

TYPO3 CMS 6.0

Die Neuerungen

27.11.2012



TYP03

Patrick Lobacher (GF typovision GmbH)

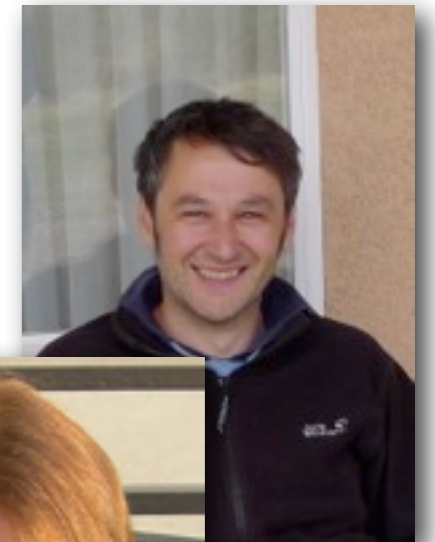
TYPO3 CMS 6.0



TYPO3 CMS 6.0

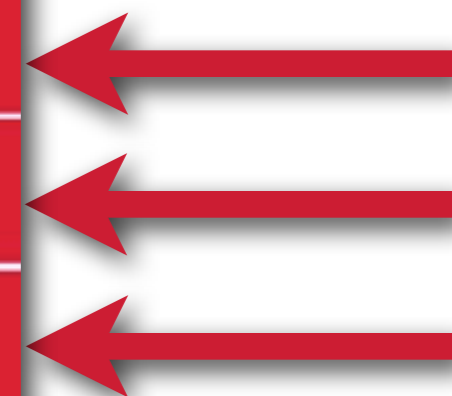
„Back to the Future“

- Veröffentlicht:** TYPO3 6.0.0 / 27.11.2012
- Release Team:** Helmut Hummel (Release Manager)
Susanne Moog
Christian Kuhn
- Hauptziele:** Mitarbeit fördern und unterstützen
Code-Qualität verbessern
File Abstraction Layer integrieren
Benutzerfreundlichkeit steigern



TYPO3 CMS - Versionen

	4.5 LTS	4.6	4.7	6.0
Status	stabil Long Term Support	stabil (alt)	stabil	stabil
Veröffentlicht am	26.01.2011	25.10.2011	25.04.2012	27.11.2012
Aktuelle Version	4.5.22 12.11.2012	4.6.15 12.11.2012	4.7.7 12.11.2012	6.0.0 27.11.2012
PHP-Version	5.2.X - 5.3.X	5.3.X	5.3.X	5.3.7 - 5.4.X
MySQL-Version	5.0.X - 5.1.X	5.0.X - 5.1.X	5.0.X - 5.1.X	5.1.X - 5.5.X
Laufzeitende	April 2014	April 2013	Oktober 2013	April 2014



- **File Abstraction Layer (6)**
- **Extension Manager (27)**
- **LocalConfiguration (32)**
- **Allgemeines (37)**
- **RTE (54)**
- **TypoScript & TSconfig (56)**
- **Namespaces (64)**
- **Logging API (70)**
- **API (74)**
- **Interna (80)**
- **Extbase & Fluid (103)**
- **Hooks (111)**
- **Deprecated Functions (116)**
- **Upgrade (119)**
- **Quellen (122)**
- **TYPO3 Branding (125)**

FILE ABSTRACTION LAYER

FAL

- **FAL: File Abstraction Layer**
⇒ Datei-Handling auf tiefer Ebene (über DB)
- Wurde 2010 auf den Developer Days in Elmshorn gestartet
- Im Wesentlichen entwickelt von Benjamin Mack, Steffen Ritter, Ingmar Schlecht und Andreas Wolf
- Ist/War Bestandteil des BLE-Projekts 2011
- **Ziele:**
 - Dateien nicht mehr über Dateipfad referenzieren
 - Datenspeicherung unabhängig von fileadmin/ ermöglichen
 - Versionierung von Dateien
 - Metadatenverwaltung (DAM light)
 - Verbessertes User-Interface

- **Keine Kopien - nur Referenzen**
- **„Storages“ sind Speicherorte (Remote oder lokal)**
- **„Driver“ agieren als Art des Speichers (WebDAV, Cloud, ...)**
- **Bilder als IRRE-Datensätze inkl. Metadaten**

Benutzungs-Schicht (Usage Layer)
Dateireferenz (FileReference)

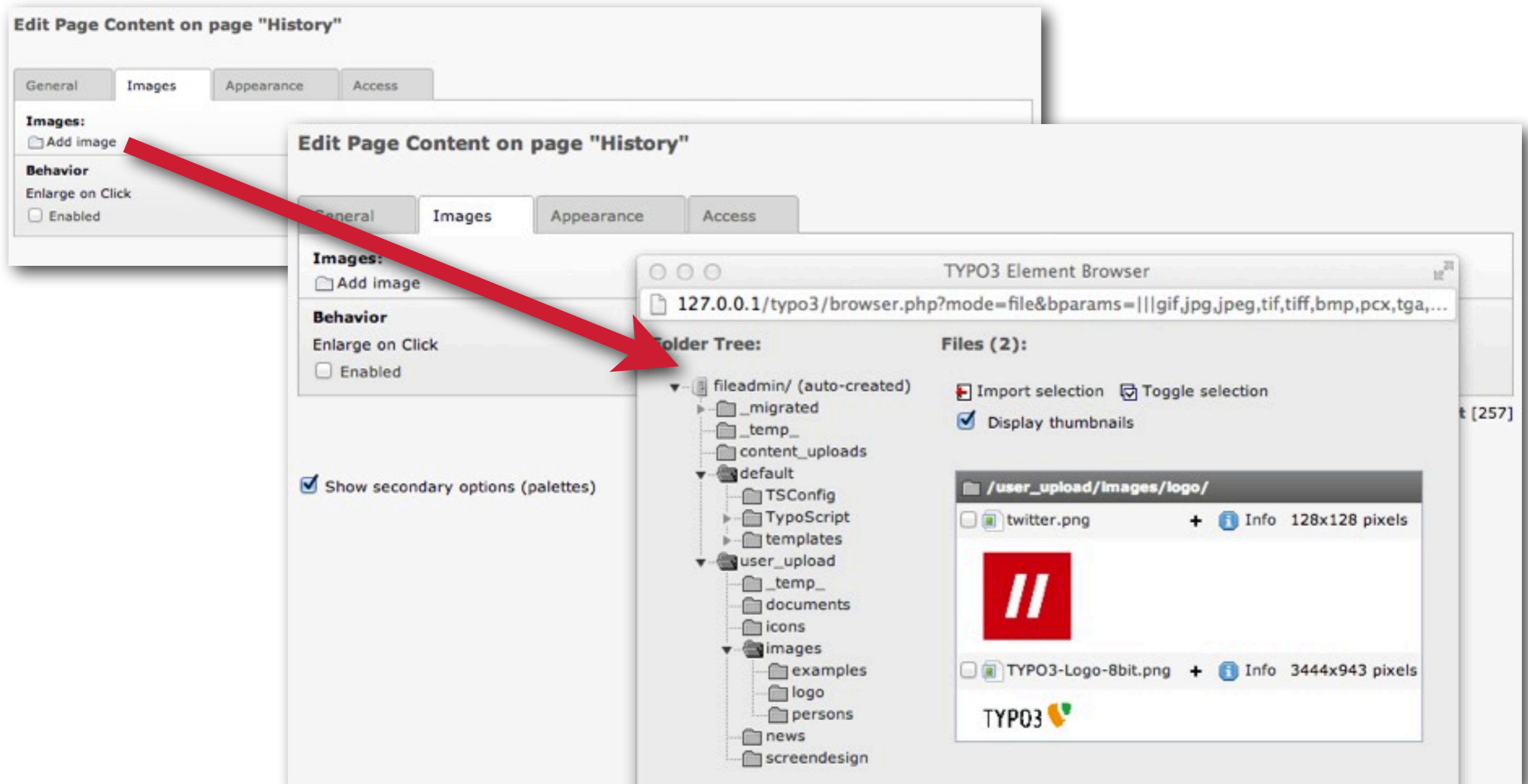
Speicher-Schicht (Storage Layer)
Datei/Verzeichnis Speicher (File/Folder Storage)

Treiber-Schicht (Driver Layer)
Treiber (Driver)

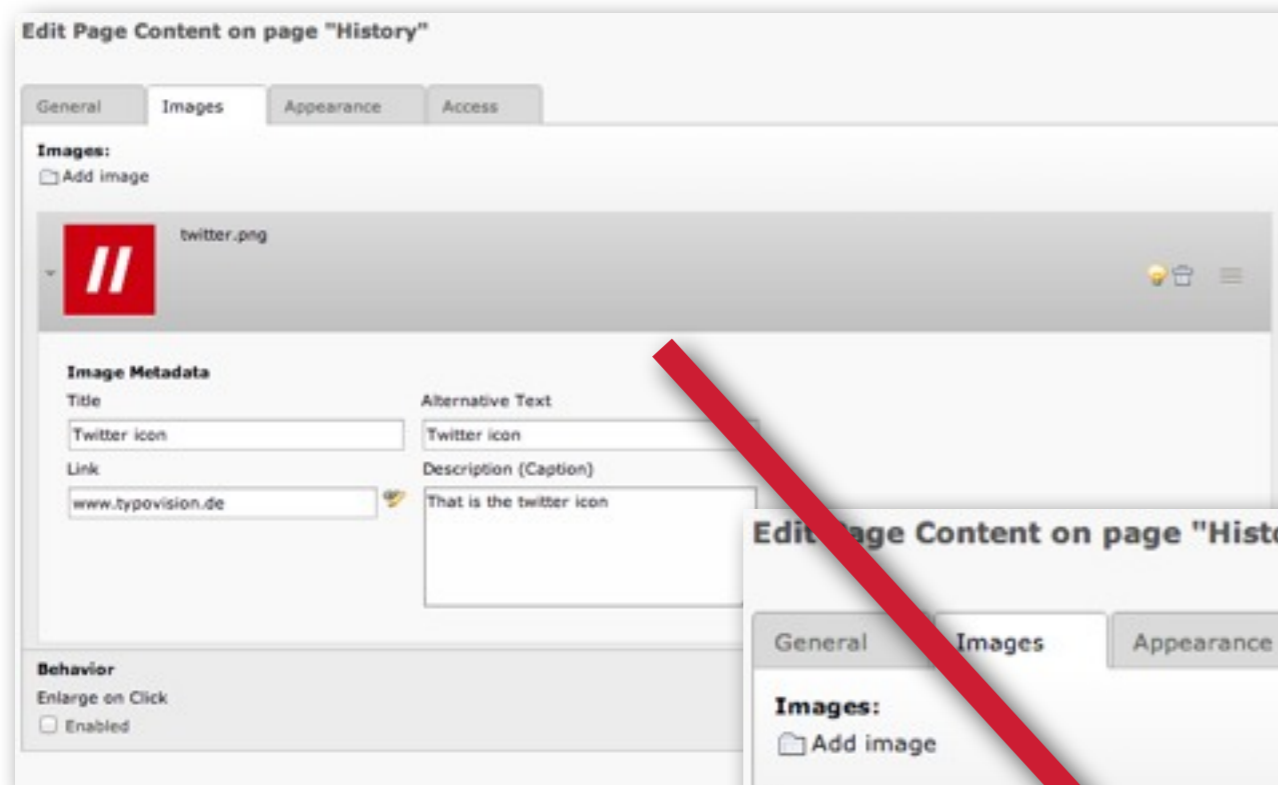
Mount-Point

low-level interface
Systemcalls

- **Integration in CE Text/Bild**



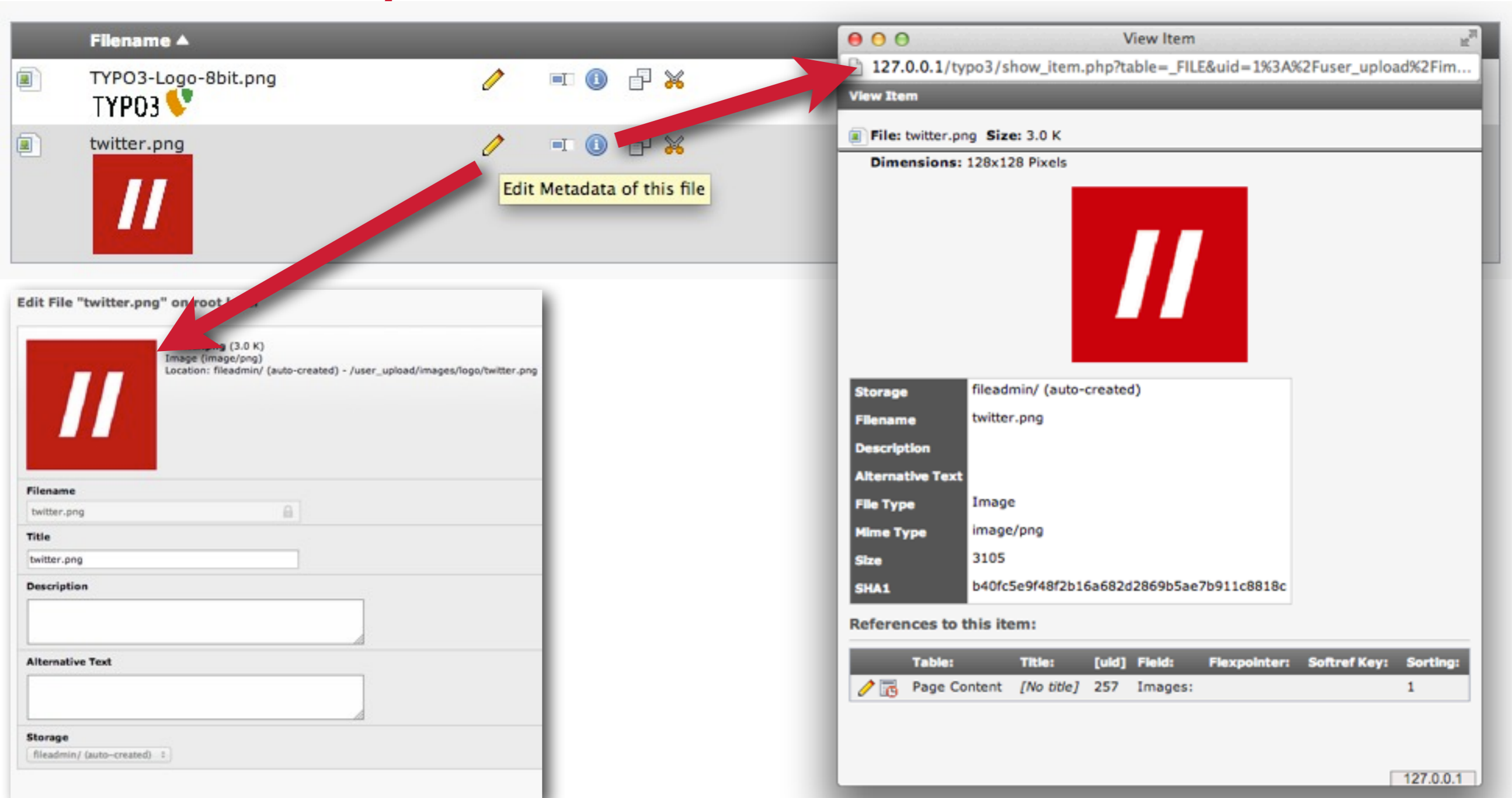
- **Integration in CE Text/Bild**



- Anlegen von Medien per IRRE
- Drag und Drop
- Meta-Daten Eingabe





- **Fileadmin Integration - Edit Metadata**



The image shows a workflow for editing file metadata in TYPO3 Fileadmin. It consists of three main parts:

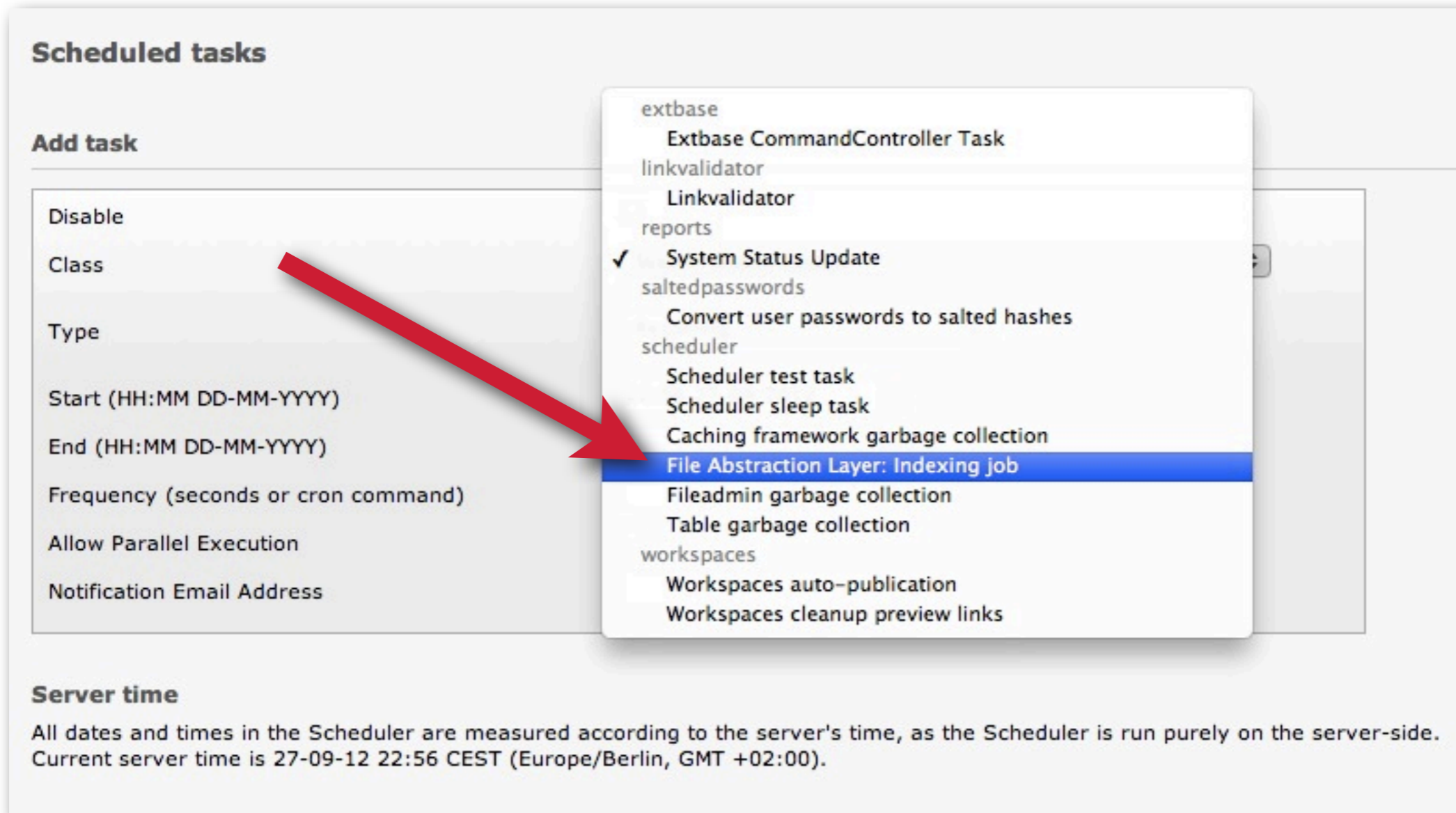
- File Listing:** A table showing files like 'TYPO3-Logo-8bit.png' and 'twitter.png'. A red arrow points from the 'twitter.png' file to the 'Edit Metadata of this file' button.
- Edit File Form:** A form titled 'Edit File "twitter.png" on root'. It contains fields for 'Filename', 'Title', 'Description', 'Alternative Text', and 'Storage'. A red arrow points from the 'Edit Metadata of this file' button to this form.
- View Item:** A detailed view of the file 'twitter.png'. It shows the file icon, dimensions (128x128 Pixels), and a metadata table.

