EXT: tollwerk squeezr

Extension Key: squeezr
Language: en
Version: 0.9.0
Keywords: squeezr, device-aware adaptive images, server side css3 media queries, bandwidth, forAdministrators, forDevelopers, forIntermediates, forAdvanced

In case you are reading this manual online at the TYPO3 website, we strongly recommend that you also visit the TYPO3 Extensions & Manuals page respectively the page about the tollwerk squeezr TYPO3-Extension at our own website. We provide a PDF version of this manual there, which probably renders more nicely than the online version on typo3.org. (Sorry for our website currently being available in German language only. However, the PDF extension manuals are in English of course.)

Table of Contents

Introduction

What does it do?

This extension provides easy means to integrate **squeezr** into your TYPO3 website. squeezr can help you reducing the size of image and CSS files delivered to your visitors, thus saving them time, bandwidth and nerves.

squeezr currently features two independent engines:

- **Image engine**: squeezr recognizes your visitor's screen size, downscales your images appropriately, caches and finally delivers them to your visitor.
- **CSS engine**: squeezr analyses your CSS files and strips out irrelevant media query sections – on the server side.

Extension features:

- **Easy installation**: Most of the necessary steps are performed automatically or by a single mouse click
- Configuration solely from inside the TYPO3 backend via the extension configuration and the constant editor
- 1-click **squeezr cache cleaning** from within the backend (useful when image or CSS files are changed)

In order to use the extension, you will have to meet these requirements on your server:

- Apache Webserver 2.2+ (with mod_rewrite)
- PHP 5.3+
- GD (mostly standard with PHP)

The extension was written for TYPO3 version 6+ (no support for 4.x!).

For detailed information about squeezr and it's features please visit the squeezr website at http://squeezr.it. Furthermore, to keep up to date you might want to follow the @squeezr Twitter account or visit the GitHub repository to report any issues.

Screenshots

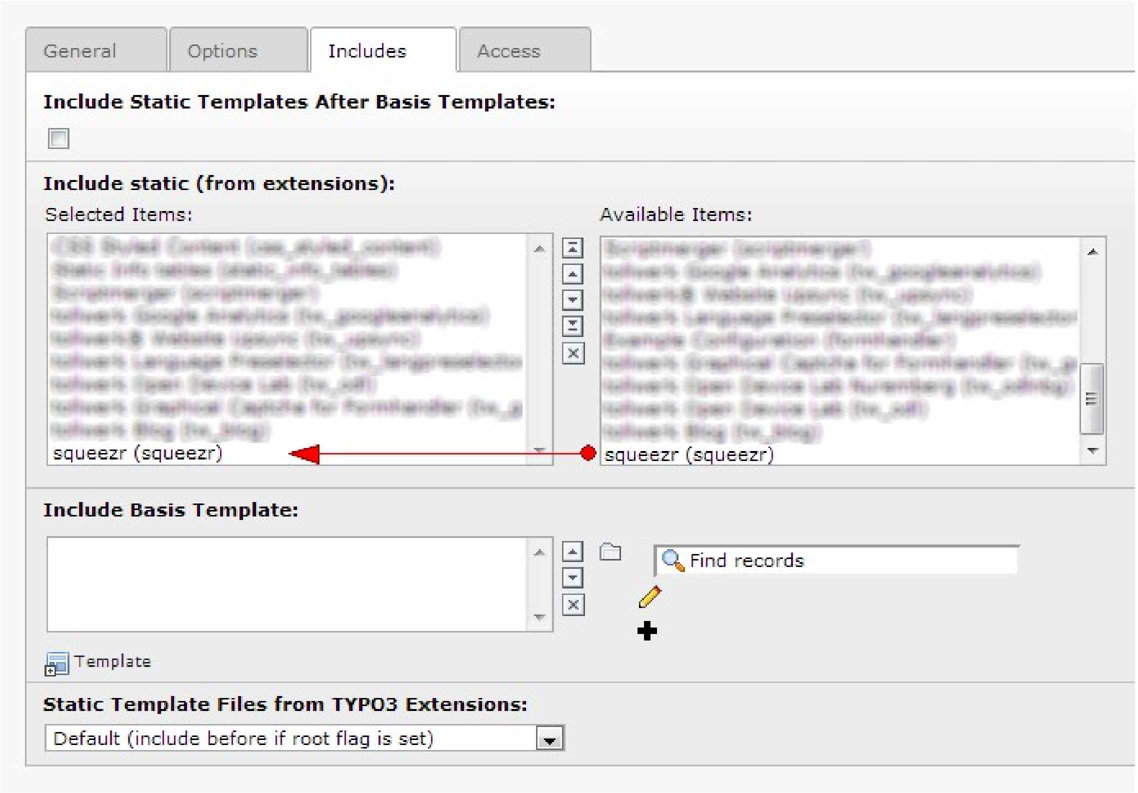
The extension doesn't produce any visible output, so there is nothing to take a screenshot of. Instead, it supports you in reducing the amount of data delivered to your visitors by shrinking your images and CSS files down to the limitations of the visitor's device. Having said this, there might be a visual impact in your specific context though.

