EXT: WebkitPDF

Extension Key: webkitpdf

Language: en

Copyright 2009, Dev-Team Typoheads, <dev@typoheads.at>

This document is published under the Open Content License

available from http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3

- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

Introduction

What does it do?

Renders given pages into a PDF file using the Webkit engine. Extension caches the rendered PDF files for reuse.

Caching is done in a DB table. Clearing the cache clears temporary files older than 10 minutes.

You can set the threshold in the configuration section in extension manager.

IMPORTANT NOTICES

- A shell script is included to perform this task. The user must have the right permissions to run this script (Script has to be executable). You can move it to another location and specify the new path via TypoScript.

- This extension will NOT work on Windows systems!

- The used script apparently has some dependencies. Make sure you have libfreetype installed.

- If you experience problems (e.g. The PDF is not readable), try to run the script in a shell environment. For the exact call made by WebkitPDF you can use the debugScriptCall TypoScript setting.

Users manual

Administration

- Install via extension manager.
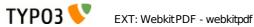- Create a new page and add a plugin 'Webkit PDF'.
- Use TypoScript to create a link for PDF generation
- Pass the page Ids to render in GET-parameter 'urls' (or custom if specified via TS)
- Parameter is an array of URLs

Example:

```
includeLibs.webkit = EXT:webkitpdf/res/user_webkitpdf.php
lib.pdf = USER
lib.pdf.userFunc = user_webkitpdf->user_getPDFLink
lib.pdf {
  pid = 23
  linkText = Save as PDF
}
```

The parameter linkText can be a String or any TypoScript object:

```
includeLibs.webkit = EXT:webkitpdf/res/user_webkitpdf.php
lib.pdf = USER
lib.pdf.userFunc = user_webkitpdf->user_getPDFLink
lib.pdf {
  pid = 23
  linkText = TEXT
  linkText.value = Now with TS object
}
```

Configuration

Reference

General Options

Property:	Data type:	Description:	Default:
filePrefix	String	Custom prefix for the name of the generated PDF file	
customScriptPath	String	If you copy the used shell script to another location for permission reasons, you can enter the new path to the executable.	
customTempOutputPath	String	Custom path to store the temp files	typo3temp
copyrightNotice	String	Enter a name of a person or organisation to be shown as a copyright notice. Copyright 2009 [name]	
additionalStylesheet	String	Enter path to an additional CSS file.	
customParameterName	String	Custom name of the GET parameter containing the page IDs	selected_pages
pageURLInHeader	Boolean (1/0)	Add the URL of the page to PDF header	0
debugScriptCall	Boolean (1/0)	Prints the call of the script to the screen for debugging purposes. You can copy/paste this call into the shell to see if the call itself throws errors.	0
staticFilename	String or cObj	Enter a filename for the downloadbale PDF. You can use a cObj to add dynamic things like timestamp to the filename. If you leave this option empty, the filename will be the same as the name of the temporary file.	
scriptParams	Array	Enter parameters for the script call directly. See section "Script parameters" for a full list. Enter parameters this way: scriptParams { margin-left = 35mm footer-line = } The script execution will look this way: wkhtmltopdf --margin-left 35mm --footer-line ...	

[plugin.tx_webkitpdf_pi1]

Script parameters

This is a list of available parameters. When using this in TypoScript omit the '--'.

Property:	Description:	Default:
--book	Set the options one would usualy set when printing a book.	
--cover <url>	Use html document as cover. It will be inserted before the toc with no headers and footers.	
--default-header	Add a default header, with the name of the page to the left, and the page number to the right, this is short for: --header-left='[webpage]' --header-right='[page]/[toPage]' --top 2cm --header-line.	
--disable-javascript	Do not allow webpages to run javascript.	
--dpi <dpi>	Change the dpi explicitly.	
--enable-plugins	Enable installed plugins (such as flash). Note: This doesn't really work. The developer's hope to integrate it in one of the next versions.	
--encoding <encoding>	Set the default text encoding, for input.	