Storage	fileadmin/ (auto-created)
Filename	twitter.png
Description	
Alternative Text	
File Type	Image
Mime Type	image/png
Size	3105
SHA1	b40fc5e9f48f2b16a682d2869b5ae7b911c8818c

Table:	Title:	[uid]	Field:	Flexpointer:	Softref Key:	Sorting:
 	Page Content	[No title]	257	Images:		1

- **FAL Driver (Treiber) spezifizieren die Art des Speicherns von Dateien**
 - Lokal: Ist bereits in TYPO3 eingebaut
 - WebDAV:
http://forge.typo3.org/projects/show/extension-fal_webdav
 - Amazon S3
http://forge.typo3.org/projects/show/extension-fal_amazons3

- **FAL Scheduler Job zum automatischen Indexieren von Dateien**



Scheduled tasks

Add task

Disable

Class

Type

Start (HH:MM DD-MM-YYYY)

End (HH:MM DD-MM-YYYY)

Frequency (seconds or cron command)

Allow Parallel Execution

Notification Email Address

Server time

All dates and times in the Scheduler are measured according to the server's time, as the Scheduler is run purely on the server-side. Current server time is 27-09-12 22:56 CEST (Europe/Berlin, GMT +02:00).

extbase
Extbase CommandController Task

linkvalidator
Linkvalidator

reports

✓ System Status Update

saltedpasswords
Convert user passwords to salted hashes

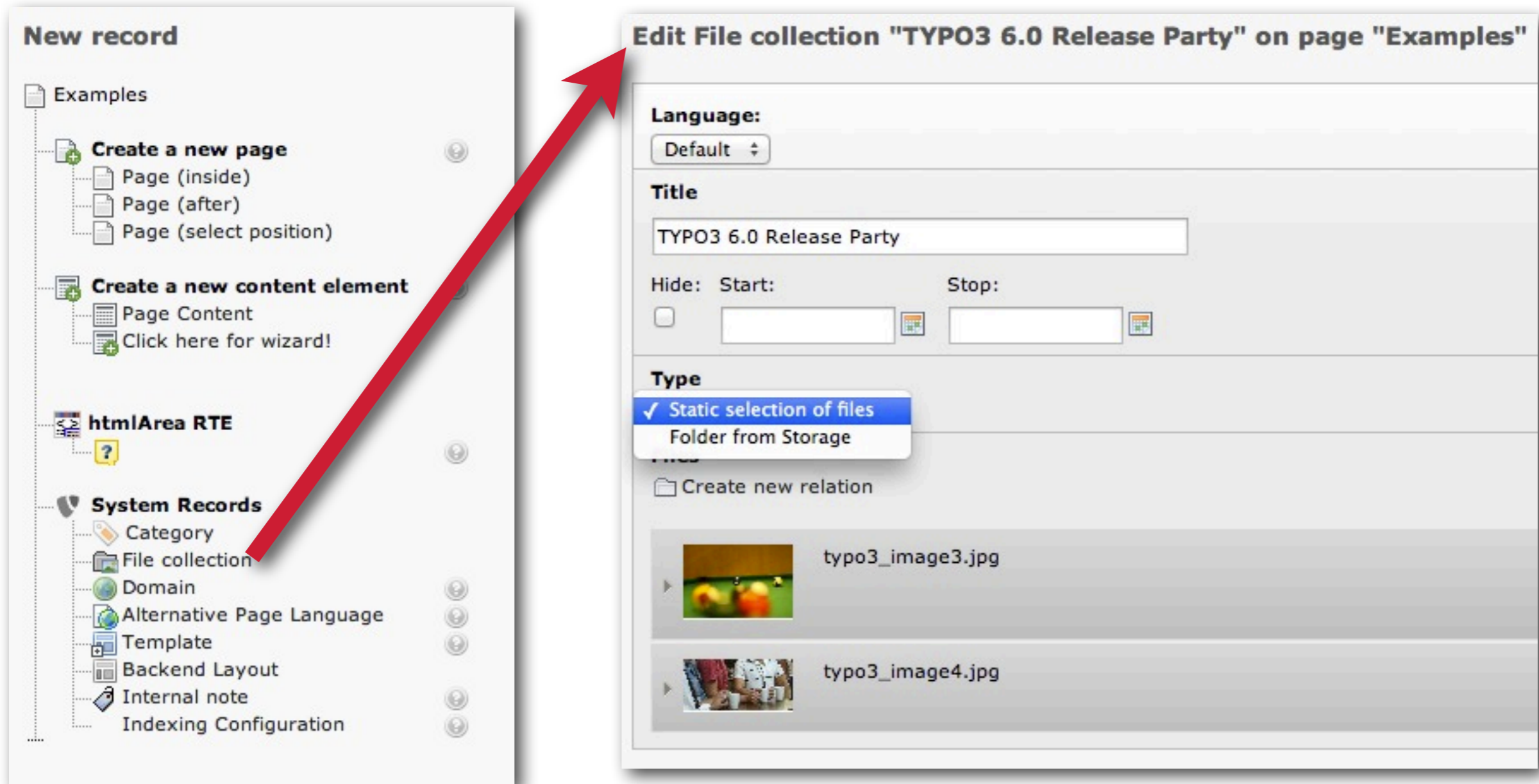
scheduler
Scheduler test task
Scheduler sleep task
Caching framework garbage collection

File Abstraction Layer: Indexing job

Fileadmin garbage collection
Table garbage collection

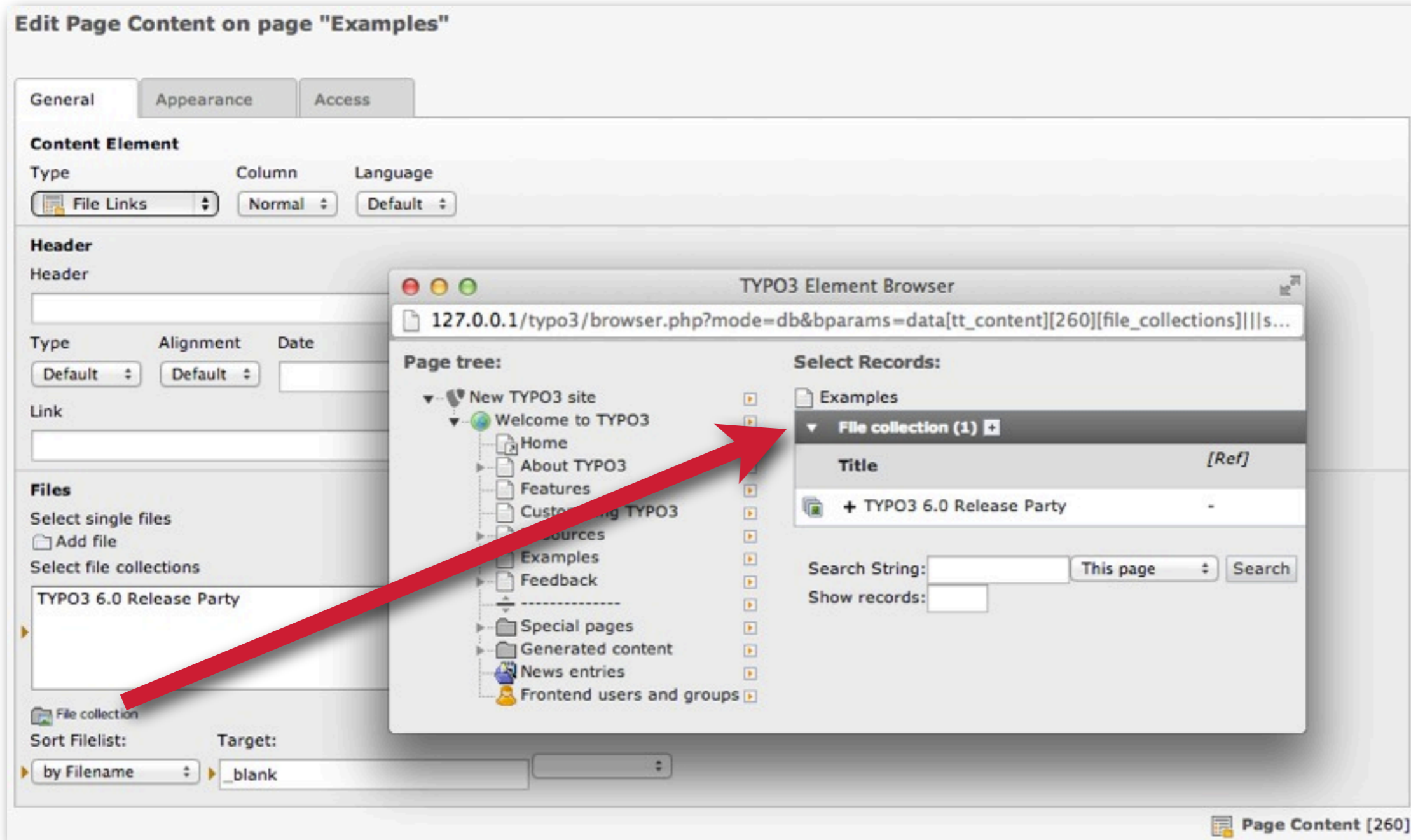
workspaces
Workspaces auto-publication
Workspaces cleanup preview links

- **Anlegen von File Collections über Systemdatensatz**
(Zwei Typen: Statisch und aus Folder)



The image shows two screenshots from the TYPO3 CMS interface. The left screenshot, titled "New record", displays a sidebar menu with categories like "Examples", "htmlArea RTE", and "System Records". Under "System Records", the "File collection" option is highlighted with a red arrow. The right screenshot, titled "Edit File collection 'TYPO3 6.0 Release Party' on page 'Examples'", shows the configuration form for a file collection. The "Language" is set to "Default", the "Title" is "TYPO3 6.0 Release Party", and the "Type" dropdown is open, showing "Static selection of files" selected and "Folder from Storage" as an alternative. Below the type selection, there is a "FILES" section with a "Create new relation" button and two file entries: "typo3_image3.jpg" and "typo3_image4.jpg", each with a small thumbnail image.

- Referenzieren von File Collections z.B. aus „File Links“ Content Objekt



The screenshot shows the 'Edit Page Content' interface for a page titled 'Examples'. The 'Content Element' section is set to 'File Links'. The 'Header' section is also visible. The 'Files' section shows a list of file collections, with 'TYPO3 6.0 Release Party' selected. The 'TYPO3 Element Browser' window is open, showing a tree view of the site structure and a 'Select Records' table. A red arrow points from the 'TYPO3 Element Browser' window to the 'File Links' content element in the main editor.

TYPO3 Element Browser

127.0.0.1/typo3/browser.php?mode=db&bparams=data[tt_content][260][file_collections]|||s...

Page tree:

- New TYPO3 site
 - Welcome to TYPO3
 - Home
 - About TYPO3
 - Features
 - Customizing TYPO3
 - Resources
 - Examples
 - Feedback
 -
 - Special pages
 - Generated content
 - News entries
 - Frontend users and groups

Select Records:

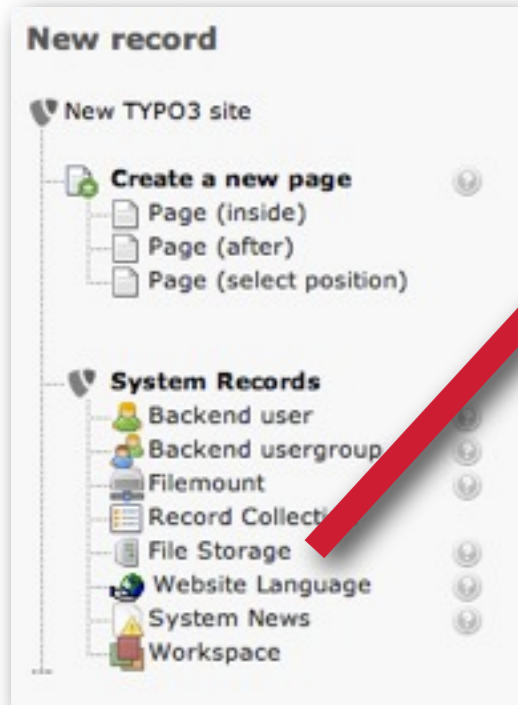
Title	[Ref]
+ TYPO3 6.0 Release Party	-

Search String: This page

Show records:

Page Content [260]

- Anlegen eines File Storages über den gleichnamigen Systemdatensatz



Create new File Storage on root level

General Configuration Access

Name
Editor

Description
Editor directory

Hide from listings

Create new File Storage on root level

General Configuration Access

Capabilities

Is browsable?	Is publically available?	Is writable?
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Is online?

Create new File Storage on root level

General Configuration Access

Driver
Local filesystem

Folder for manipulated and temporary images etc.
processed

Edit File Storage "Editor" on root level

General Configuration Access

Driver
Local filesystem

Driver Configuration

Base path:
fileadmin/editor/

Path type:
 relative
 absolute

Folder for manipulated and temporary images etc.
processed

FAL FILE STORAGE 2 / MOUNT

- **Auswahl des File Storages z.B. im Filemount**

Create new File Storage on root level

General Configuration Access

Name

Editor

Description

Editor directory

Hide from listings

Edit Filemount "News" on root level

Label: News Disable:

Storage: Editor

Folder: /user_upload/news/

 Filemount [1]

- **Es gibt 5 neue Tabellen für die Verwaltung von FAL**

- **sys_file**

Repräsentiert alle Dateien, die von TYPO3 getrackt werden - Dateien mit zusätzlichen Metadaten

- **sys_file_collection**

Repräsentation einer Liste von sys_file Datensätzen

- **sys_file_processedfile**

Temporäre Daten, wie Thumbnails, Previews, ...

- **sys_file_reference**

Benutzung eines sys_file Eintrages irgendwo in der TYPO3-Installation - ist wie eine MM-Tabelle zwischen sys_file und z.B. tt_content:image inkl. den zusätzlichen Meta-Daten für die Verwendung

- **sys_file_storage**

Definiert einen Root-Zugang zu einem Datei-Speicher - ähnlich einem Mount-Point. Jeder Storage ist an einen Treiber gebunden (local, webdav, amazons3) und ist damit der Einstiegspunkt

- **Programmierung: Datei kopieren**

```
$storageUid = 17;
$someFileIdentifier = 'templates/images/banner.jpg';
$someFolderIdentifier = 'website/images/';

$storageRepository = \TYPO3\CMS\Core\Utility
\GeneralUtility::makeInstance('\TYPO3\CMS\Core\Resource\StorageRepository');
$storage = $storageRepository->getByUid($storageUid);

// returns a t3lib_file_File object
$file = $storage->getFile($someFileIdentifier);

// returns a t3lib_file_File object
$folder = $storage->getFolder($someFolderIdentifier);

// returns the t3lib_file_File object of the new, copied file
$file->copyTo($folder);

// oder equivalent
$folder->addCopyOfFile($file);
```

- **Programmierung: Alle Dateien eines Verzeichnisses auslesen**

```
$storageRepository = \TYPO3\CMS\Core\Utility
\GeneralUtility::makeInstance('\TYPO3\CMS\Core\Resource\StorageRepository');
$availableStorages = $storageRepository->findAll();

foreach($availableStorages as $storage) {
    $rootFolder = $storage->getRootFolder();
    $subFolders = $rootFolder->getSubFolders();
    foreach($subFolders as $subFolder) {
        $filesInSubFolder = $subFolder->getFiles();
        $foldersInSubFolder = $subFolder->getSubFolders();
        ...
    }
}
```

- **Programmierung: General API**

```
getPublicUrl (\TYPO3\CMS\Core\Resource\ResourceInterface $fileOrFolder, $relativeToCurrentScript = FALSE)
isPathValid ($theFile)
sanitizeFileName ($fileName, $charset = '')
getAbsolutePath ($file)
getLowLevelFileInfo (\TYPO3\CMS\Core\Resource\FileInterface $file)
hash (\TYPO3\CMS\Core\Resource\FileInterface $file, $hashAlgorithm)
resourceExists ($identifier)

fileExists ($identifier)
fileExistsInFolder ($fileName, \TYPO3\CMS\Core\Resource\Folder $folder)

folderExists ($identifier)
folderExistsInFolder ($folderName, \TYPO3\CMS\Core\Resource\Folder $folder)
```

- **Programmierung: Storage API**

```
addFile($localFilePath, \TYPO3\CMS\Core\Resource\Folder $targetFolder, $fileName, \TYPO3\CMS\Core\nResource\AbstractFile $updateFileObject = NULL)  
createFile($fileName, \TYPO3\CMS\Core\Resource\Folder $parentFolder)  
getFile($identifier)  
getFileList($path, $start = 0, $numberOfItems = 0, $useFilters = TRUE, $loadIndexRecords = TRUE)  
hasFile($identifier)  
getFileContents(\TYPO3\CMS\Core\Resource\FileInterface $file)  
setFileContents(\TYPO3\CMS\Core\Resource\FileInterface $file, $contents)  
deleteFile(\TYPO3\CMS\Core\Resource\FileInterface $file)  
copyFile(\TYPO3\CMS\Core\Resource\FileInterface $file, \TYPO3\CMS\Core\Resource\Folder $targetFolder, $targetFileName = NULL, $conflictMode = 'renameNewFile')  
copyFileWithinStorage(\TYPO3\CMS\Core\Resource\FileInterface $file, \TYPO3\CMS\Core\Resource\Folder $targetFolder, $fileName)  
copyFileToTemporaryPath(\TYPO3\CMS\Core\Resource\FileInterface $file)  
moveFile($file, $targetFolder, $targetFileName = NULL, $conflictMode = 'renameNewFile')  
moveFileWithinStorage(\TYPO3\CMS\Core\Resource\FileInterface $file, \TYPO3\CMS\Core\Resource\Folder $targetFolder, $fileName)  
renameFile(\TYPO3\CMS\Core\Resource\FileInterface $file, $newName)  
updateFile(\TYPO3\CMS\Core\Resource\AbstractFile $file, $identifier = '', $storage = NULL)  
replaceFile(\TYPO3\CMS\Core\Resource\FileInterface $file, $localFilePath)  
getFileForLocalProcessing(\TYPO3\CMS\Core\Resource\FileInterface $fileObject, $writable = TRUE)  
getFileInfoByIdentifier($fileIdentifier)  
getFileInfo($identifier)  
getFilePermissions(\TYPO3\CMS\Core\Resource\FileInterface $file)
```

- **Programmierung: Folder API**

```
getRootLevelFolder()  
getDefaultFolder()  
createFolder($newFolderName, \TYPO3\CMS\Core\Resource\Folder $parentFolder)  
getFolderInFolder($name, \TYPO3\CMS\Core\Resource\Folder $parentFolder)  
moveFolderWithinStorage(\TYPO3\CMS\Core\Resource\Folder $folderToMove, \TYPO3\CMS\Core\Resource  
\Folder $targetFolder, $newFolderName)  
copyFolderWithinStorage(\TYPO3\CMS\Core\Resource\Folder $folderToCopy, \TYPO3\CMS\Core\Resource  
\Folder $targetFolder, $newFolderName)  
renameFolder(\TYPO3\CMS\Core\Resource\Folder $folder, $newName)  
deleteFolder(\TYPO3\CMS\Core\Resource\Folder $folder, $deleteRecursively = FALSE)  
isFolderEmpty(\TYPO3\CMS\Core\Resource\Folder $folder)  
getFolderPermissions(\TYPO3\CMS\Core\Resource\Folder $folder)  
isWithin(\TYPO3\CMS\Core\Resource\Folder $container, $content)
```

- **File-Information**

Über `getText data = file:[UID|current]:[fieldName]` kann entweder auf eine `sys_file` UID oder auf das aktuelle Element (über „current“) zugegriffen werden. Anschließend wird der Feldname notiert, den man auslesen will.

```
page.10.data = file:current:title
```

oder

```
page.10.data = file:17:title
```


- **cObject FILES**

```
20 = FILES
20 {
    references {
        table = pages
        uid.data = page:uid
        fieldName = media
    }
    # oder: sys_file_references mit UID 27
    references = 27

    # UIDs aus sys_files
    files = 12,14,15
    # Mit stdWrap
    files.field = some_field

    collections.field = file_collections

    folders.field = select_key

    renderObj = COA
    renderObj {
        10 = IMAGE
        10 {
            file.import.data = file:current:originalUid // file:current:uid
            file.width = 150
            ...
        }
    }
}
```

```
page.10 = FILES
page.10.files = 12,15,16
page.10.collections = 2,9
page.10.folders = 1:mypics/

page.10.renderObj = TEXT
page.10.renderObj {
    data = file:current:size
    wrap = <p>File size:<strong>|</strong></p>
}
```

- **cObject FILES**

- **files (stdWrap)**

Laden von Dateien aus sys_file - angegeben wird eine kommaseparierte Liste von UIDs

- **references**

Ermittelt alle Referenzen eines Feldes (vom Typ IRRE mit sys_file_reference als Kind).

Man kann entweder die UID direkt angeben, oder mittels table, uid und fieldName arbeiten.

- **references.table (stdWrap)**

Spezifiziert die Tabelle, aus der die Referenz ermittelt wird (z.B. tt_content oder pages)

- **references.uid (stdWrap)**

Gibt die UID des Datensatzes an, aus welchem die Referenz ermittelt wird (z.B. Seite mit UID 7)

Hier ist auch stdWrap möglich: uid.data = page:uid)

- **references.fieldName (stdWrap)**

Feldname für die Ermittlung der Referenzen

- **collections (stdWrap)**

Kommaseparierte Liste von sys_file_collection UIDs

- **folders (stdWrap)**

Kommaseparierte Liste von Folder-Identifizier: [storageUid]:[folderIdentifizier] (2:mypics/,4:myimages/)

storageUid = UID des Storages (sys_file_storage) / folderIdentifizier = Relativer Pfad

- **renderObject (cObject +optionSplit)**

Rendering eines jeden Files. Infos über file und current: z.B. data = file:current:...