Installation

To install the extension simply download it from the TYPO3 extension repository and enable it in the Extension Manager. There are some settings you will want to configure in the extension manager.

Static TypoScript

Include the extension's static TypoScript into the root template of your site.



Include Static Templates After Basis Templates:
☐

Include static (from extensions):
Selected Items: squeezr (squeezr)
Available Items: squeezr (squeezr)

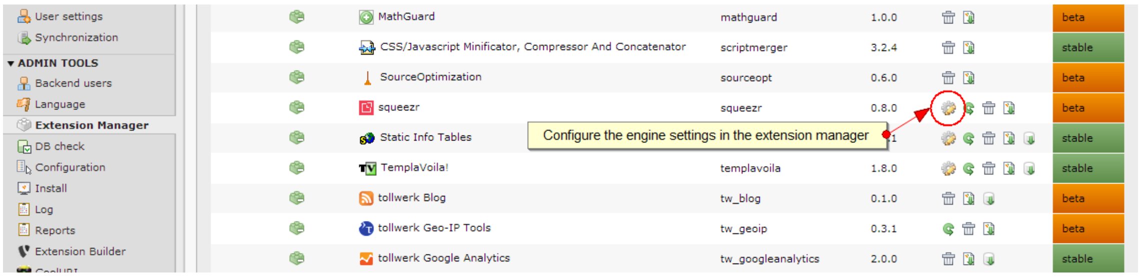
Include Basis Template:

Static Template Files from TYPO3 Extensions:
Default (include before if root flag is set)

Configuration

Extension configuration

Start by configuring the engine settings in the extension manager:



MathGuard	mathguard	1.0.0	beta
CSS/Javascript Minificator, Compressor And Concatenator	scriptmerger	3.2.4	stable
SourceOptimization	sourceopt	0.6.0	beta
squeezr	squeezr	0.8.0	beta
Static Info Tables		1	stable
TemplaVoila!	templavoila	1.8.0	stable
tollwerk Blog	tw_blog	0.1.0	beta
tollwerk Geo-IP Tools	tw_geoip	0.3.1	beta
tollwerk Google Analytics	tw_googleanalytics	2.0.0	stable

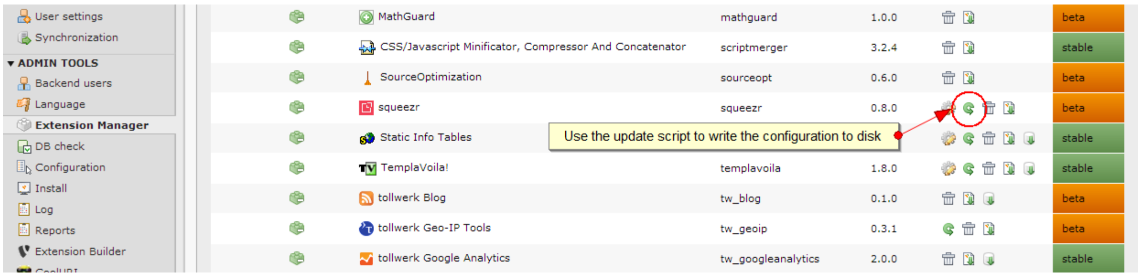
Configure the engine settings in the extension manager