Property:	Description:	Default:
--footer-center <text>	Centered footer text.	
--footer-font-name <name>	Set footer font name.	Arial
--footer-font-size <size>	Set footer font size.	11
--footer-left <text>	Left aligned footer text.	
--footer-line	Display line above the footer.	
--footer-right <text>	Right aligned footer text.	
--grayscale	PDF will be generated in grayscale.	
--header-center <text>	Centered header text.	
--header-font-name <name>	Set header font name.	Arial
--header-font-size <size>	Set header font size.	11
--header-left <text>	Left aligned header text.	
--header-line	Display line below the header.	
--header-right <text>	Right aligned header text.	
--lowquality	Generates lower quality pdf/ps. Useful to shrink the result document space.	
--margin-bottom <unitread>	Set the page bottom margin.	10mm
--margin-left <unitread>	Set the page left margin.	10mm
--margin-right <unitread>	Set the page right margin.	10mm
--margin-top <unitread>	Set the page top margin.	10mm
--no-background	Do not print background.	
--orientation <orientation>	Set orientation to Landscape or Portrait.	
--outline	Put an outline into the pdf.	
--outline-depth <level>	Set the depth of the outline.	4
--page-offset <offset>	Set the starting page number.	1
--page-size <size>	Set pape size to: A4, Letter, ect..	
--password <password>	HTTP Authentication password.	
--print-media-type	Use print media-type instead of screen.	
--proxy <proxy>	Use a proxy.	
--redirect-delay <msec>	Wait some miliseconds for js-redirects,	200
--toc	Insert a table of content in the beginning of the document.	
--toc-depth <level>	Set the depth of the toc.	3
--toc-font-name <name>	Set the font used for the toc.	Arial
--toc-header-fs <size>	The font size of the toc header	15
--toc-header-text <text>	The header text of the toc.	Table Of Contents
--toc-l1-font-size <size>	Set the font size on level 1 of the toc.	12
--toc-l1-indentation <num>	Set indentation on level 1 of the toc.	0
--toc-l2-font-size <size>	Set the font size on level 2 of the toc.	10
--toc-l2-indentation <num>	Set indentation on level 2 of the toc.	20
--toc-l3-font-size <size>	Set the font size on level 3 of the toc.	8
--toc-l3-indentation <num>	Set indentation on level 3 of the toc.	40
--toc-no-dots	Do not use dots, in the toc.	
--user-style-sheet <url>	Specify a user style sheet, to load with every page.	
--username <username>	HTTP Authentication username.	

Note: The following parameters will override some TypoScript settings:

Parameter	Overwrites this TypoScript setting
user-style-sheet	additionalStylesheet
footer-left	copyrightNotice
header-center	pageURLInHeader

Additional script information

Proxy:

  By default proxy information will be read from the environment

  variables: proxy, all_proxy and http_proxy, proxy options can

  also by specified with the -p switch

  <type> := "http://" | "socks5://"

  <userinfo> := <username> (":" <password>)? "@"

  <proxy> := "None" | <type>? <userinfo>? <host> (":" <port>)?

Header and footer text:

In a header or footer text the following variables can be used

 * [page]   Replaced by the number of the pages currently beeing printed

 * [fromPage]   Replaced by the number of the first page to be printed

 * [toPage]   Replaced by the number of the last page to be printed

 * [webpage]   Replaced by the url of the page beeing printed

 * [section]   Replaced by the name of the current section

 * [subsection] Replaced by the name of the current subsection

Known problems

– Non currently

To-Do list

– Waiting for Flash support. This would be really awesome!

ChangeLog

– 1.0.0: initial release.

– 1.0.1:

    – Fixed usage with multiple GET parameters

    – Added userFunc to generate PDF link

    – New TS option: pageURLInHeader

    – Parameter linkText for userFunc can be any TS object now

– 1.1.0

    – Extension caches the rendered PDF files for reuse. Clear cache in backend.

    – Caching is done in a DB table. Clearing the cache clears temporary files older than 10 minutes.