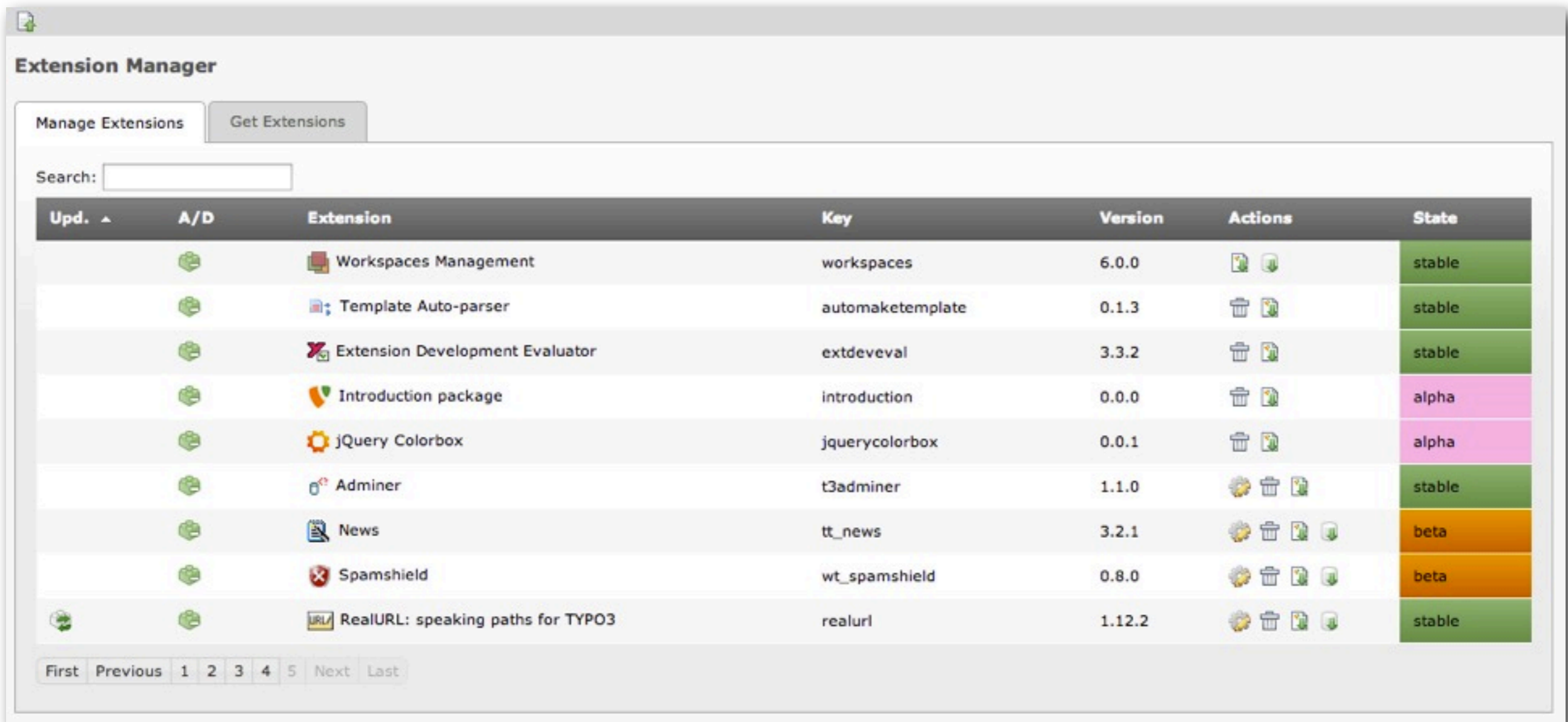
- **stdWrap (-> stdWrap)**

EXTENSION MANAGER

EM die Dritte :-)

- Der Extension Manager wurde komplett neu auf Basis von Extbase geschrieben. Dabei wurde dieser auch deutlich verschlankt.
- Sprachenbehandlung => Wurde in eine eigene Extension („lang“) ausgelagert
- Editieren von Dateien => Kann durch andere Extensions durchgeführt werden
- Upload extension ins TER => Wurde (Wird) in die Extension „extdeveval“ integriert (Stand 27.11.2012 noch nicht im TER)
- Die zugehörigen Tabellen in der Datenbank wurden verändert:
 - **ALT:**
sys_ter
cache_extensions
 - **NEU:**
tx_extensionmanager_domain_model_extension
tx_extensionmanager_domain_model_repository

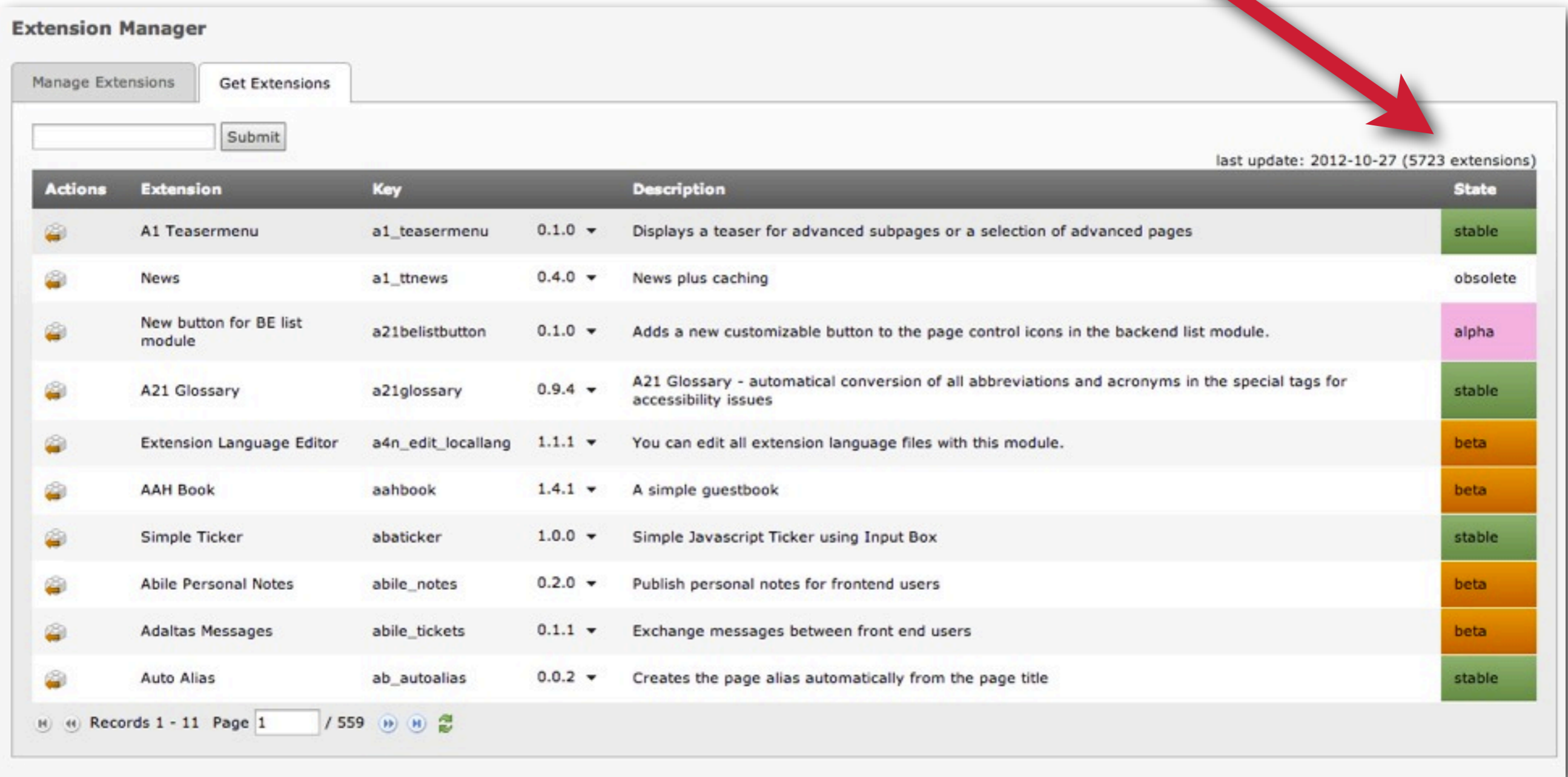
- Alle Aktionen sind klar gekennzeichnet
- Manueller Upload links oben













The screenshot shows the 'Extension Manager' interface with a search bar and a table of installed and available extensions. The table columns are: Upd., A/D, Extension, Key, Version, Actions, and State. The extensions listed are: Workspaces Management (stable), Template Auto-parser (stable), Extension Development Evaluator (stable), Introduction package (alpha), jQuery Colorbox (alpha), Adminer (stable), News (beta), Spamshield (beta), and RealURL: speaking paths for TYPO3 (stable). The interface includes buttons for 'Manage Extensions' and 'Get Extensions' at the top left, and a pagination bar at the bottom.

Upd. ^	A/D	Extension	Key	Version	Actions	State
		Workspaces Management	workspaces	6.0.0		stable
		Template Auto-parser	automaketemplate	0.1.3		stable
		Extension Development Evaluator	extdeveval	3.3.2		stable
		Introduction package	introduction	0.0.0		alpha
		jQuery Colorbox	jquerycolorbox	0.0.1		alpha
		Adminer	t3adminer	1.1.0		stable
		News	tt_news	3.2.1		beta
		Spamshield	wt_spamshield	0.8.0		beta
		RealURL: speaking paths for TYPO3	realurl	1.12.2		stable

- Remote Update erfolgt automatisch - manuell durch Klick rechts oben

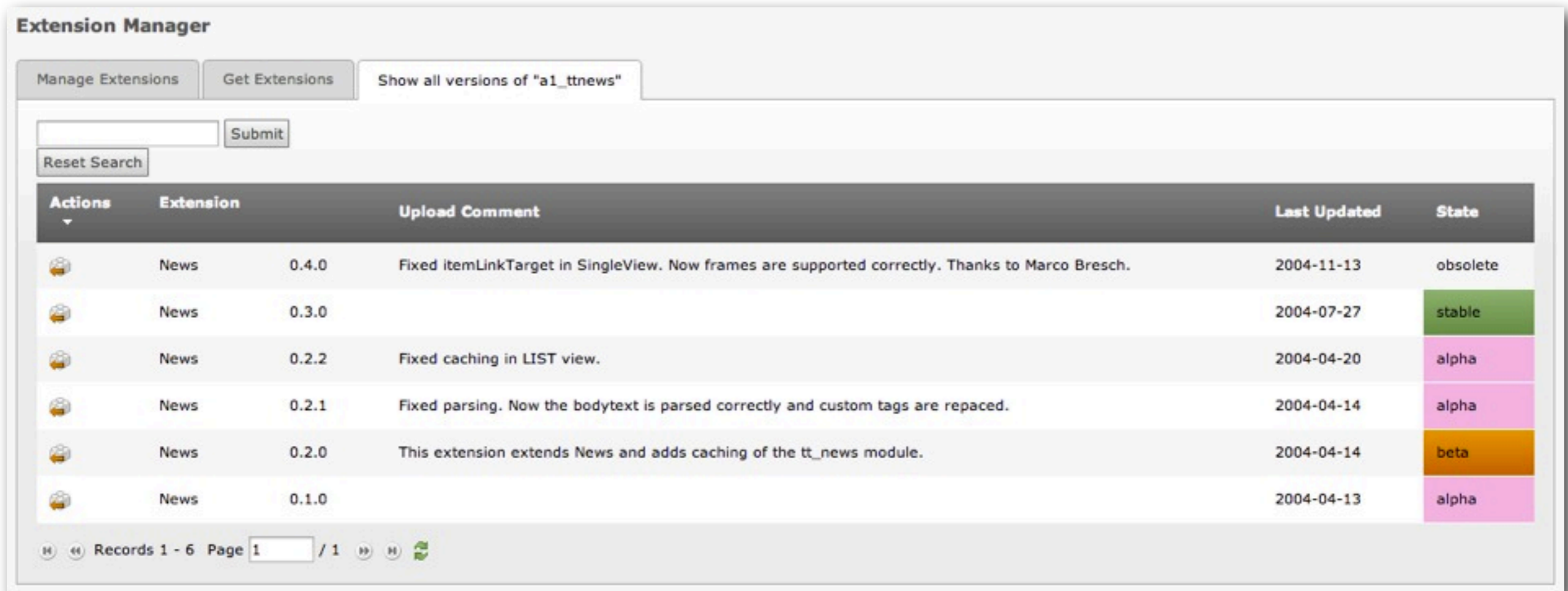


last update: 2012-10-27 (5723 extensions)

Actions	Extension	Key		Description	State
	A1 Teasermenu	a1_teasermenu	0.1.0 ▾	Displays a teaser for advanced subpages or a selection of advanced pages	stable
	News	a1_ttnews	0.4.0 ▾	News plus caching	obsolete
	New button for BE list module	a21belistbutton	0.1.0 ▾	Adds a new customizable button to the page control icons in the backend list module.	alpha
	A21 Glossary	a21glossary	0.9.4 ▾	A21 Glossary - automatical conversion of all abbreviations and acronyms in the special tags for accessibility issues	stable
	Extension Language Editor	a4n_edit_locallang	1.1.1 ▾	You can edit all extension language files with this module.	beta
	AAH Book	aahbook	1.4.1 ▾	A simple guestbook	beta
	Simple Ticker	abaticker	1.0.0 ▾	Simple Javascript Ticker using Input Box	stable
	Abile Personal Notes	abile_notes	0.2.0 ▾	Publish personal notes for frontend users	beta
	Adaltas Messages	abile_tickets	0.1.1 ▾	Exchange messages between front end users	beta
	Auto Alias	ab_autoalias	0.0.2 ▾	Creates the page alias automatically from the page title	stable

Records 1 - 11 Page 1 / 559

- **Alle Versionen einer Extension durch Klick auf „Show all versions“**



The screenshot shows the 'Extension Manager' interface. At the top, there are three tabs: 'Manage Extensions', 'Get Extensions', and 'Show all versions of "a1_ttnews"'. Below the tabs is a search bar with a 'Submit' button and a 'Reset Search' button. The main content is a table with the following columns: 'Actions', 'Extension', 'Upload Comment', 'Last Updated', and 'State'. The table lists six versions of the 'News' extension:

Actions	Extension	Version	Upload Comment	Last Updated	State
	News	0.4.0	Fixed itemLinkTarget in SingleView. Now frames are supported correctly. Thanks to Marco Bresch.	2004-11-13	obsolete
	News	0.3.0		2004-07-27	stable
	News	0.2.2	Fixed caching in LIST view.	2004-04-20	alpha
	News	0.2.1	Fixed parsing. Now the bodytext is parsed correctly and custom tags are repaced.	2004-04-14	alpha
	News	0.2.0	This extension extends News and adds caching of the tt_news module.	2004-04-14	beta
	News	0.1.0		2004-04-13	alpha

At the bottom of the table, there is a pagination control showing 'Records 1 - 6 Page 1 / 1' and navigation icons.

LOCALCONFIGURATION

Ersatz für die localconf.php

- **Die Datei `typo3conf/localconf.php` wird abgelöst**
 - Stattdessen gibt es nun die neue Datei `typo3conf/LocalConfiguration.php`
 - Diese Datei gibt lediglich ein **Konfigurations-Array** zurück
 - Sobald die Konfiguration verändert wird (beispielsweise durch den Extensionmanager), wird diese Datei komplett neu geschrieben
 - Daher ist kein eigener Code mehr in dieser Datei erlaubt (keine PHP-Logik)
 - Um aber eine Erweiterung dieses Codes in einer Instanz zu erlauben, kann dieser in der Datei `typo3conf/AdditionalConfiguration.php` platziert werden, die durch den neuen Bootstrapper direkt hinter der `LocalConfiguration.php` ausgeführt wird.
 - Ein Upgrade Wizard überführt die alte `localconf.php` Datei in das neue Schema

- **Aufbau der Datei LocalConfiguration.php**

```
• return array(  
    'BE' => array(  
        'disable_exec_function' => 0,  
        ...  
    ),  
    'DB' => array(  
        'database' => 'typo360b1',  
        ...  
    ),  
    'EXT' => array(  
        'extConf' => array(  
            'automaketemplate' => 'a:0:{}',  
            ...  
        ),  
        'extList' => 'info,perm, ...',  
        'extListArray' => array(  
            0 => 'info',  
            1 => 'perm',  
            ...  
        ),  
    ),  
);
```

- **Setzen und Lesen von Konfigurationsoptionen**

- **Lesen (am Beispiel BE/loginSecurityLevel)**

```
\TYPO3\CMS\Core\Configuration  
\ConfigurationManager::getLocalConfigurationValueByPath('BE/  
loginSecurityLevel') == 'rsa'
```

- **Setzen (am Beispiel BE/loginSecurityLevel)**

```
\TYPO3\CMS\Core\Configuration  
\ConfigurationManager::setLocalConfigurationValueByPath('BE/  
loginSecurityLevel', 'rsa')
```

- **Schlüssel**

```
BE  
DB  
EXT / extListArray / 1 === 'perm' (Pfade mit / bis zum benötigten Schlüssel)  
EXTCONF  
FE  
GFX  
SYS  
INSTALL
```

- **Default-Konfiguration**

- Die Default-Konfiguration befindet sich nun nicht mehr unter:

`t3lib/config_default.php`

sondern unter

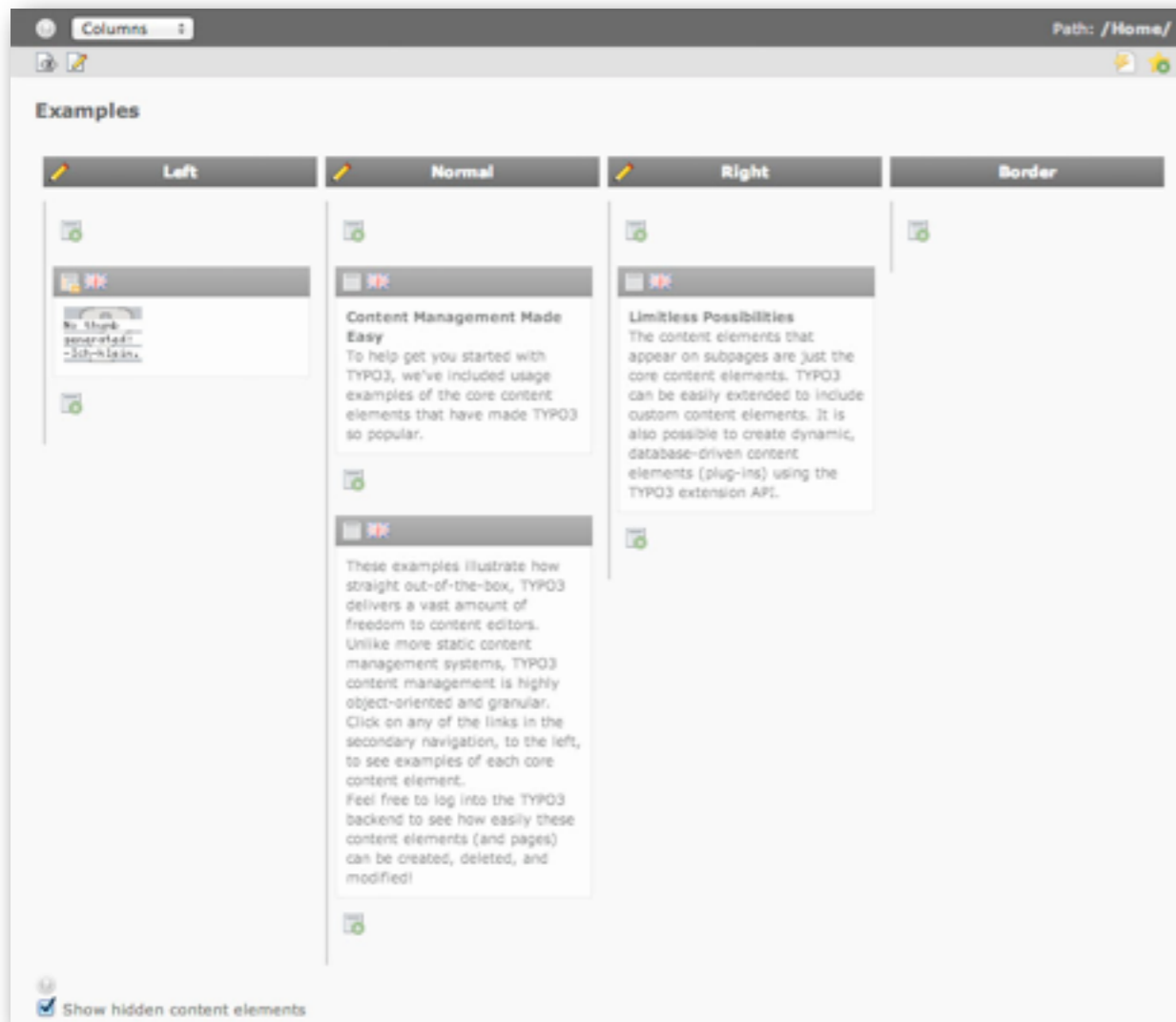
`t3lib/stddb/DefaultConfiguration.php`

Dies betrifft nur die Configuration ansich (`$TYPO3_CONF_VARS`) - alle anderen Teile sind nun im Bootstrap.