Most of the available options correspond to a squeezr configuration constant, which you can also find on the squeezr Website:

Property:	Data type:	Description:	Default:
common.lifetime	integer	By default, squeezr instructs client browsers to cache any file for one week (604800 seconds). You can change this by providing another **expiration period** here. squeezr constant: SQUEEZR_CACHE_LIFETIME	604800
images.images	boolean	Set this to FALSE to **disable the image engine temporarily**. To disable it permanently (and prevent PHP from being involved), just remove or comment out the corresponding image rewrite rules from the main .htaccess file. squeezr constant: SQUEEZR_IMAGE	1
images.quality	integer	Control the **quality of JPEG** images with this setting. You may specifiy an integer between 1 and 100 (reasonable values are 60 - 80; defaults to 80). squeezr constant: SQUEEZR_IMAGE_JPEG_QUALITY	80
images.sharpen	boolean	When images are downscaled, they tend to become somewhat blurry. This is why they get **slightly sharpened** by default. Set this option to FALSE to disable the sharpening. squeezr constant: SQUEEZR_IMAGE_SHARPEN	1
images.undersized	boolean	Sometimes images don't need to be downscaled for a specific breakpoint as they are already small enough by default. Nevertheless, squeezr creates breakpoint specific symlinks in such cases in order to speed up subsequent requests for the same files. If your system doesn't support symlinks for some reason (e.g. due to PHP restrictions), you can still let squeezr **create real copies of those files**. Set this option to TRUE to enable this behaviour, but please be aware of the potentially higher disk space requirements. squeezr constant: SQUEEZR_IMAGE_COPY_UNDERSIZED	0
css.css	boolean	Set this to FALSE to **disable the css engine temporarily**. As with the image engine, to disable it permanently (and prevent PHP from being involved), just remove or comment out the corresponding CSS rewrite rules from the main .htaccess file. squeezr constant: SQUEEZR_CSS	1

Property:	Data type:	Description:	Default:
css.minification	string	squeezr offers the capability to optionally **apply CSS minification**. Third party libraries such as Minify may be used as minification providers (in fact, at the time of this writing, Minify is the only supported minification provider yet, but others might be implemented easily due to the modular architecture of squeezr). This setting defaults to the string “minify”, indicating that minification will be performed using Minify as provider. Set this to “---” to disable minifaction altogether. squeezr constant: SQUEEZR_CSS_MINIFICATION_PROVIDER	Minify

After having modified any of these extension settings, it **is essential that the changes are written to disk**. So any time you alter some values here – also after the initial configuration – be sure to **use the extension's update script** in order to write out the new configuration. **squeezr will not work otherwise!**



MathGuard	mathguard	1.0.0	beta
CSS/Javascript Minificator, Compressor And Concatenator	scriptmerger	3.2.4	stable
SourceOptimization	sourceopt	0.6.0	beta
squeezr	squeezr	0.8.0	beta
Static Info Tables			stable
TemplaVoila!	templavoila	1.8.0	stable
tollwerk Blog	tw_blog	0.1.0	beta
tollwerk Geo-IP Tools	tw_geoip	0.3.1	beta
tollwerk Google Analytics	tw_googleanalytics	2.0.0	stable

