein Intermezzo: ich durchsuche das Verzeichnis nach Dateien, die älter als `$leaveTime` Sekunden sind – diese werden gelöscht. Dann lege ich eine temporäre Datei an; um damit arbeiten zu können, muss ich sie in Kleinbuchstaben umwandeln und die Endung `.pdf` anhängen. Der Zugriffsmodus wird gesetzt. Die letzte PHP-Codezeile bewirkt dann die Umwandlung in PDF.

Der HTML-Teil darunter zeigt ein Beispiel für das Erstellen eines Links auf diese Datei.

Tja, damit bin ich bald fertig. Fehlt nur noch das Zurechtbasteln eines kleinen Icons für die Schaltfläche unten – und der Eintrag von `$lang['txt_domepdf']` in die Datei `lang.php` – mindestens einmal im Unterverzeichnis `lang/en` des Templates, ich habe zusätzlich noch eine

deutsche Übersetzung eingetragen




Bekannte Probleme

- Bilder werden auf Grund der Art und Weise, wie die Links im Quelltext stehen, nicht ins PDF übernommen. Stattdessen erscheint der Platzhalter.
- relative Links (also innerhalb des DokuWiki) werden zwar als Link erkannt und von HTML2FPDF auch umgesetzt, jedoch als `file:` statt `http:`.
- Smileys werden nicht angezeigt (wie Bilder), jedoch wird an dieser Stelle die Zeile umbrochen.
- Die CSS-Fähigkeiten von HTML2FPDF sind ... ähm... ausbaufähig... z. B. wird `display: none;` ignoriert.

Neue Wege

Auf  [plugin:pdfdownload](#) wurde ich darauf hingewiesen, dass das Vorgehen unzweckmäßig ist. Daraufhin habe ich es umgestrickt und auf  [plugin:pdfex](#) beschrieben – deutsche Fassung in diesem

Wiki folgt 

From:

<http://wernerflamme.name/> - **Werners Wiki**

Permanent link:

<http://wernerflamme.name/doku.php?id=comp:pdfoutput>

Last update: **2008-08-23 18:22**