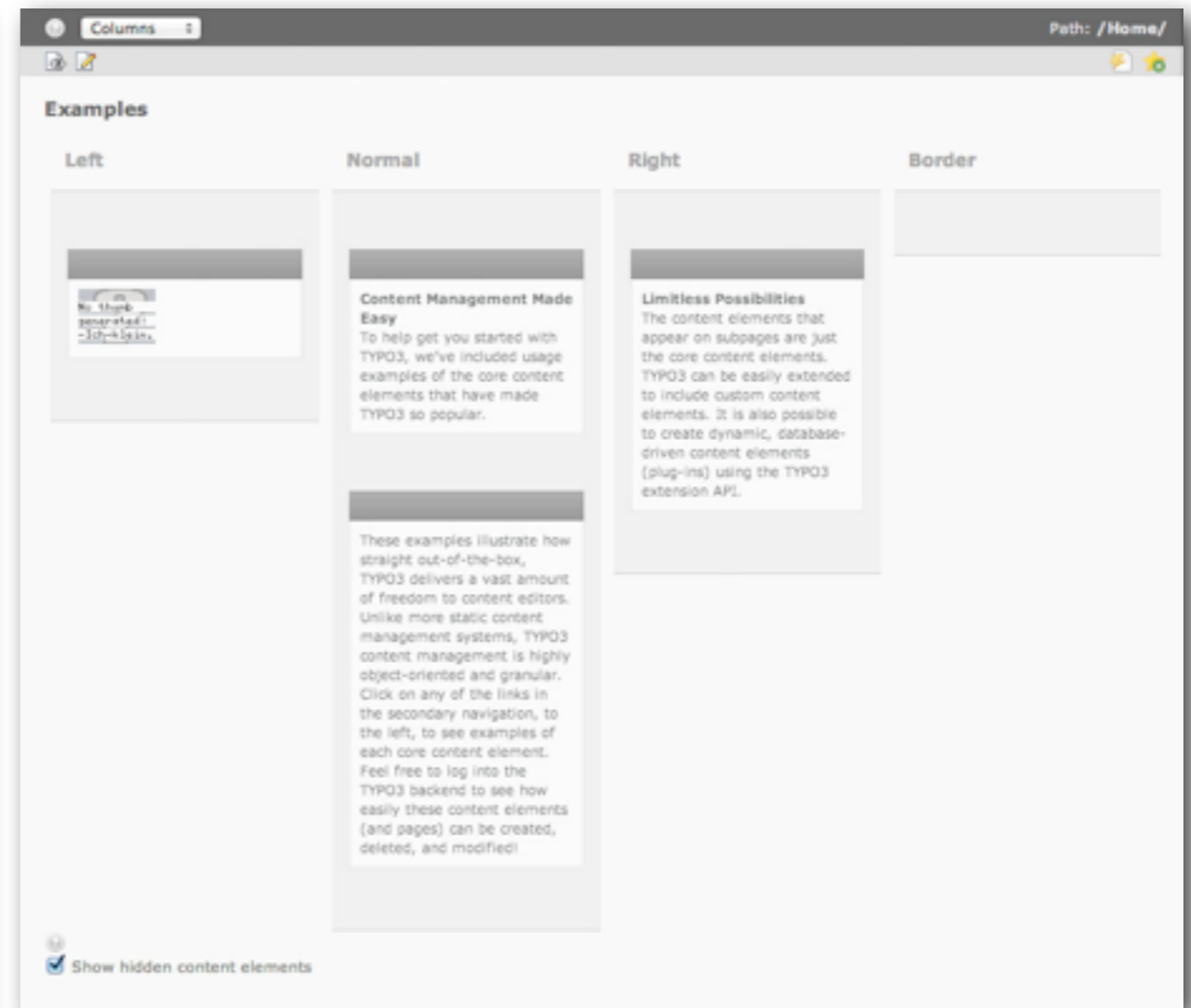
ALLGEMEINES

Änderungen

- Das Backend-Modul „Page“ wurde hinsichtlich der Usability umgestaltet:

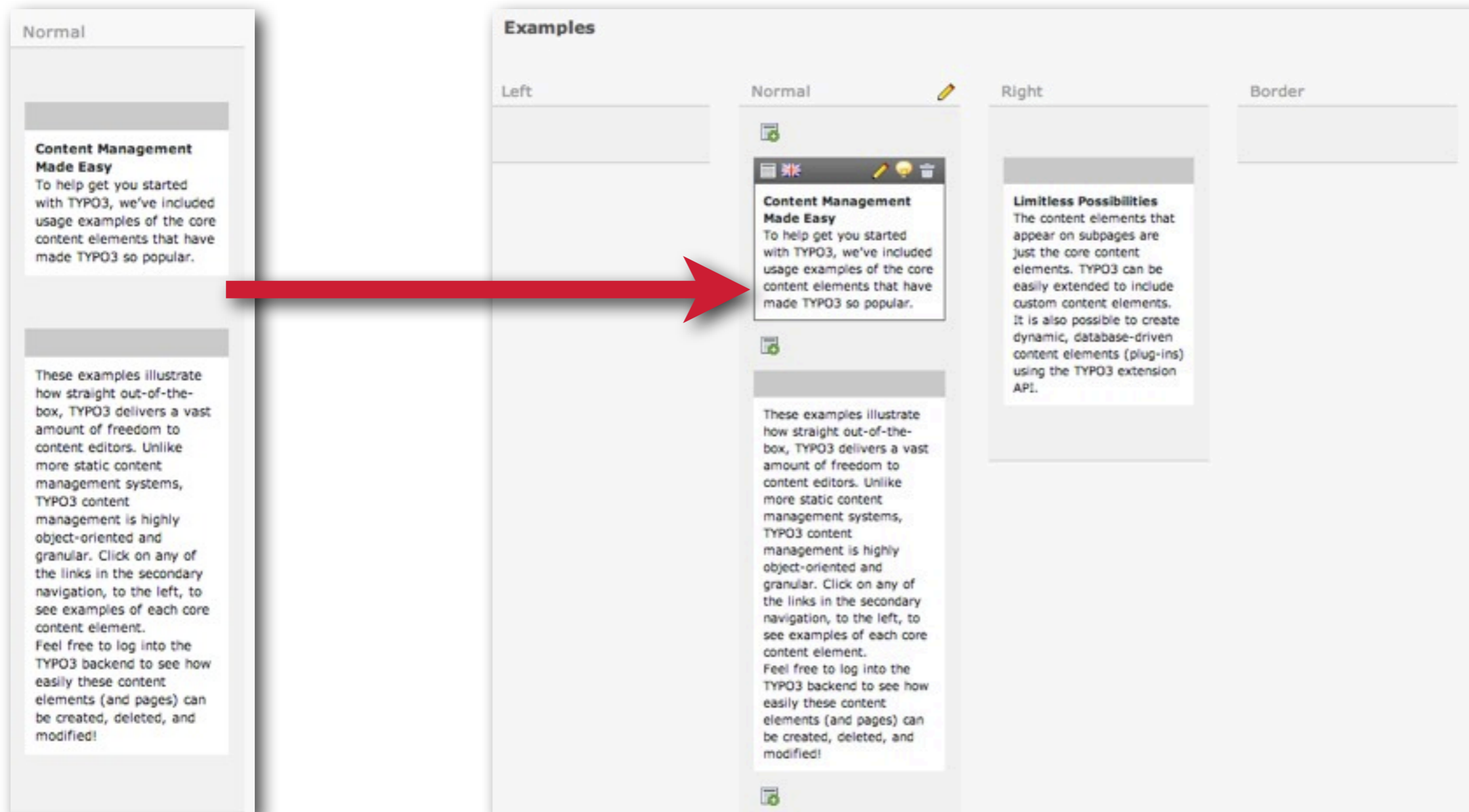


alt



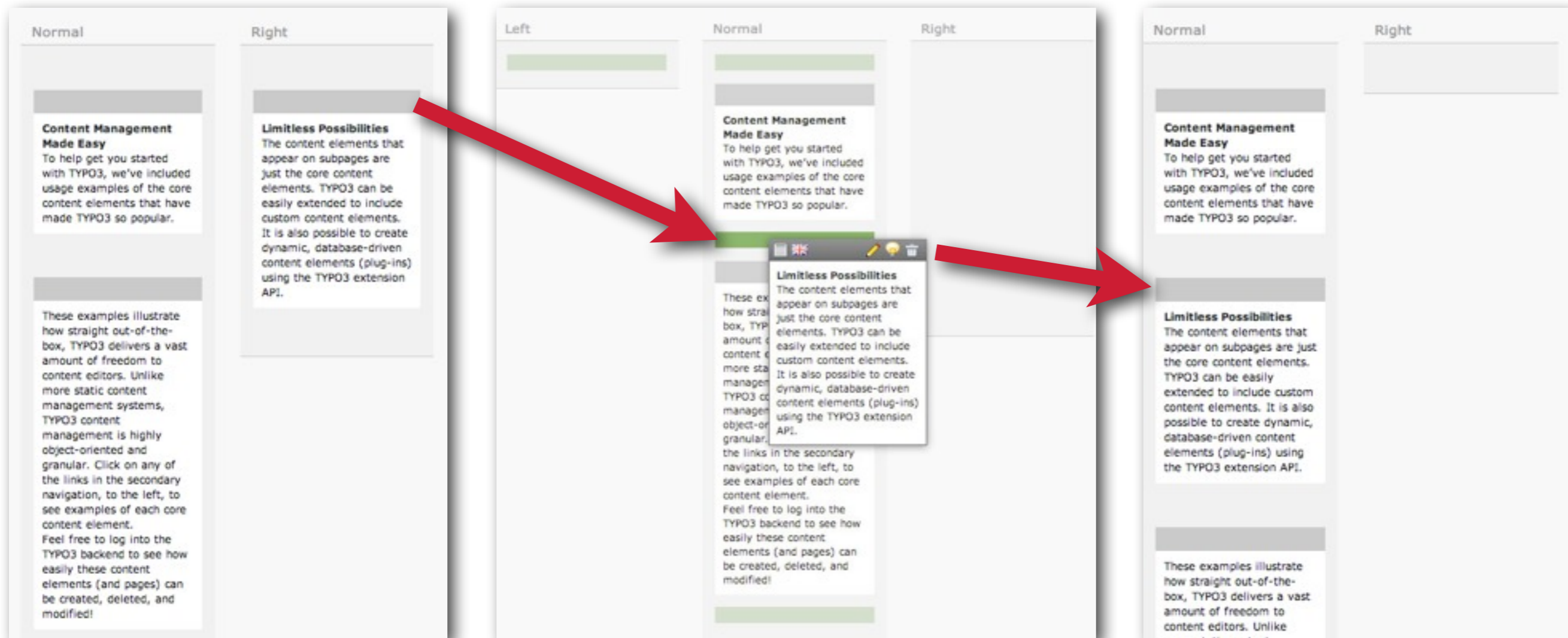
neu

- Die Elemente werden erst beim darüberfahren mit der Maus sichtbar:

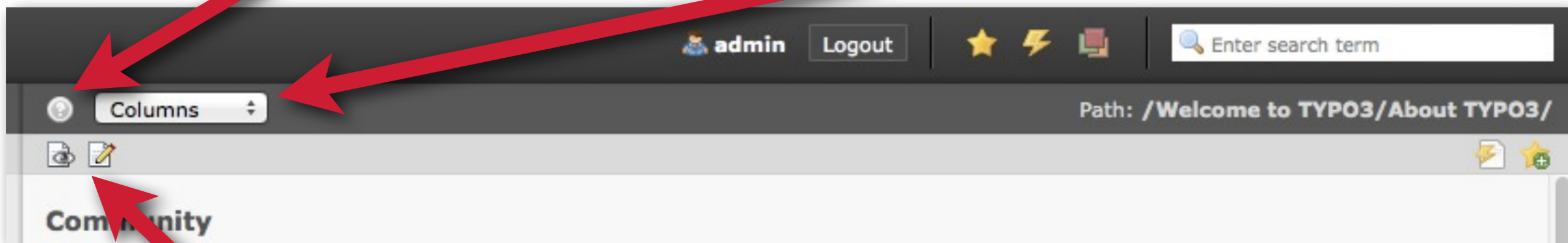


DRAG & DROP

- Die Content-Elemente können nun per Drag&Drop im Page-Modul verschoben werden:



- Die oberen zwei Reihen wurden bei allen Backend-Modulen vertauscht um damit eine bessere Usability zu erreichen.
- Zuerst kommt die Context Sensitive Hilfe (CSH) und das Funktionen-Menü und

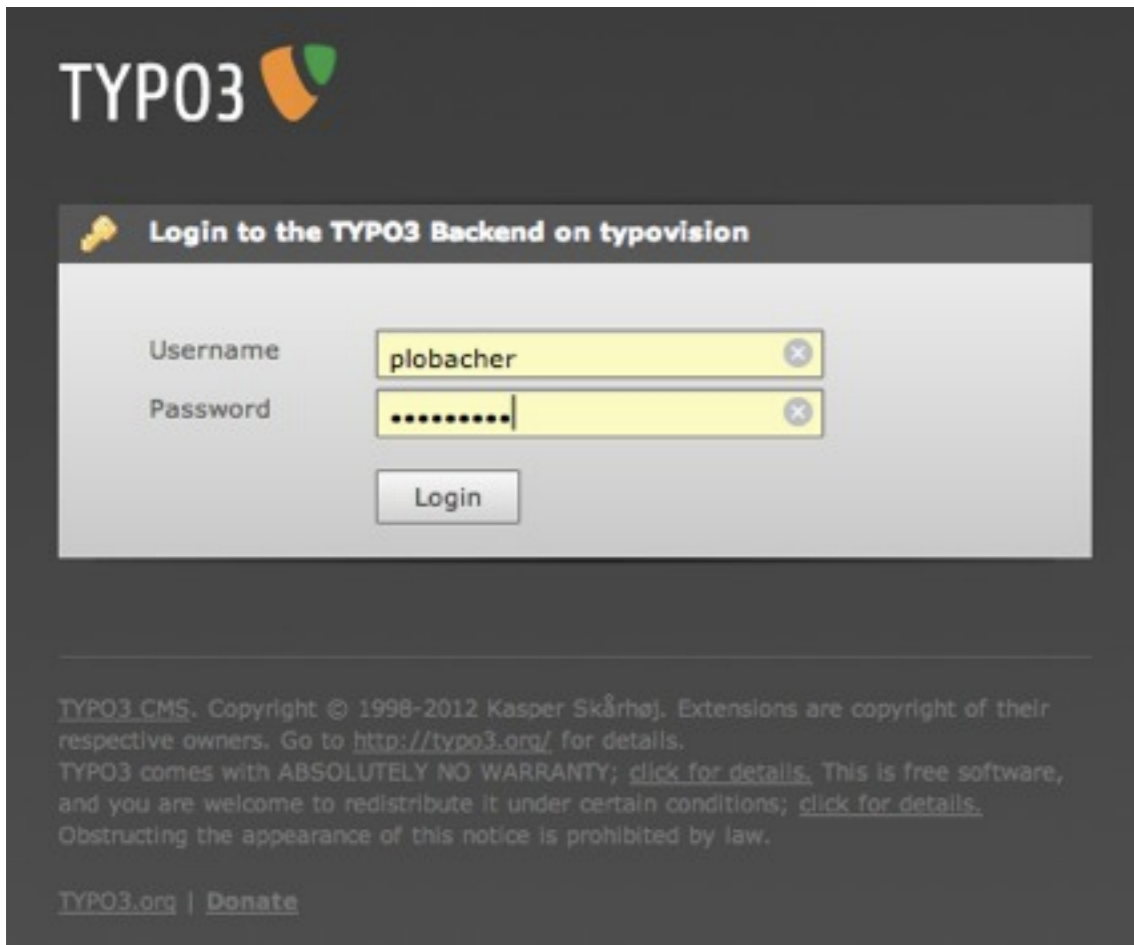


- In der zweiten Reihe befinden sich nun die Buttons (Speichern, Schließen, Shortcut, ...)

NEUER LOGIN SCREEN

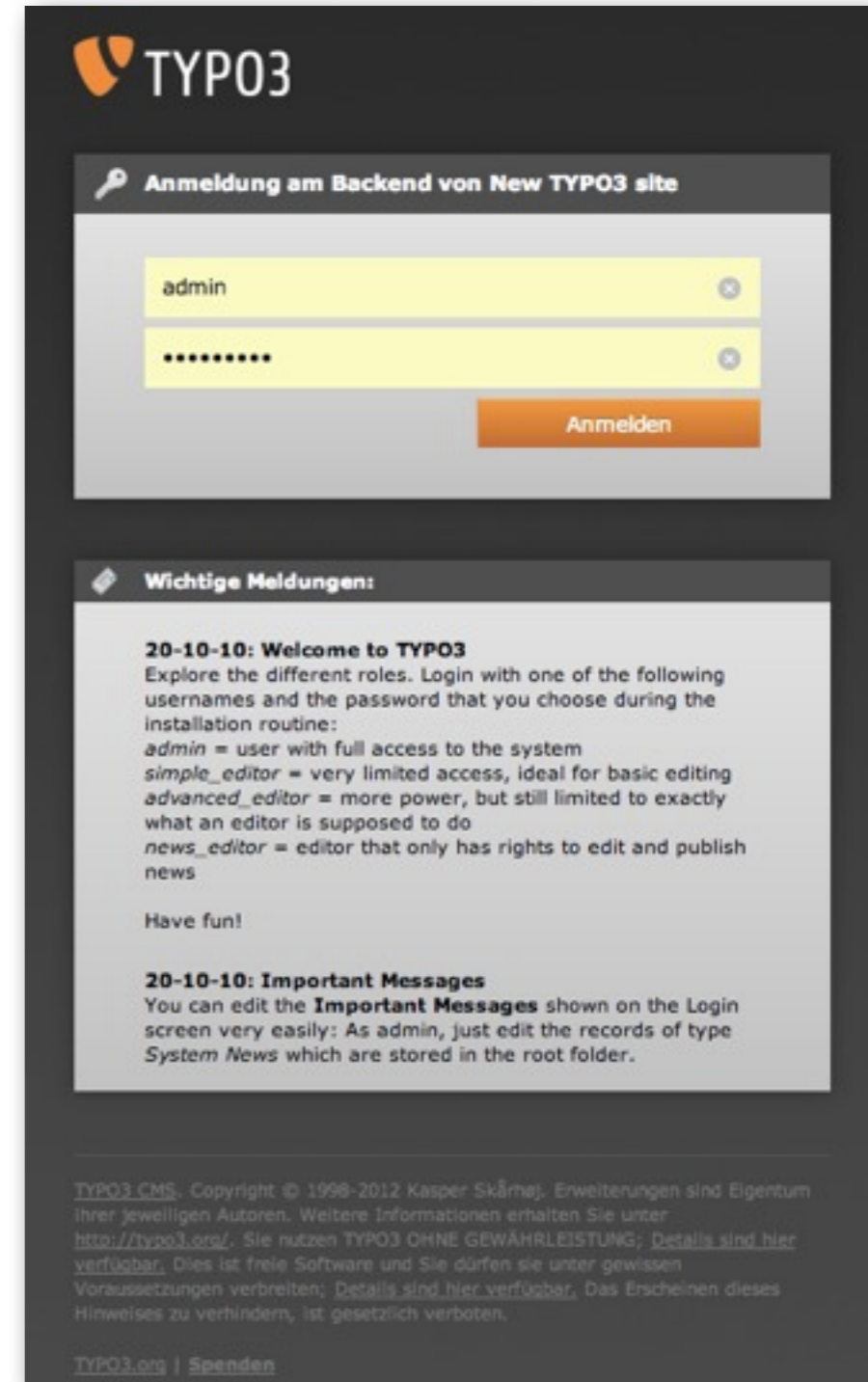
- Der Login-Screen wurde überarbeitet

alt



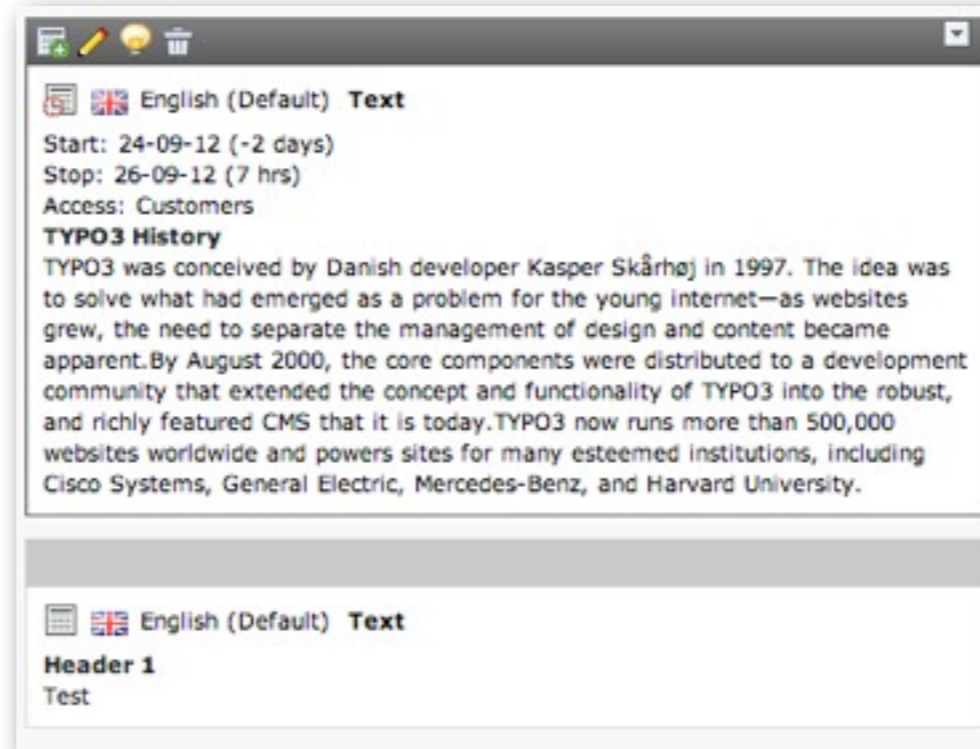
The screenshot shows the old login interface. At the top left is the TYPO3 logo. Below it is a header bar with a key icon and the text "Login to the TYPO3 Backend on typovision". The main form area contains two input fields: "Username" with the value "plobacher" and "Password" with masked characters. A "Login" button is positioned below the password field. At the bottom, there is a copyright notice and a "Donate" link.