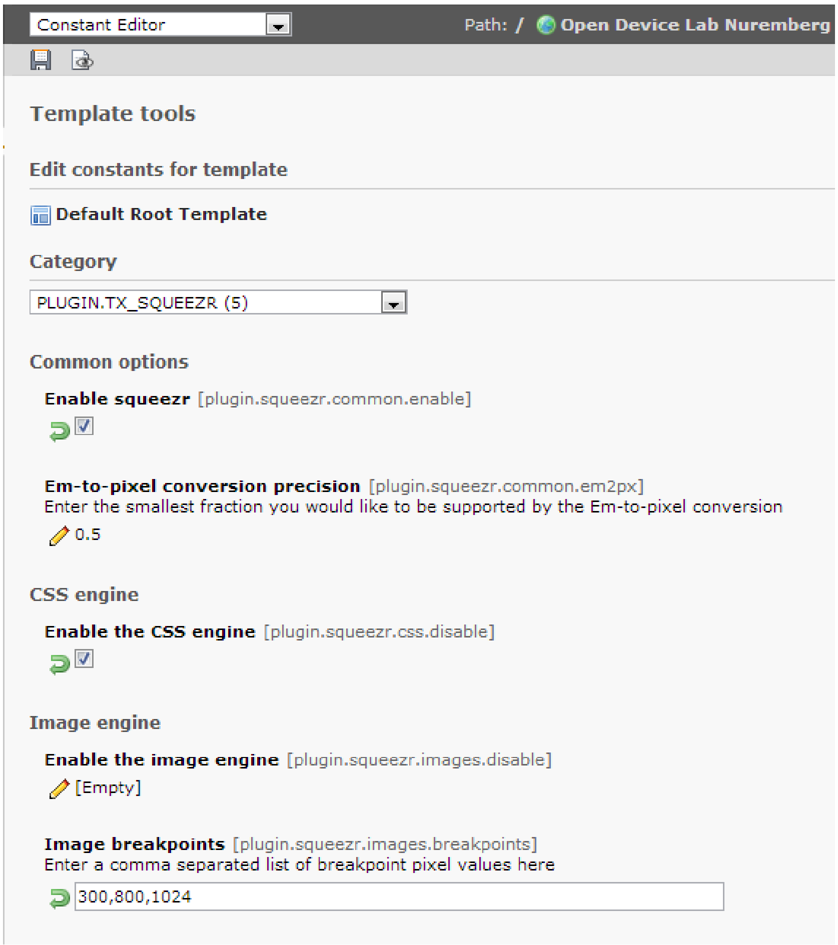
Use the update script to write the configuration to disk

Constants

There are a couple of constants controlling the frontend behaviour of squeezr. Edit them using the constant editor:

Property:	Data type:	Description:	Default:
plugin.squeezr.common.enable	boolean	You need to check this option in order to **make squeezr work**. If you don't, squeezr's JavaScript will not be integrated into your frontend output and none of the necessary cookies will be sent to the server. Use this as a general on/off switch for squeezr.	0
plugin.squeezr.common.em2px	float	When the squeezr script measures the **em-to-pixel ratio** of a visitor's screen, it does so by employing a certain precision, which defaults to a value of 0.5. Therefore, squeezr measures in steps like 10em, 10.5em and 11em by default. You can alter this precision by specifying an arbitrary positive floating point value like 0.01, which means that squeezr will be as precise as 10.00em, 10.01em and 10.02em.	0.5
plugin.squeezr.css.disable	boolean	If set, the **CSS engine is disabled** altogether. You don't need to take care of the em-to-pixel ratio in this case either.	0
plugin.squeezr.images.disable	boolean	If set, the **image engine is disabled** altogether. You don't need to specify the image breakpoints in this case either.	0
plugin.squeezr.images.breakpoints	string	If you want to use the image engine, this attribute has to be present and must carry a **comma separated list of breakpoints** to be used. The breakpoints have to be expressed in pixels. If this option is empty, the image engine will stay inactive.	

[tsref:plugin.squeezr]



Template tools

Edit constants for template

Default Root Template

Category

PLUGIN.TX_SQUEEZR (5)

Common options

Enable squeezr [plugin.squeezr.common.enable]
☑

Em-to-pixel conversion precision [plugin.squeezr.common.em2px]
Enter the smallest fraction you would like to be supported by the Em-to-pixel conversion
0.5

CSS engine

Enable the CSS engine [plugin.squeezr.css.disable]
☑

Image engine

Enable the image engine [plugin.squeezr.images.disable]
[Empty]

Image breakpoints [plugin.squeezr.images.breakpoints]
Enter a comma separated list of breakpoint pixel values here
300,800,1024

Setup

TypoScript

This extension doesn't have any TypoScript options that need to be configured. As soon as you have installed and configured the extension, included the static TypoScript into your root template and configured the available constants, squeezr is almost operational. Just don't forget to incorporate the necessary Apache rewrite rules as a last step.

Apache rewrite rules

You will have to install some Apache rewrite rules in order to make squeezr work. Depending on whether you already have a **.htaccess file** on the top level of your website, you may use the default one shipping with squeezr, or you will have to manually craft the necessary rules into your existing one.