neu

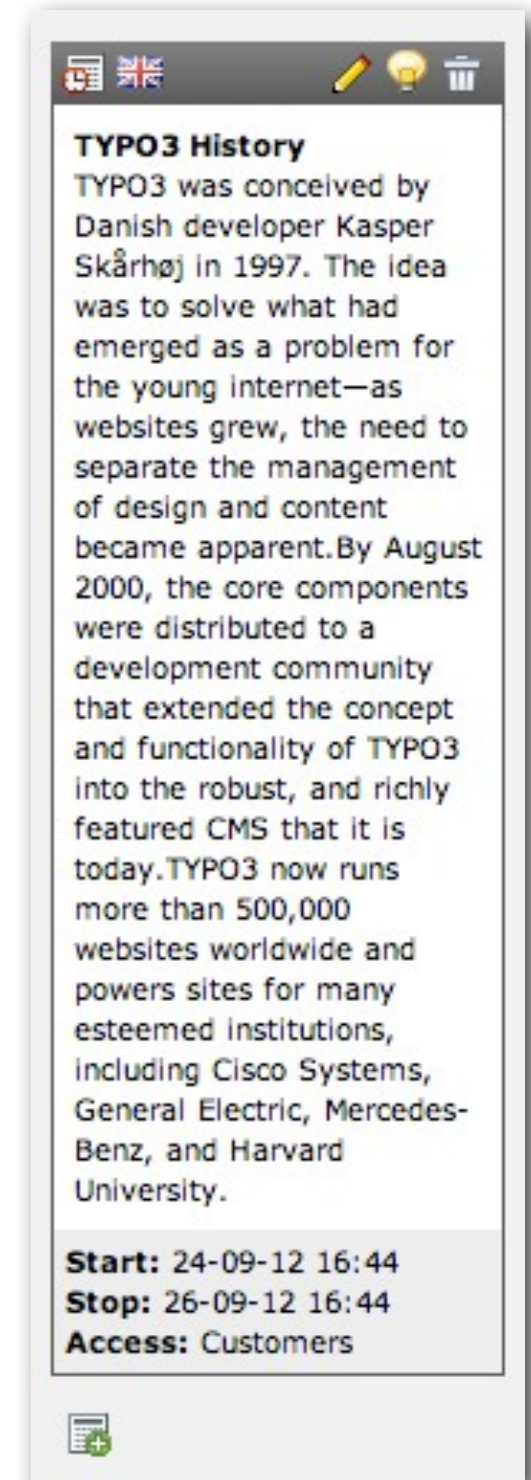


The screenshot shows the new login interface. At the top left is the TYPO3 logo. Below it is a header bar with a key icon and the text "Anmeldung am Backend von New TYPO3 site". The main form area contains two input fields: the first contains "admin" and the second contains masked characters. An "Anmelden" button is positioned below the password field. Below the form is a section titled "Wichtige Meldungen:" containing two messages: "20-10-10: Welcome to TYPO3" and "20-10-10: Important Messages". At the bottom, there is a copyright notice and a "Spenden" link.

- Zusätzliche Infos wie Hidden/Start/Stop wurden in den Footer des CE verschoben
- Sortiert-Buttons sind verschwunden, da Drag & Drop
- Action-Buttons wurden nach rechts geschoben
- Label des Content-Types entfernt
- Content-Typ Icon in den Header verschoben
- Lock/language Icon in den Header verschoben
- Icon "Neues Content Element danach anlegen" wurde nach den Footer des CE verschoben



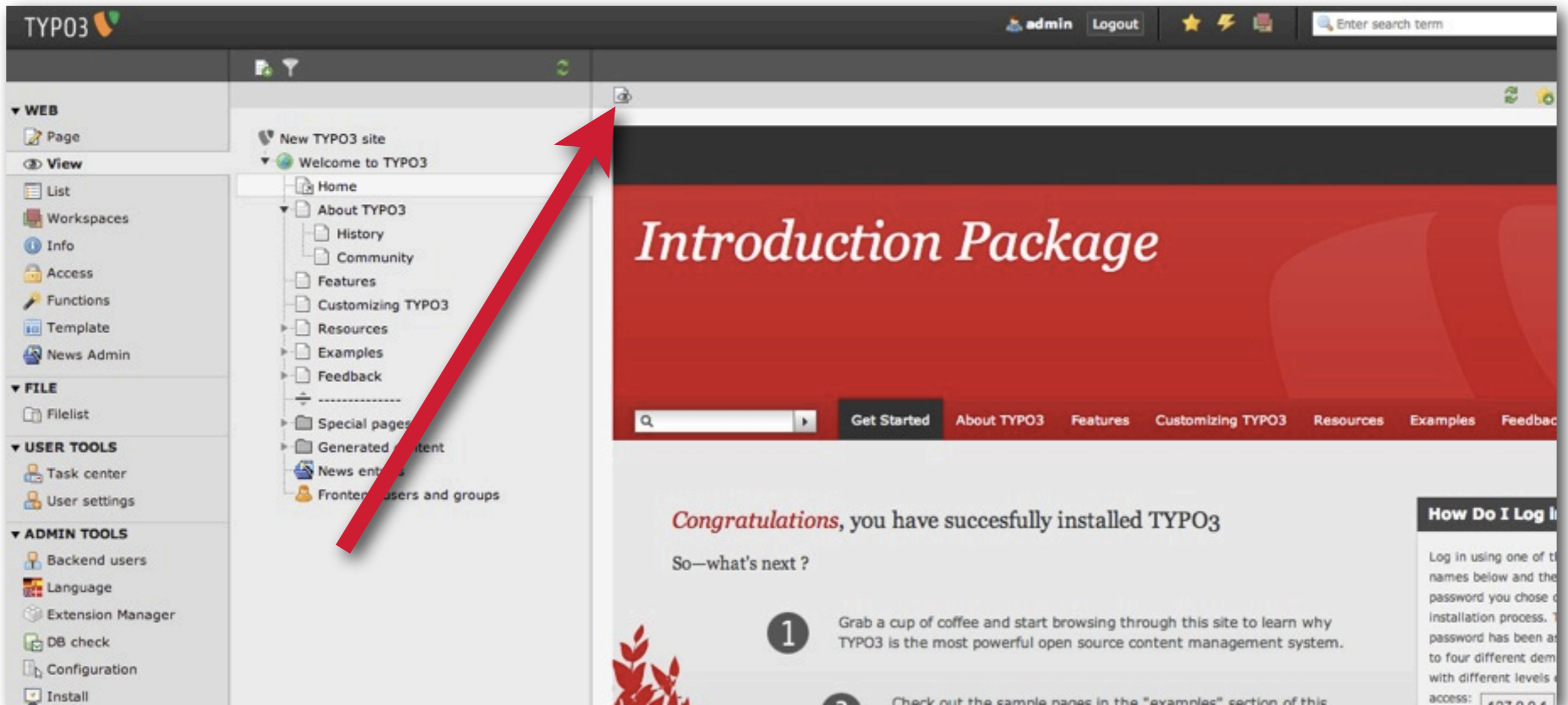
alt



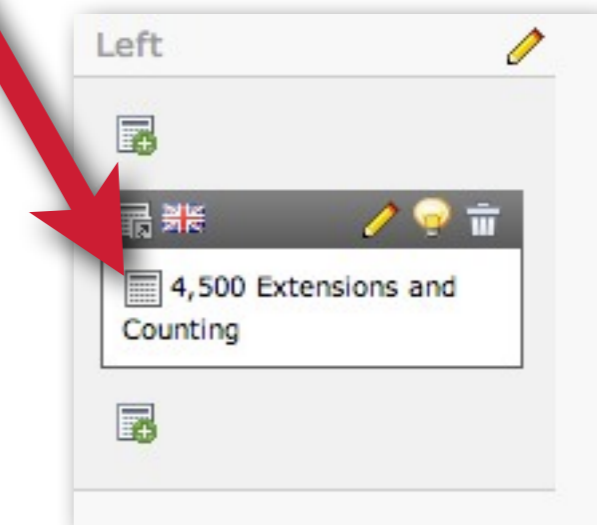
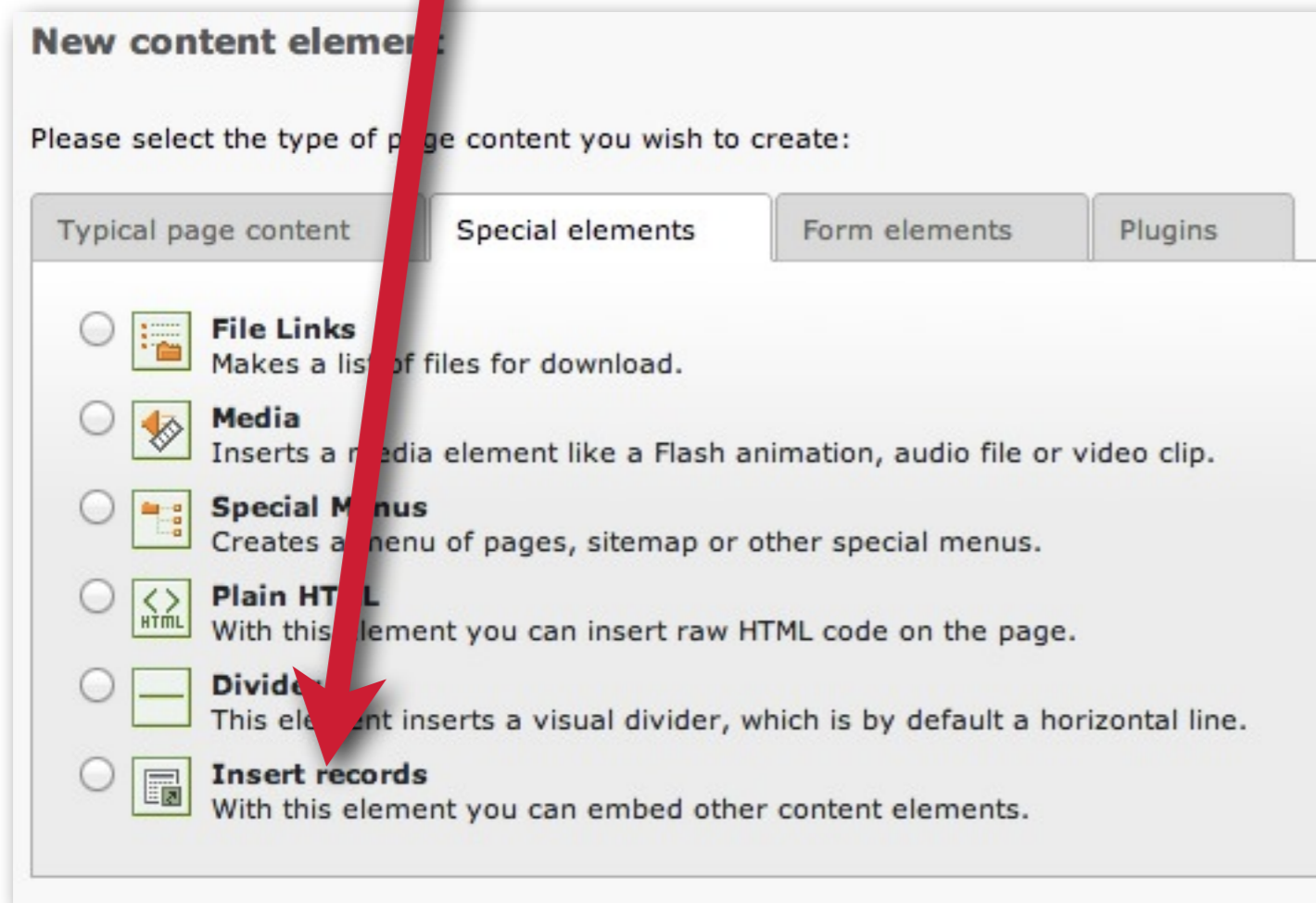
neu

ICON IN MODUL VIEW

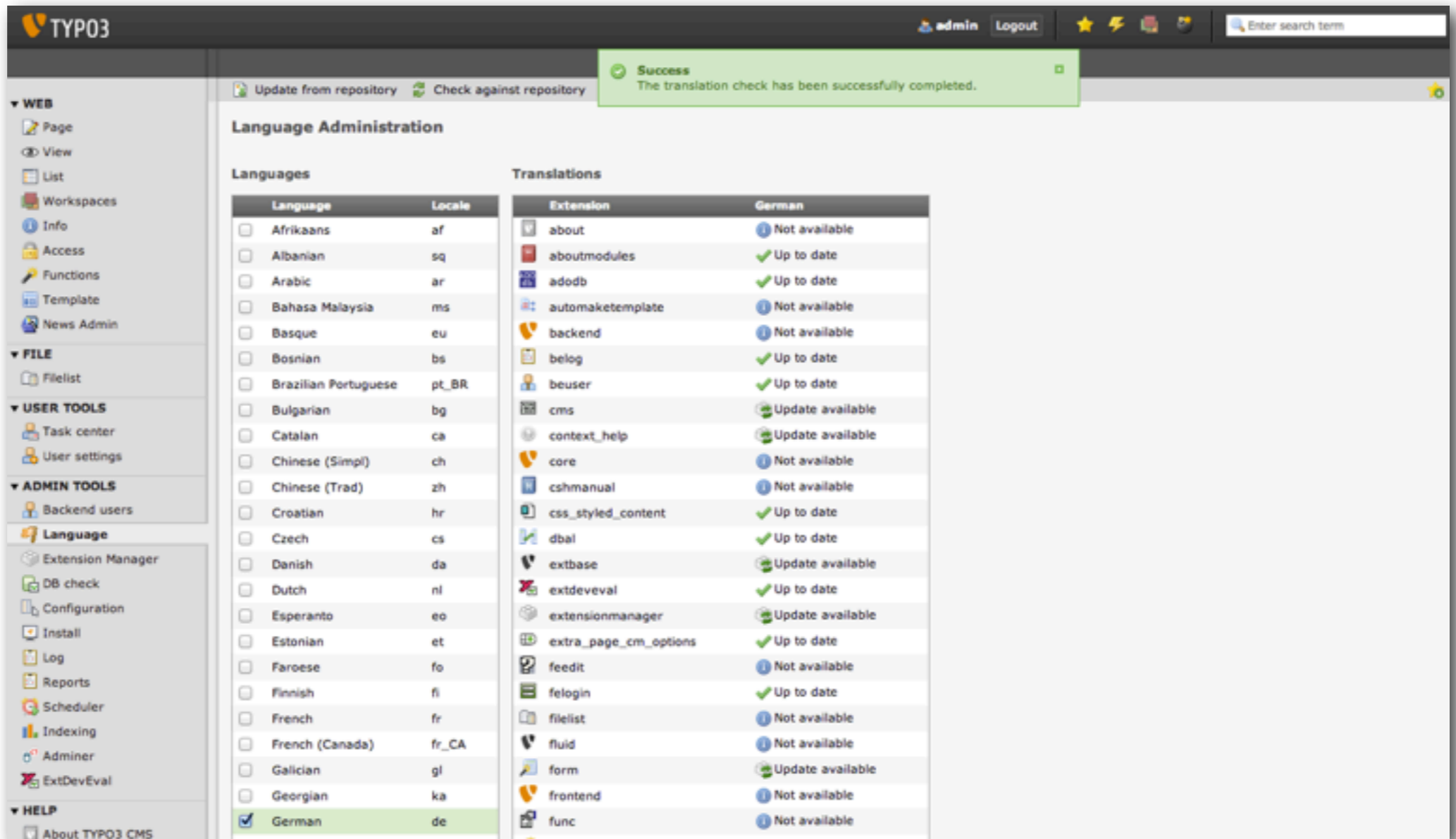
- Im Modul „View“ (viewpage) kann man nun per Icon die Ansicht in einem neuen Fenster öffnen lassen. Die Extension **ext:view** ein Rewrite auf Basis von TYPO3 Extbase.



- Das „Insert Record“ Content Element wurde insofern verbessert, als es nun im New Content Element Wizard auswählbar ist und zudem im Modul Page das referenzierte Content-Elements angezeigt wird (das Original ist von dort aus editierbar)..



- Da der Extension-Manager neu geschrieben wurde, hat man die früher dort enthaltene Funktion zum Sprachhandling nun in ein eigenes Modul ausgelagert:

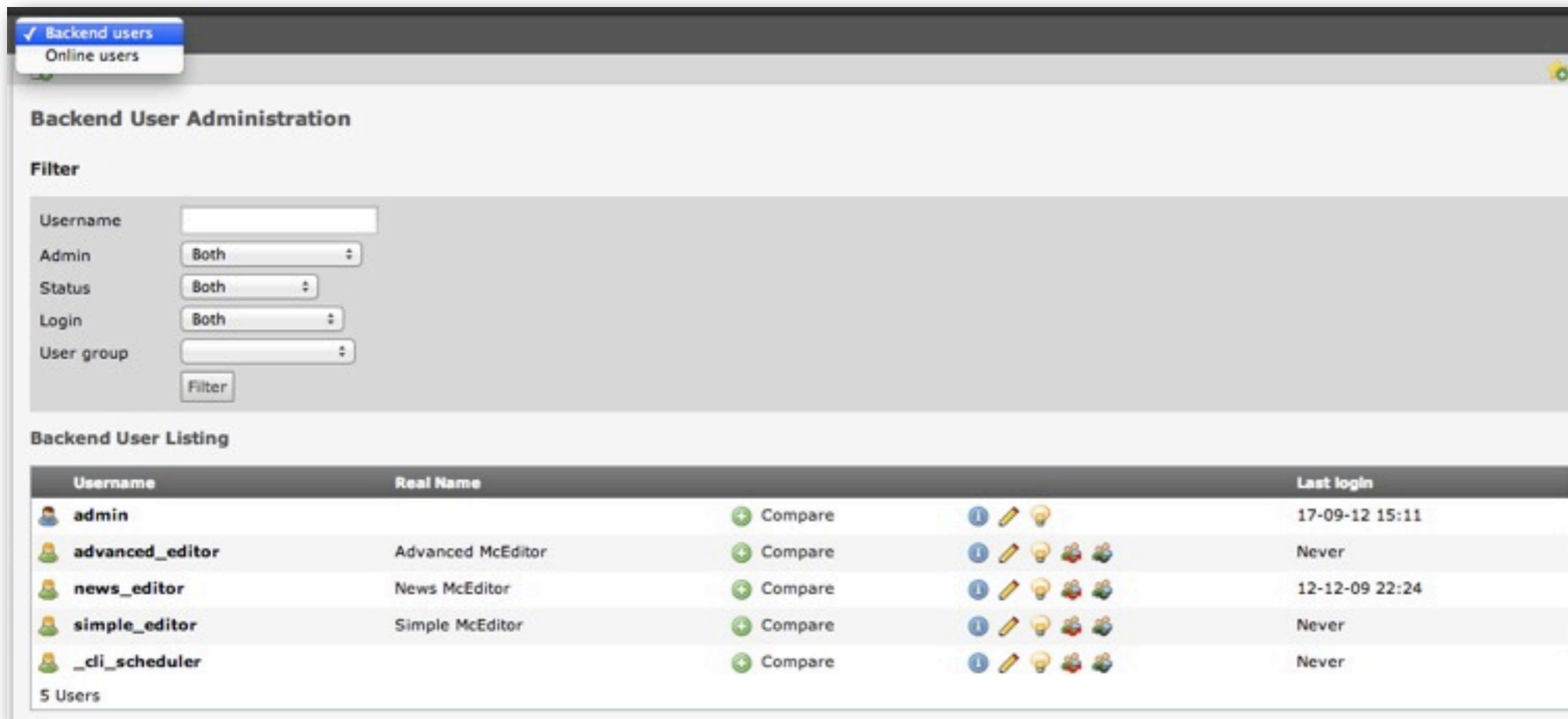


Success
The translation check has been successfully completed.

Language Administration

Languages		Translations	
Language	Locale	Extension	German
<input type="checkbox"/> Afrikaans	af	about	Not available
<input type="checkbox"/> Albanian	sq	aboutmodules	Up to date
<input type="checkbox"/> Arabic	ar	adodb	Up to date
<input type="checkbox"/> Bahasa Malaysia	ms	automaketemplate	Not available
<input type="checkbox"/> Basque	eu	backend	Not available
<input type="checkbox"/> Bosnian	bs	belog	Up to date
<input type="checkbox"/> Brazilian Portuguese	pt_BR	beuser	Up to date
<input type="checkbox"/> Bulgarian	bg	cms	Update available
<input type="checkbox"/> Catalan	ca	context_help	Update available
<input type="checkbox"/> Chinese (Simpl)	ch	core	Not available
<input type="checkbox"/> Chinese (Trad)	zh	cshmanual	Not available
<input type="checkbox"/> Croatian	hr	css_styled_content	Up to date
<input type="checkbox"/> Czech	cs	dbal	Up to date
<input type="checkbox"/> Danish	da	extbase	Update available
<input type="checkbox"/> Dutch	nl	extdeveval	Up to date
<input type="checkbox"/> Esperanto	eo	extensionmanager	Update available
<input type="checkbox"/> Estonian	et	extra_page_cm_options	Up to date
<input type="checkbox"/> Faroese	fo	feedit	Not available
<input type="checkbox"/> Finnish	fi	felogin	Up to date
<input type="checkbox"/> French	fr	filelist	Not available
<input type="checkbox"/> French (Canada)	fr_CA	fluid	Not available
<input type="checkbox"/> Galician	gl	form	Update available
<input type="checkbox"/> Georgian	ka	frontend	Not available
<input checked="" type="checkbox"/> German	de	func	Not available






























- Das Modul „Backend User“ (beuser) wurde neu auf Basis von Extbase geschrieben und bietet nun neben einer Suche und diversen Filtern (Admin - Nicht Admin, Active - Not Active, Login Before - Never logged in) auch einen Page-Browser und das Aufheben der Beschränkung auf 1000 Datensätze :



The screenshot displays the 'Backend User Administration' interface. At the top, there are tabs for 'Backend users' and 'Online users'. Below the tabs is a 'Filter' section with the following fields:

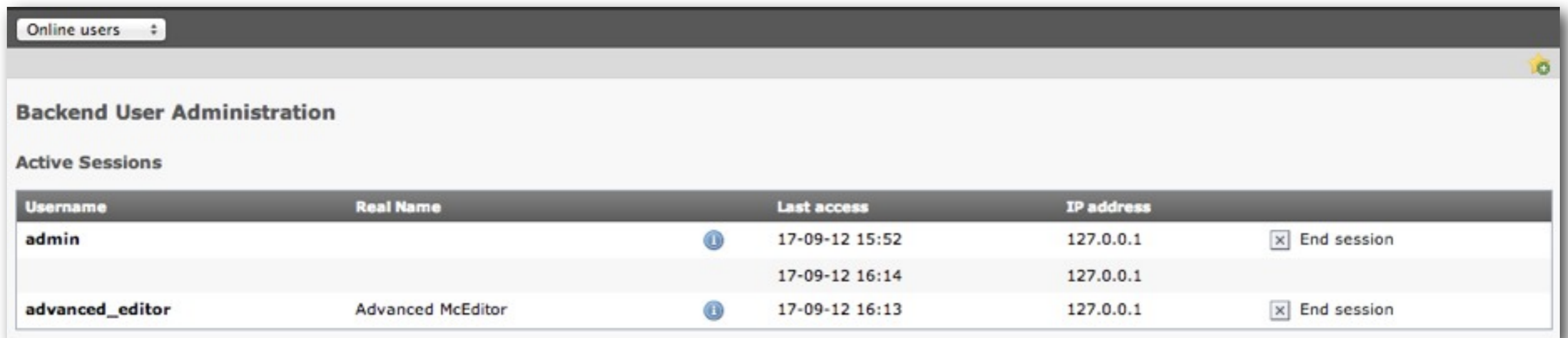
- Username:
- Admin:
- Status:
- Login:
- User group:
- Filter button

Below the filter section is the 'Backend User Listing' table:

Username	Real Name			Last login
 admin		 Compare	  	17-09-12 15:11
 advanced_editor	Advanced McEditor	 Compare	   	Never
 news_editor	News McEditor	 Compare	   	12-12-09 22:24
 simple_editor	Simple McEditor	 Compare	   	Never
 _cli_scheduler		 Compare	   	Never

5 Users

- In der Anzeige der aktiven User („Online User“) kann zudem nun eine User-Session beendet werden, indem man auf den Button bei „End Session klickt“:

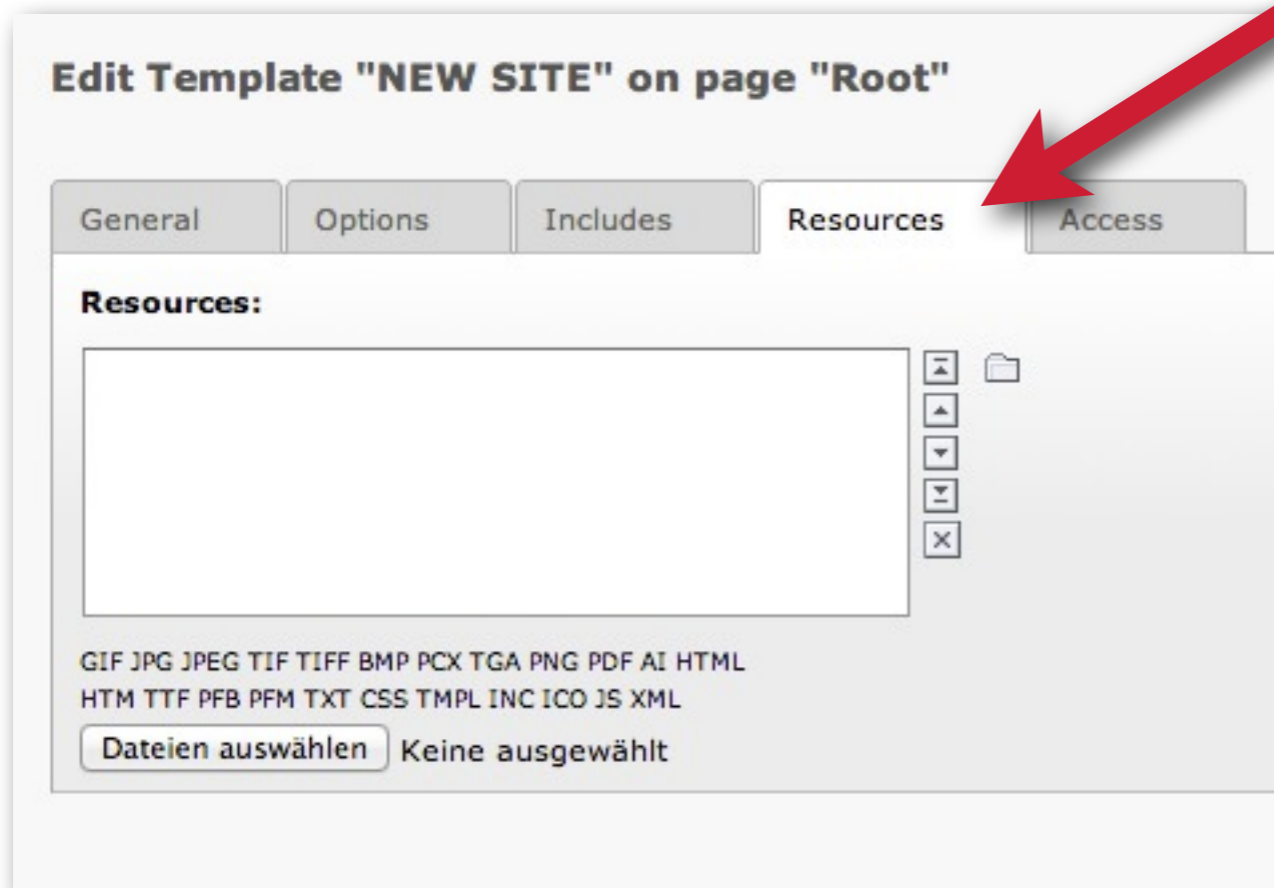


The screenshot shows the 'Online users' interface in the Typo3 backend. It features a table with columns for Username, Real Name, Last access, and IP address. Each row includes an 'End session' button. The 'admin' user has two active sessions, and the 'advanced_editor' user has one active session.

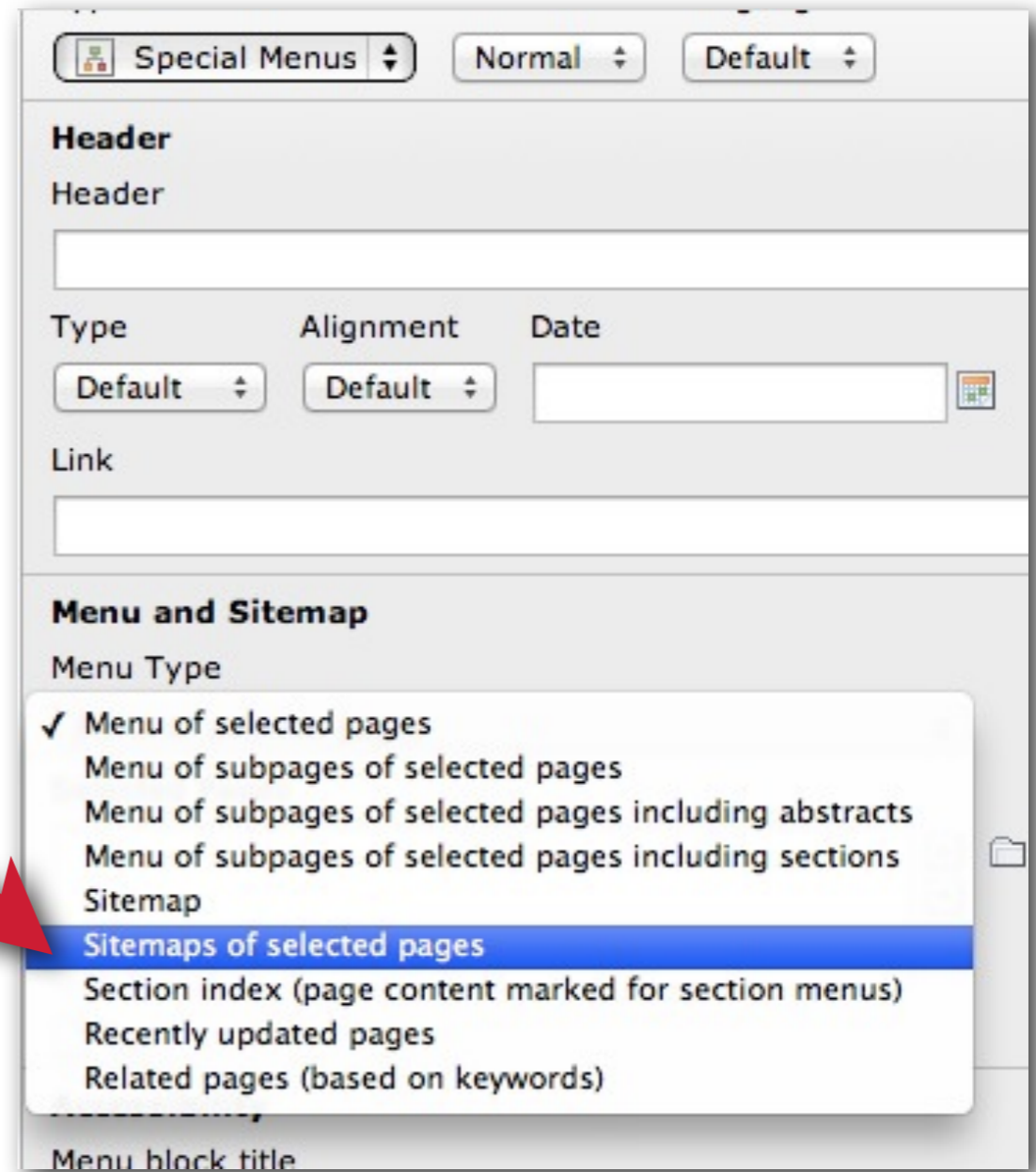
Username	Real Name	Last access	IP address	
admin		17-09-12 15:52	127.0.0.1	<input type="checkbox"/> End session
		17-09-12 16:14	127.0.0.1	
advanced_editor	Advanced McEditor	17-09-12 16:13	127.0.0.1	<input type="checkbox"/> End session

- Im Datensatz „Template“ (TypoScript Template) wurden die „Resources“ entfernt, da diese Funktion praktisch nie produktiv verwendet wurde. Damit konnte man früher Assets wie Bilder an den TypoScript-Datensatz binden.

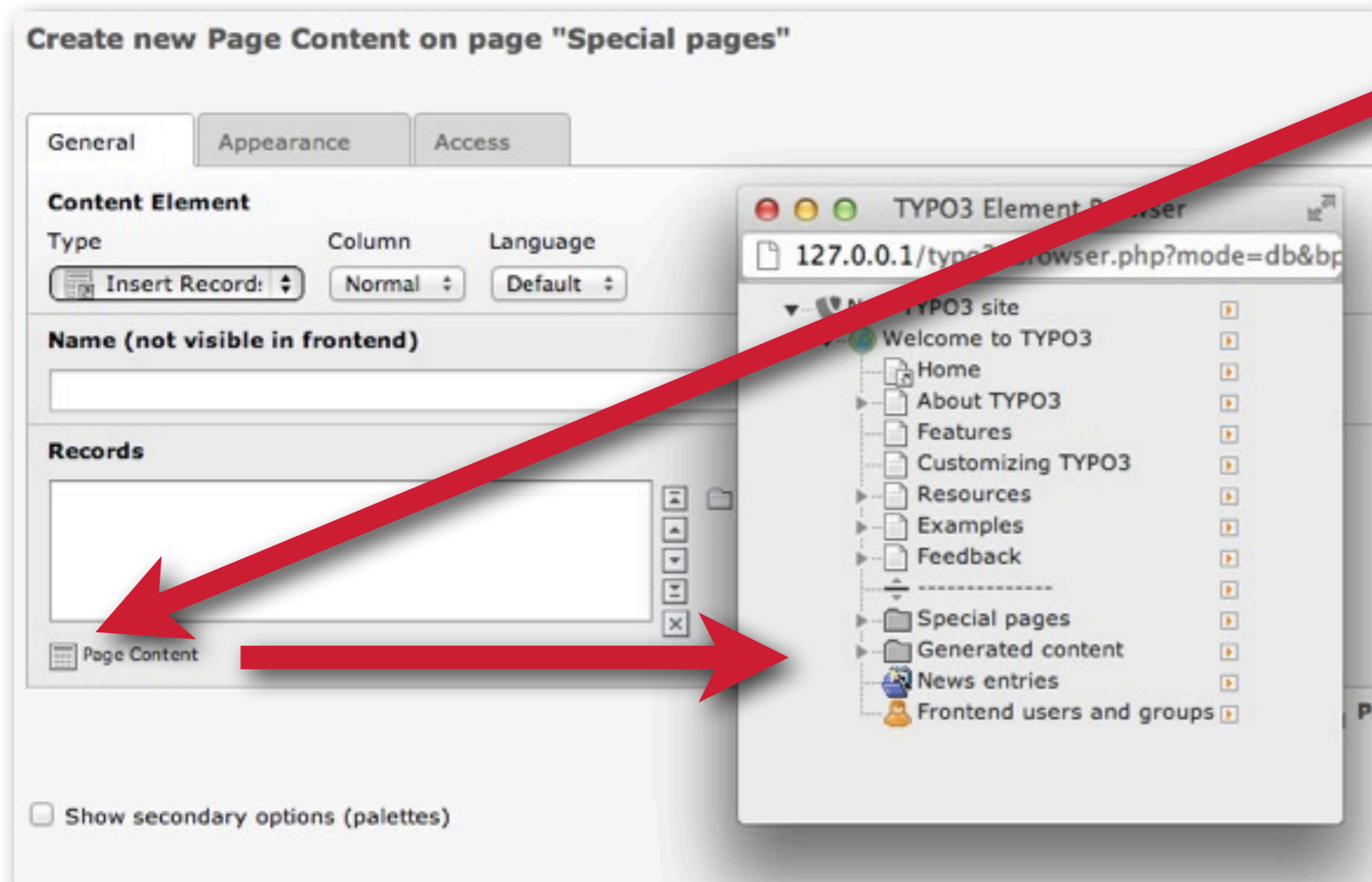
wurde entfernt



- Das Content-Element „Menu/Sitemap“ wurde umbenannt in „Special Menus“
- Dort wurde ein neuer Sitemap-Typ „Sitemaps of selected pages“ eingeführt.
- Ab sofort kann man damit die Root-Seite für die Sitemap auswählen (beim Typ „Sitemap“ war dies immer der Tree-Root).

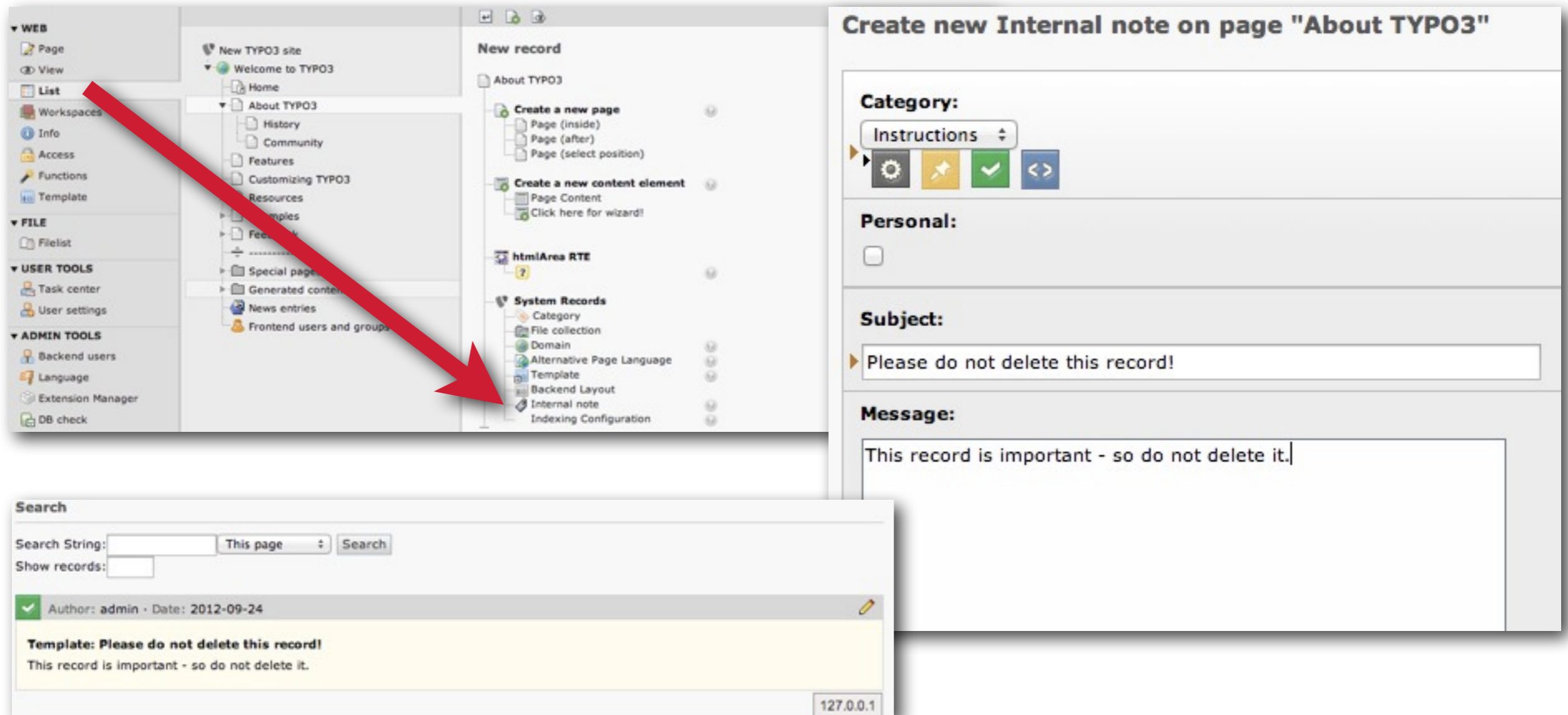


- Die Icons, die im TCA-Typ „group=db“ die zur Verfügung stehenden Record-Typen symbolisieren, sind ab sofort anklickbar und agieren als Filter - d.h. es werden nur Elemente dieses Typs im Element-Browser angezeigt.



klickbar!