Start a new .htaccess file

In case you don't have a top level .htaccess file yet, just copy squeezr's default one from **Resources/Private/Squeezr/.htaccess** (within the extension's installation directory) to your website's root directory. As the extensions ships with a vanilla distribution of squeezr, there are some little tweaks that have to be made in order to make squeezr work within the TYPO3 context. Open your new .htaccess file with a text editor and change the line

```
RewriteRule ^(.+)(\.(?:jpe?g|gif|png))$ squeezr/cache/$1-%1$2 [NC,E=BREAKPOINT:%1,L]
```

to

```
RewriteRule ^(.+)(\.(?:jpe?g|gif|png))$ typo3temp/squeezr/cache/$1-%1$2 [NC,E=BREAKPOINT:
%1,L]
```

as well as

```
RewriteRule ^(.+)\.css$ squeezr/cache/$1-%1.css [NC,E=BREAKPOINT:%1,L]
```

to

```
RewriteRule ^(.+)\.css$ typo3temp/squeezr/cache/$1-%1.css [NC,E=BREAKPOINT:%1,L]
```

Finally, you will want to exclude the TYPO3 backend from being affected by squeezr. Therefore, put another rewrite condition into **both the rewrite rules for images and CSS files**, just before the lines starting with “RewriteRule”:

```
RewriteCond %{HTTP_REFERER} !/typo3/
```

For example, the complete CSS rewrite rule should look like this now:

```
RewriteCond %{REQUEST_FILENAME} -f
RewriteCond %{ENV:REDIRECT_BREAKPOINT} !\d+px
RewriteCond %{QUERY_STRING} !^([^&]*&)*squeezr=(0|false|no)
RewriteCond %{HTTP_COOKIE} squeezr.css=(\d+x\d+@\d+(?:\.\d+)?) [NC]
RewriteCond %{HTTP_REFERER} !/typo3/
RewriteRule ^(.+)\.css$ typo3temp/squeezr/cache/$1-%1.css [NC,E=BREAKPOINT:%1,L]
```

That's it. squeezr should be able to run with this setup now.

Incorporating the rewrite rules into an existing .htaccess file

In case you already have a .htaccess file for your site, please open it using a text editor. Find out if there's already a rewrite rule section (which is likely if you're using e.g. cooluri or RealURL). It might e.g. look like this:

```
<IfModule mod_rewrite.c>
      RewriteEngine On
      RewriteBase /

      # HERE COME SOME REWRITE RULES
      # ...
</IfModule>
```

If there's no such section yet, then you might use the above as a template (or better get it from squeezr's .htaccess file). Then copy the rewrite rules from squeezr's default .htaccess file (**Resources/Private/Squeezr/.htaccess**) into this section:

```
############################################################################################
# REDIRECT ANY DIRECT IMAGE REQUEST TO A CACHED VERSION
#
# You may add files or directories that shouldn't be touched by squeezr like this:
#
#    RewriteCond %{REQUEST_URI} !path/to/some/file-or-directory
#
# Please refer to the mod_rewrite documentation at
# http://httpd.apache.org/docs/2.0/mod/mod_rewrite.html for further possibilities and
# instructions.
#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    RewriteCond %{REQUEST_FILENAME} -f
    RewriteCond %{ENV:REDIRECT_BREAKPOINT} !\d+px
    RewriteCond %{QUERY_STRING} !^([^&]*&)*squeezr=(0|false|no)
    RewriteCond %{HTTP_COOKIE} squeezr.images=(\d+px) [NC]
    RewriteRule ^(.+)(\.(?:jpe?g|gif|png))$ squeezr/cache/$1-%1$2 [NC,E=BREAKPOINT:%1,L]
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Please make sure that you set this path     ^^^     to the squeezr root directory that is
# also specified for the SQUEEZR_ROOT constant in the common engine configuration
# (SQUEEZR_ROOT/conf/common.php). If you apply the default setup for squeezr (i.e. put
# everything into a directory named "squeezr" under your website's document root),
# then you shouldn't have to change anything.
############################################################################################


############################################################################################
# REDIRECT ANY DIRECT CSS REQUEST TO A CACHED VERSION
#
```

```
# See above for some hints about excluding files or directories from the squeezr processing
#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    RewriteCond %{REQUEST_FILENAME} -f
    RewriteCond %{ENV:REDIRECT_BREAKPOINT} !\d+px
    RewriteCond %{QUERY_STRING} !^([^&]*&)*squeezr=(0|false|no)
    RewriteCond %{HTTP_COOKIE} squeezr.css=(\d+x\d+@\d+(?:\.\d+)?) [NC]
    RewriteRule ^(.+)\.css$ squeezr/cache/$1-%1.css [NC,E=BREAKPOINT:%1,L]
# ~~~~~~~~~~~~~~~~~~~~~~~~~~       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# See above for hints on      ^^^     this path.
############################################################################################
```