- Die „sys notes“ (erreichbar über WEB->List >> Create new record >> Internal note) wurden komplett überarbeitet.



The image displays two screenshots from the TYPO3 CMS interface. The top screenshot shows the 'Create new Internal note on page "About TYPO3"' form. The form includes a 'Category' dropdown menu set to 'Instructions', a 'Personal' checkbox, a 'Subject' field containing 'Please do not delete this record!', and a 'Message' text area containing 'This record is important - so do not delete it.'. A red arrow points from the 'List' menu item in the left sidebar to the 'Internal note' option in the 'System Records' list. The bottom screenshot shows a search result for the same message, with the search string 'Please do not delete this record!' and the message text 'This record is important - so do not delete it.' displayed.

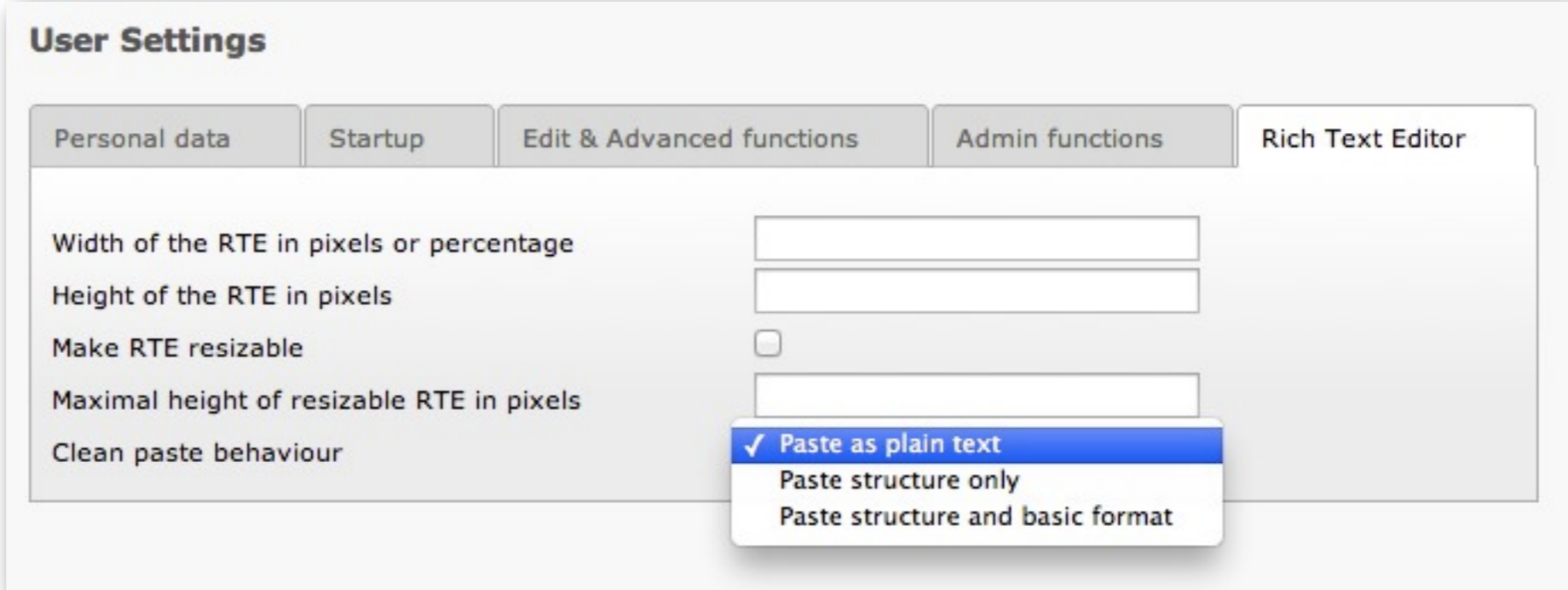
- Bei Klick auf den Titel im Modul List, gelangt man nun in die Edit-Ansicht
- Doctype ist per Default HTML5: `<!DOCTYPE html>`

RTE

Änderungen im Rich Text Editor

- Fügt man aus der Zwischenablage mittels der Funktion **pasteFormat** („Paste structure and basic format“) oder **pasteStructure** („Paste structure only“) Inhalt ein, so werden nun die folgenden Tags zusätzlich erhalten und nicht weggelöscht:

**article, aside,
footer, header,
nav, section**



User Settings

Personal data | Startup | Edit & Advanced functions | Admin functions | **Rich Text Editor**

Width of the RTE in pixels or percentage

Height of the RTE in pixels

Make RTE resizable

Maximal height of resizable RTE in pixels

Clean paste behaviour

- Paste as plain text
- Paste structure only
- Paste structure and basic format

- Die kann über das folgenden PageTSconfig geändert werden:
`buttons.pastebehaviour.[behaviour].keepTags.`

TYPOSCRIPT & TSCONFIG

Neuheiten und Änderungen

- Über das Pipe-Symbol | können nun in Conditions mehrere Werte angegeben und dagegen getestet werden.

z.B.:

```
[globalVar = TSFE:id = 10|12|15] (ist enthalten)
```

```
[globalVar = TSFE:id != 10|12|15] (ist nicht enthalten)
```

- Über die beiden Funktionen **ifNull** und **isNull** kann nun geprüft werden, ob ein Wert NULL ist:

z.B.:

```
10 = TEXT
10 {
    field = description
    stdWrap.ifNull = Keine Beschreibung angegeben!
}
```

```
10 = TEXT
10 {
    if.isNull.field = description
    value = Keine Beschreibung angegeben!
}
```

- Über die Eigenschaft **sectionIndex** in den Menü-Objekten des **HMENU** konnte erreicht werden, dass das Menü nicht auf Unterseiten verlinkt, sondern auf Content-Elemente der aktuellen Seite mittels eines Anker-Links.
- Allerdings wurden bisher lediglich Content-Elemente dort aufgenommen, die sich in der Spalte „Normal“ (colPos = 0) befanden.
- Nun gibt es die Option **useColPos** mit der man die Spalte konkret angeben kann. Über den Wert -1 kann man die Elemente aller Spalten anzeigen. Zudem ist **stdWrap** an dieser Stelle möglich
- z.B.: `tt_content.menu.20.3.1.sectionIndex.useColPos = -1`

- jQuery 1.8 wird nun im Verzeichnis `contrib/jquery/` mitgeliefert
- Die Integration wird noch als „experimentell“ bezeichnet, da eine saubere API hierfür fehlt
- jQuery kann mittels `page.javascriptLibs` direkt gesetzt werden (inkl. Version, Source und Namespace) und wird über den PageRenderer in TYPO3 integriert. Source kann „local“ (dann aus `contrib/jquery`), „msn“, „google“ oder „jquery“ sein - hier werden die CDN der jeweiligen Anbieter verwendet:

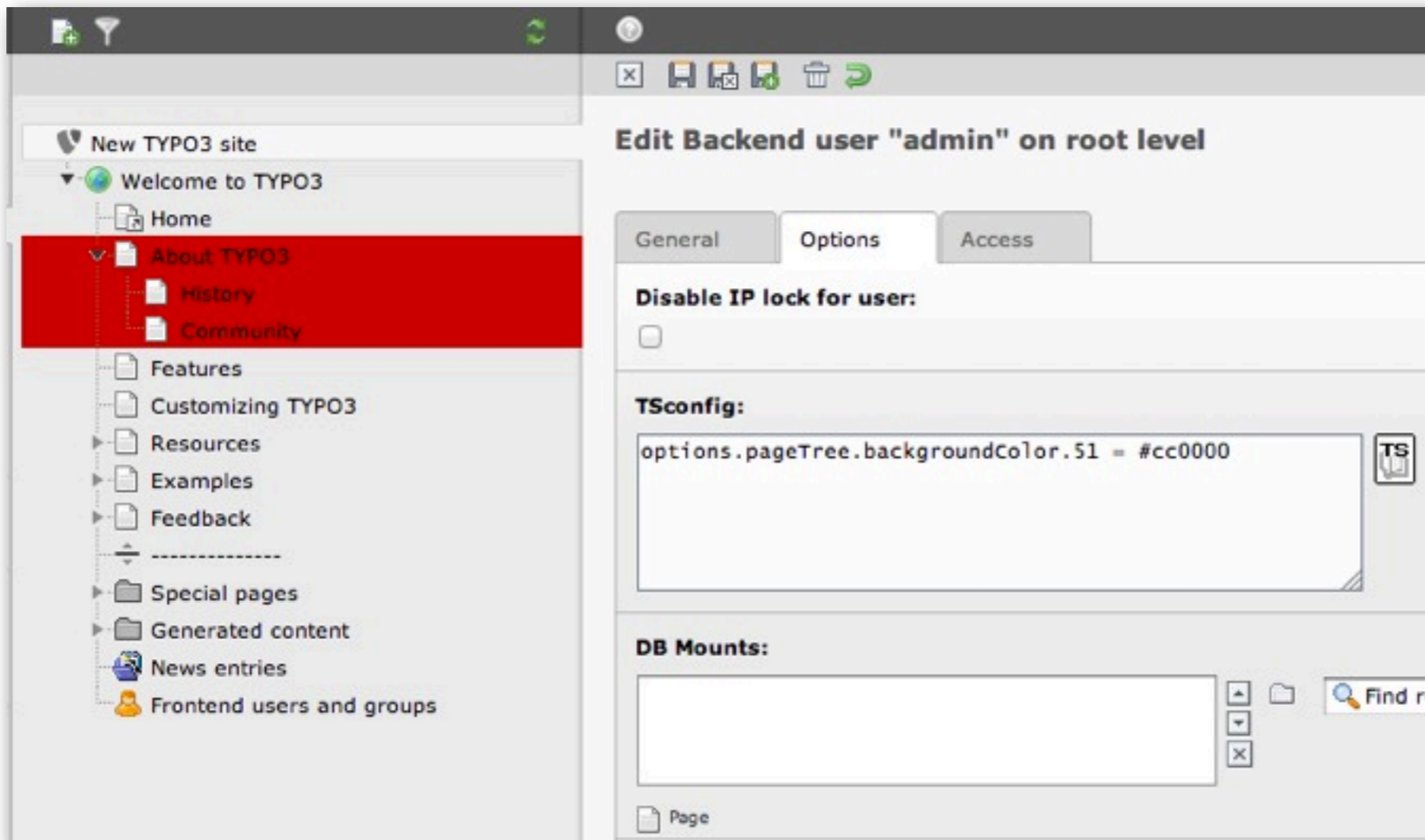
```
page.javascriptLibs {  
  
    jQuery = 1  
    jQuery {  
        version = 1.8.1  
        source = local  
        noConflict = 1  
        noConflict.namespace = test  
    }  
}
```

- Das cObject USER (und USER_INT) hat nun die Eigenschaft stdWrap (mit den gleichnamigen Properties)

```
20 = USER
20 {
    userFunc = tx_cssstyledcontent_pi1->renderSpace
    space = before
    constant = {$content.spaceBefore}
    stdWrap {
        noTrimWrap = |csc-space-before-| |
    }
}
```

- Über UserTSconfig kann nun der Seitenbaum eingefärbt werden

```
options.pageTree.backgroundColor.51 = #cc0000
```



- **Das cObject HTML wurde entfernt.**
An der Stelle wird nun ausschließlich **TEXT** verwendet.

- Beim Rendern von Links wird nun das Attribut

```
onfocus="blurLink(this);
```

weggelassen.

Damit verliert die Option `noBlur` ihre Bedeutung und wurde komplett entfernt.

NAMESPACES

im TYPO3 Core

- **Der gesamte Core ist auf PHP Namespaces umgestellt worden**
(größte Änderung des Cores der Geschichte *g* / ca. 3300 Dateien betroffen!)
- Folgt damit dem PSR-0 (<https://github.com/php-fig/fig-standards>)
- Dafür wurde der Core vorwiegend in drei System-Extension verschoben mit den Namen:
 - **core** (/typo3/sysext/core/): Bootstrap, Logging Functions, Mail- and Caching, TCEmain
 - **backend** (/typo3/sysext/backend/): z.B. Forms und Modules
 - **frontend** (/typo3/sysext/frontend/): z.B. TypoScript und Rendering
 - dort sind im Ordner „Classes/“ alle Klassen separat nach Verwendung enthalten.
- Der Zugriff darauf erfolgt nach dem Schema - z.B.:
 - Datei `Writer.php` im Verzeichnis `typo3/sysext/core/Classes/Log/Writer`
 - Zugriff darauf: `\TYPO3\CMS\Core\Log\Writer\Writer`
- Jede Datei enthält eine Namespace-Deklaration (bis zum Verzeichnis):

```
<?php
namespace TYPO3\CMS\Core\Log;
```

- Die Klasse `ClassAliasMap.php` im Verzeichnis `typo3/sysext/core/Migrations/Code` enthält Mapping-Angaben für den Autoloader (bis voraussichtlich TYPO3 CMS 6.2).

```
<?php
return array(
    'Tx About Controller AboutController' => 'TYPO3\
\CMS\About\Controller\AboutController',
    ...
)
```

- Dort sind > 1690 „alte“ Klassennamen aus Core und System-Extensions samt Mapping auf den Namespace enthalten.
- Extensions können eine eigene `ClassAliasMap` mitliefern - sodaß man einen alten Klassennamen verwenden kann und das intern in PHP auf den neuen Namen umgemapt wird, und sich so der alte Classname verhält wie der neue (sehr nützlich z.B. bei TypeHints).

- **Struktur von Klassenname mit Namespace:**
`{VendorName} \ {PackageName} \ ({CategoryName} \) * {ClassName}`
- Namespaces, class, interface, exception, method, variable, constant und Dateinamen sind in den FLOW3 CGL geregelt: <http://flow3.typo3.org/documentation/codingguidelines/php.html#naming>
- Der TYPO3 Core benutzt `\TYPO3\CMS` als „**Vendor-Name**“. Extension Entwickler dürfen diesen Vendor-Name NICHT in ihren eigenen Extensions verwenden! Verwenden soll man daher den eigenen Firmennamen. Bislang gibt es keine Möglichkeit den Vendor-Namen zentral zu registrieren (so wie beim Extension-Key), daher muss man selbst dafür Sorge tragen, einen eindeutigen Vendor-Namen zu wählen. Der eigene gewählte Vendor (Hersteller) darf aus nur einem Bestandteil bestehen. Gültige Vendor für Extensions wären beispielsweise `\Typovision`, `\Enet` oder `\DKD`. Vorher sollte unbedingt überprüft werden, ob der Vendor eindeutig ist (z.B. Web-Recherche oder bei github nachsehen. Domain- oder Markennamen sind beispielsweise auch gute Vendor-Namen (ohne Punkte und TLDs).

- Der **zweite Teil des Namespaces** (im Falle von Core-Extensions der dritte Teil) wird als **Extension-Key** behandelt. Dank des Vendor-Names kann nun ein beliebiger Extension-Key gewählt werden. Der Extension-Key Teil im Namespace muss mit der UpperCamelCase-Schreibweise notiert werden und wird intern zu „kleinschreibweise_mit_unterstrich“ transformiert.
- Mit TYPO3 CMS 6.0 kann man also noch nicht z.B. \Enet\Shop und \Dkd\Shop gleichzeitig laden, da beide Extensions als „shop“ im Verzeichnis liegen würden.
- **Weitere Teile des Namespaces** - sofern verwendet - müssen die **Kategorie** beschrieben, zu der die Klasse gehört. Während man durch den Klassenbaum browsst, gibt so der Kategorie-Name einen guten Eindruck davon, zu welchem Thema die Klasse gehört.
- Klassennamen, welche in Strings verwendet werden, müssen vollständig und mit Escaping angegeben werden um Fehler zu vermeinden (siehe <http://www.php.net/manual/en/language.namespaces.faq.php#language.namespaces.faq.quote>). Die führenden Slashes müssen dabei weggelassen werden
`$configurationManager = $objectManager->get('TYPO3\CMS\Extbase`

- In Extensions sollte **include()** und/oder **require()** **NICHT** mehr verwendet werden.
- Anstellen dessen sollte man die **Autoloader**-Funktionalität verwenden. Dazu legt man eine Datei `ext_autoload.php` innerhalb des Extension-Verzeichnisses an und referenziert dort entsprechend alle Dateien. Halten sich die Extensions an die Extbase Dateistruktur, dann ist auch die `ext_autoload.php` überflüssig - im Core sieht man dies z.B. bei „lang“ und „extensionmanager“.
- Wenn die Klasse dem Extbase-Namensschema entspricht, dann ist der entsprechend Eintrag in `ext_autoload.php` nicht notwendig. So wird die Klasse `\Tx\ExtName\Foo\Bar` automatisch aufgelöst, wenn sich die Klasse in der Datei `ExtName/Classes/Foo/Bar.php` befindet.
- Bei **Extbase Backend-Modulen** muss man dem **Extension-Key den Vendor-Namen voran stellen:**

```
\TYPO3\CMS\Extbase\Utility
\ExtensionUtility::registerModule (
    'Typovision.' . $_EXTKEY,
    ...
);
```

LOGGING API

Einführung

- Logging in TYPO3 wurde bislang an vielen Stellen individuell durchgeführt und daher nun zu einer gemeinsamen API - der Logging API - zusammengeführt. Die API wird im Core derzeit noch NICHT verwendet.
- **TYPO3 Logging besteht aus den folgenden Teilen:**
 - Ein „**Logger**“, die die Log-Message und die zugehörigen Details (wie z.B. die Wichtigkeit) empfängt
 - Die Konfiguration des Logging-Systems
 - „**Writers**“ die die Log-Einträge in verschiedene Ziel-Systeme schreiben (z.B. Datei, Datenbank, rsyslog Server, u.s.w.)
 - „**Processors**“ welche detailliertere Informationen zum Log-Eintrag hinzufügen.

- **Beispiel für die Logging API:**

```
<?php

/** @var $logger \TYPO3\CMS\Core\Log\Logger */
$loggerInstance = \TYPO3\CMS\Core\Utility
\GeneralUtility::makeInstance('TYPO3\CMS\Core\Log\LogManager');
$logger = $loggerInstance->getLogger(__CLASS__);

// Logging
$logger->info('everything went fine, puhh');
$logger->warning('A warning!');

// Zusätzliche Informationen hinzufügen
$logger->error(
    'This was not a good idea',
    array(
        'param1' => $param1,
        'param2' => $param2,
    )
);

// Die Hauptmethode ist „log“. Die anderen sind nur Abkürzungen hierzu.
$logger->log(TYPO3\CMS\Core\Log\LogLevel::WARNING, 'Another warning');
```


- **Die Methode log():**

```
/**
 * Adds a log record.
 *
 * @param integer $level Log level.
 * @param string $message Log message.
 * @param array $data Additional data to log
 * @return mixed
 */
public function log($level, $message, array $data = array()) {
```

- **Es gibt folgende Log Levels:**

```
TYPO3\CMS\Core\Log\LogLevel::EMERGENCY
TYPO3\CMS\Core\Log\LogLevel::ALERT
TYPO3\CMS\Core\Log\LogLevel::CRITICAL
TYPO3\CMS\Core\Log\LogLevel::ERROR
TYPO3\CMS\Core\Log\LogLevel::WARNING
TYPO3\CMS\Core\Log\LogLevel::NOTICE
TYPO3\CMS\Core\Log\LogLevel::INFO
TYPO3\CMS\Core\Log\LogLevel::DEBUG
```

API

Änderung in den Programmier-Interfaces

- In der Klasse `\TYPO3\CMS\Core\Utility\ArrayUtility` gibt es nun eine statische Methode

```
public static function flatten(array $array, $prefix = '')
```

mit der es möglich ist, multidimensionale Arrays zu „verflachen“.

Aus

```
array('first.' => array('second' => 1))
```

wird demnach

```
array('first.second' => 1)
```

- Weitere neue Methoden in dieser Klasse sind `getValueByPath(array $array, $path, $delimiter = '/')` und `setValueByPath(array $array, $path, $value, $delimiter = '/')`

- Über die Methode

```
\TYPO3\CMS\Backend\Utility
```

```
\BackendUtility::getLabelFromItemList($table, $col, $key)
```

ist es möglich, das Label eines TCA-Feldes im Backend zu erhalten. Bislang wurden aber immer Änderungen ignoriert, die über folgendes TSconfig gemacht wurden:

```
TCEFORM.<table>.<field>.addItem.<key> = New value or
```

```
TCEFORM.<table>.<field>.altLabels.<key> = Change me
```

- Das wurde nun geändert.