Afterwards, make sure to apply the same patches as described above in the section about when not having an exisiting .htaccess file (essentially, modifiy the rewrite path to be “**typo3temp/squeezr**” instead of just “**squeezr**” and adding a rewrite condition for excluding the TYPO3 backend).

Also make sure that your .htaccess file contains

```
Options +FollowSymlinks
```

somewhere. This is necessary for Apache to follow symlinks, which is an essential requirement for squeezr.

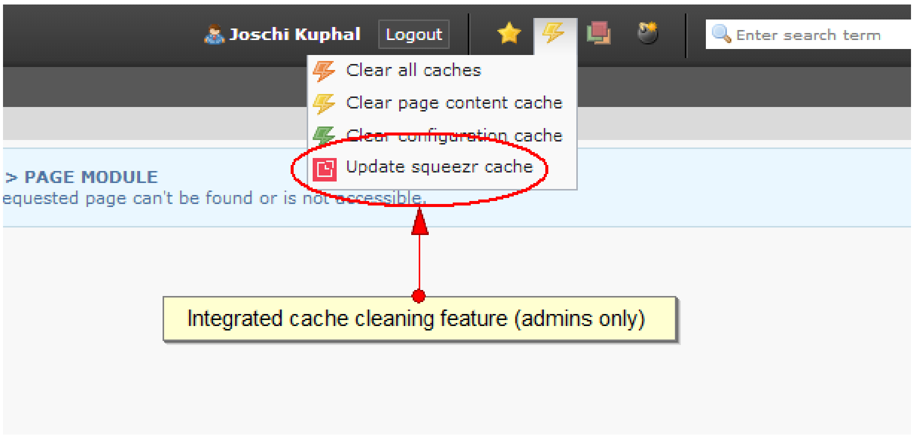
There is some more useful stuff at the end of squeezr's default .htaccess file. These parts are not mandatory, but you may want to use them as well as they could lead to a significantly improved performance of your website.

Please see the official mod_rewrite documentation on further instructions on rewrite rules in general.

Features

Cache cleaning

When squeezr downscales images or shrinks CSS files by stripping out irrelevant media query sections, it generates file variants and caches them to disk. However, when you alter the original images or CSS files afterwards, the generated variants will likely get outdated. In order to keep file requests for your visitors as efficient as possible, **squeezr does not impose an automatic cache validity check at request time**. Instead, it's the editor's (that is: your) obligation to explicitly clean the cache each time you want some changes to become effective. Therefore, the extension introduces a new option as part of the cache operations dropdown which you can use for manually cleaning / regenerating squeezr's cache. Both cached images and CSS files are affected and will be deleted if the original file has a newer modification date.



equested page can't be found or is not accessible.

Integrated cache cleaning feature (admins only)

Known problems

There are no known problems specific to the TYPO3 extension. However, squeezr itself has some aspects you might want to know. Please report any problems to the author.

For detailed information about squeezr and it's features please visit the squeezr website at http://squeezr.it. Furthermore, to keep up to date you might want to follow the @squeezr Twitter account or visit the GitHub repository to report any issues.

To-Do list

Currently there are no particular plans for the future except keeping pace with the improvements of squeezr itself. If there's something you would like to see in this extension then please let us know!

ChangeLog

Version:	Changes:
0.9.0	Initial public release to the TYPO3 Extension Repository, implementing the first public beta of squeezr