- Über die Methode

`TYPO3\CMS\Core\Utility\MathUtility::isIntegerInRange($value, $minimum, $maximum)`

kann überprüft werden, ob eine Zahl `$value` ein Integer-Wert ist und sich innerhalb von `$minimum` und `$maximum` befindet.

- In User Functions

```
\TYPO3\CMS\Core\Utility  
\GeneralUtility::callUserFunction($funcName, &$params, &  
$ref, $checkPrefix = '', $errorMode = 0)
```

kann man nun ab sofort Closures (<http://php.net/manual/de/functions.anonymous.php>) verwenden.

```
/* @param string $funcName Function/Method reference or Closure, '[file-  
reference":']["&"]class/function["->"method-name]'. You can prefix this  
reference with "[file-reference]:" and t3lib_div::getFileAbsFileName() will  
then be used to resolve the ...
```

```
public static function callUserFunction($funcName, &$params, &$ref,  
$checkPrefix = '', $errorMode = 0) {
```

- Das Caching-Framework wurde auf den letzten Stand der korrespondierenden TYPO3 Flow Version gebracht:
 - Neues Interface: TaggableBackendInterface (welches Tagging unterstützt)
 - Und ein neues SimpleFileBackend wurde zugefügt (dies unterstützt kein Tagging, daher ist das Flushing deutlich schneller)
 - Das Cache-Backend kann nun eingefroren werden (das FileBackend unterstützt dies beispielsweise)

INTERNA

Interne Änderungen

- **Der Bootstrap (index.php) wurde komplett überarbeitet:**

Datei index.php:

```
require 'typo3/sysex/core/Classes/Core/Bootstrap.php';
\TYPO3\CMS\Core\Core\Bootstrap::getInstance()->baseSetup('');
require(PATH_tslib . 'index_ts.php');
```

Datei typo3/sysex/cms/tslib/index_ts.php (kann sich noch leicht ändern):

```
\TYPO3\CMS\Core\Core\Bootstrap::getInstance()
->registerExtDirectComponents()
->populateLocalConfiguration()
->initializeCachingFramework()
->registerAutoloader()
->checkUtf8DatabaseSettingsOrDie()
->transferDeprecatedCurlSettings()
->setCacheHashOptions()
->enforceCorrectProxyAuthScheme()
->setDefaultTimezone()
->initializeL10nLocales()
...
```

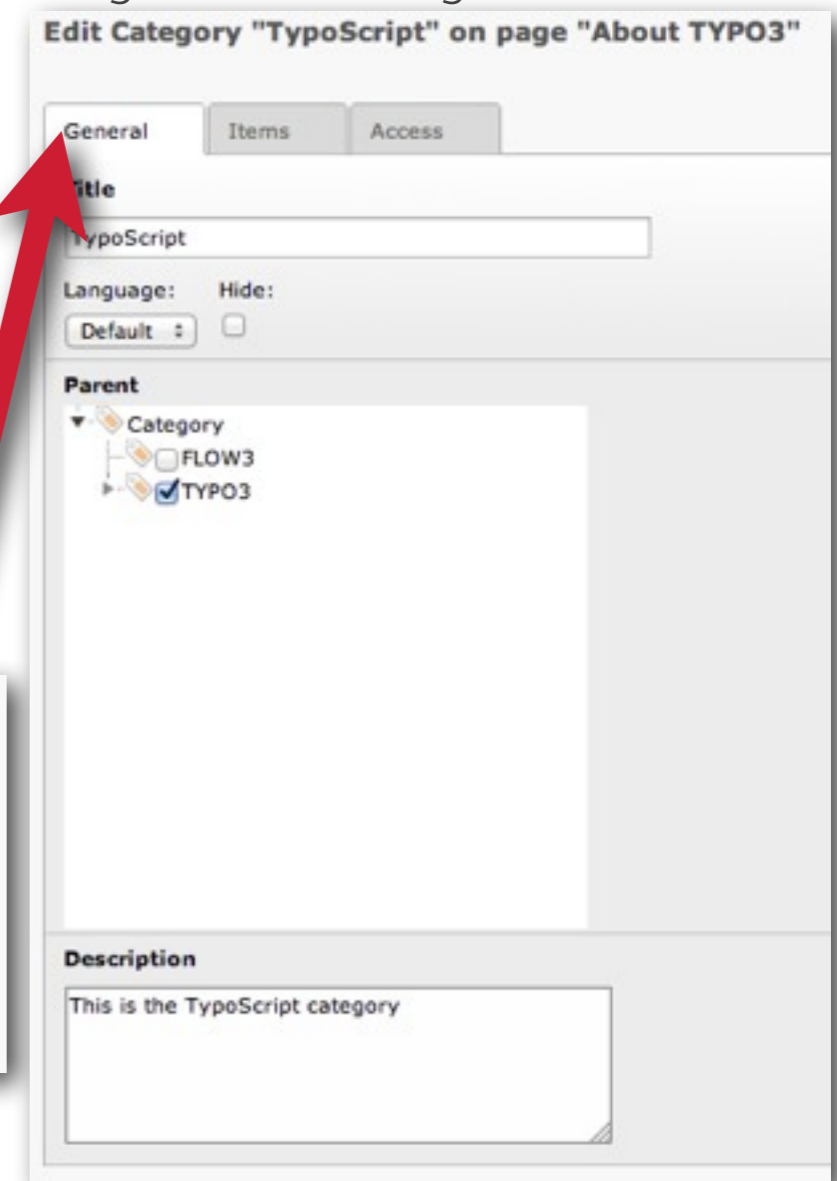
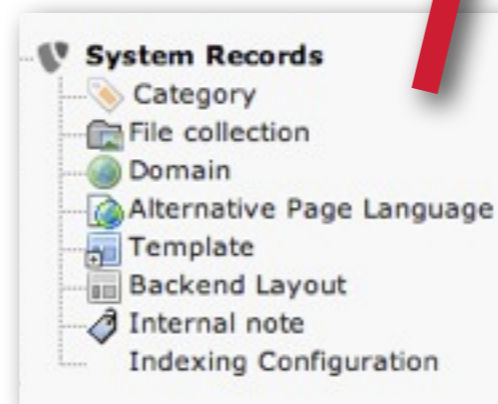
- **Der Core wurde nun mit einer Category-API ausgestattet**
- Damit ist es möglich, generische Kategorien Extension-übergreifend zu verwenden. Diese können von jeder Extension genutzt werden. Damit braucht nicht jeder Entwickler sein eigenes Handling für Kategorien zu erstellen.

- Ausrüsten kann man als Extension-Entwickler seine Extension mit der Möglichkeit Kategorien zu verwenden, wie folgt:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::  
makeCategorizable($extensionKey,  
$tableName, $fieldName = 'categories',  
$options = array());
```

- Anlegen der Kategorien kann man im BE als Systemdatensatz.
- Genutzt wird die Category-API allerdings im Core noch nicht.
- Zuständige Tabellen sind:

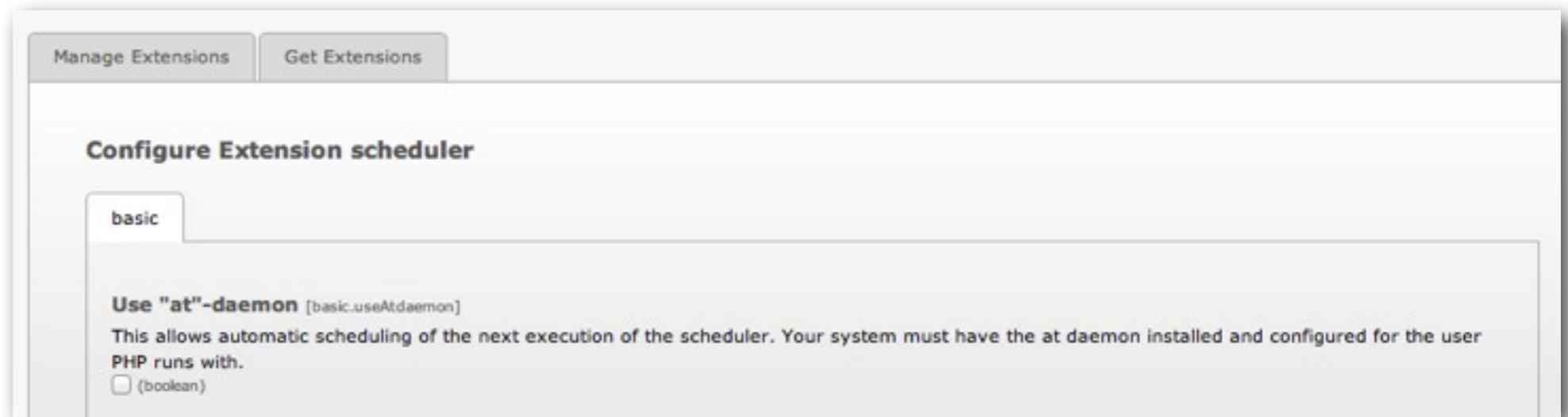
```
sys_category  
sys_category_record_mm
```



- Das Page-Modul basiert per Default nun nicht mehr auf der Config von mod.SHARED.colPos_list, sondern auf einem Default-Backend-Layout:

- ```
backend layout {
 colCount = 4
 rowCount = 1
 rows {
 1 {
 columns {
 1 {
 name = LLL:EXT:cms/loclang_ttc.xlf:colPos.I.0
 colPos = 1
 }
 2 {
 name = LLL:EXT:cms/loclang_ttc.xlf:colPos.I.1
 colPos = 0
 }
 3 {
 name = LLL:EXT:cms/loclang_ttc.xlf:colPos.I.2
 colPos = 2
 }
 4 {
 name = LLL:EXT:cms/loclang_ttc.xlf:colPos.I.3
 colPos = 3
 }
 }
 }
 }
}
```

- Während der sonst benutzte **Cron-Job** beim Scheduler regelmäßig aufgerufen wird (auch wenn eventuell keine Ausführung des dahinterliegenden Jobs notwendig wäre), kann man ab sofort auch den sogenannte **AT-Daemon** verwenden. Dieser wird einmalig am angegebenen Zeitpunkt ausgeführt.
- Dies ist insbesondere auch in Umgebungen notwendig, wo kein Zugriff auf die Crontab möglich ist, aber ein AT-Daemon (atd) läuft.
- Einschaltbar über die Extension-Config:



## Dabei wird wie folgt vorgegangen:

1. Herausfinden, wann der nächste Task ausgeführt werden muss.
2. Wenn der nächste Task erst in (z.B.) 60 Minuten ausgeführt werden müsste, gibt es keinen Grund, warum der Scheduler alle 5 Minuten gestartet würde
3. Wenn der letzte reguläre Job deaktiviert ist, führe den Scheduler über den AT-Deamon (atd) nicht erneut aus
4. Wenn das Intervall für den Job geändert wird, wird die nächste Ausführung erneut durch den atd bestimmt.
5. Während der Ausführung sorgt der Scheduler automatisch dafür, den nächsten benötigten Durchlauf zu ermitteln.

**Es ist nun möglich, einen Scheduler-Task über die Kommandozeile auszuführen:**

```
/typo3/cli_dispatch.phpsh scheduler -i <uid>
```

Die <uid> entspricht der ID des Scheduler-Tasks den man ausführen möchte.

Über den Parameter -f (force) kann festgelegt werden, dass sogar deaktivierte Tasks und/oder Tasks ohne bevorstehende Terminierung ausgeführt werden können.

Ab nun könnten CLI-Skripte im Core durch Tasks ersetzt werden.

Ab sofort kann das TYPO3-Backend auch dedizierte NULL-Werte speichern (vorher konnte man nur den leeren String bzw. die Zahl 0 als solche speichern):

```
'columns' => array(
 'title' => array(
 'config' => array(
 'type' => 'text',
 'eval' => 'null',
)
)
)
```

Dies war initial notwendig geworden, da man mittels FAL überprüfen will, ob z.B. die Beschreibung die mittels Overlay ermittelt wird, wirklich leer ist, oder vererbt wird (dann müsste dort NULL drin stehen).

- Einige Felder in der Datenbank sind nun vom Typ „int“ (und nicht mehr tinyint), da man ansonsten nur Werte von 0-255 dort ablegen kann, man aber unter Umständen mehr Werte benötigt - gerade bei großen Webseiten.
- Zudem kann man damit Werte abspeichern, die nicht mehr kollidieren können (ähnlich den Timestamps bei Extbase Exception Codes).
- Dies betrifft die Felder:
  - doktype (Tabelle pages und pages\_language\_overlay)
  - layout (Tabelle pages und pages\_language\_overlay)
  - section\_frame (Tabelle tt\_content)
  - image\_frames (Tabelle tt\_content)
  
  - snapshot (Tabelle pages) (neu in TYPO3 6.0)
  - cols (Tabelle pages\_language\_overlay) (neu in TYPO3 6.0)
  - colPos (Tabelle pages\_language\_overlay) (neu in TYPO3 6.0)
  - table\_bgColor (Tabelle pages\_language\_overlay) (neu in TYPO3 6.0)



- **Neue Eigenschaft für**  
**`['columns'][field name]['config'] / TYPE: "group" / "inline":`**
- **filter (Datentyp array)**
- Dies ist nützlich, wenn im Group-Feld nur bestimmte „foreign records“ erlaubt sind. Die Werte werden sowohl im Element-Browser wie auch bei der Abarbeitung von TCEMain gefiltert.
- Filter-UserFuncs sollten zwei Input-Parameter haben: `$parameters` und `$parentObject`. Das erste Argument (`$parameters`) ist ein Array mit den Parametern des TCA aber auch mit einem zusätzlichen Parameter "values", welcher das Array der Werte beinhaltet, die durch die UserFunc gefiltert werden. Die Funktion gibt das gefilterte Array der Werte zurück.
- Es können mehrere Filter gleichzeitig definiert werden.

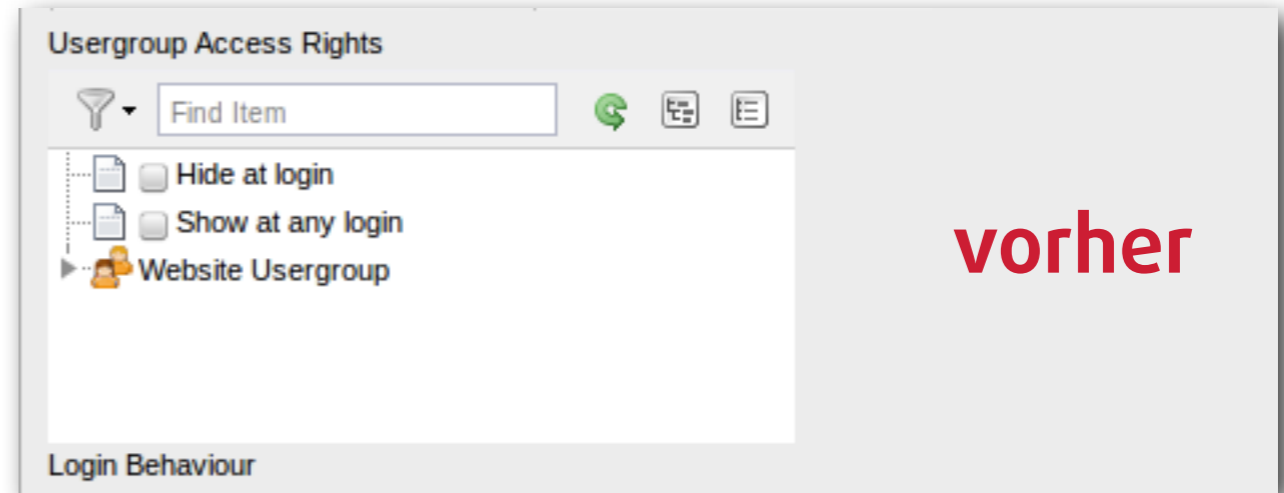
- Beispiel**

```
// Bekommt nur Datensätze angezeigt, deren Geschlecht auf „female“ steht
'mother' => array (
 'label' => 'Mother',
 'config' => array (
 'type' => 'group',
 'internal_type' => 'db',
 'allowed' => 'tx_myext_person',
 'size' => 1,
 'filter' => array (
 'userFunc' => 'EXT:myext/
class.tx_myext_filter.php:tx_myext_filter->doFilter',
 'parameters' => array(
 'evaluateGender' => 'female',
),
),
),
),
),
)

// Die korrespondierende Filter-Klasse könnte wie folgt aussehen
class tx_myext_filter {
 public function doFilter(array $parameters, $parentObject) {
 $fieldValues = $parameters['values'];
 // Filter-Algorithmus kommt hier
 ...
 return $fieldValues;
 }
}
```

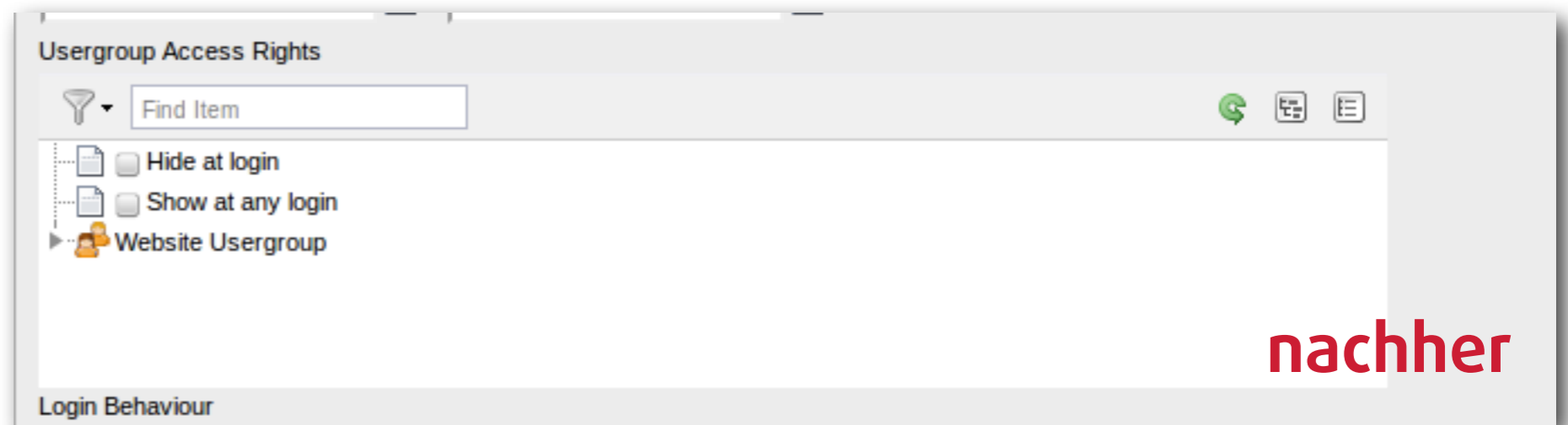
# BREITE DES TCA-TREES

Man kann nun die Breite eines gerenderten TCA-Trees festlegen (der Default-Wert ist 280px)



**vorher**

```
['TCA']['pages']...['config']['treeConfig']['appearance']['width']
```



**nachher**

Verzeichnisse werden vom Introduction Package mit der Unix-Berechtigung 2777 und die Dateien mit 0666 angelegt.

Besser wäre allerdings 2770 (nicht schreibbar für „others“) und 0660 (ebefalls nicht schreibbar für „others“) - dies wird nun als Warnung im Reportsmodul angezeigt.

▼  **Configuration**

|                                                                                                                                                                                                                                                                                                                                                                                          |             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
|  <b>Permissions of created directories</b>                                                                                                                                                                                                                                                             | <b>2777</b> |
| Directories created by TYPO3 are configured to be world writeable. Depending on your server configuration, this can be a security risk. It is usually better to configure the create mask to not allow writing to directories by "others". A sane default is often '2770' for <code>\$GLOBALS['TYPO3_CONF_VARS']['BE']['folderCreateMask']</code> . This can be set in the install tool. |             |
|  <b>Permissions of created files</b>                                                                                                                                                                                                                                                                   | <b>0666</b> |
| Files created by TYPO3 are configured to be world writeable. Depending on your server configuration, this can be a security risk. It is usually better to configure the create mask to not allow writing to files by "others". A sane default is often '0660' for <code>\$GLOBALS['TYPO3_CONF_VARS']['BE']['fileCreateMask']</code> . This can be set in the install tool.               |             |

- Normalerweise speichert TYPO3 bei Zeit/Datum-Angaben den Unix-Timestamp in der Datenbank. Dies kann nun geändert werden auf „date“ bzw. „datetime“ sofern gewünscht:

```
'my_native_date' => array(
 'exclude' => 0,
 'label' => 'My native date',
 'config' => array(
 'dbType' => 'date', // Möglich ist: date und datetime
 'type' => 'input',
 'size' => '8',
 'max' => '20',
 'eval' => 'date',
 'checkbox' => '0',
 'default' => '0'
)
) ,
```

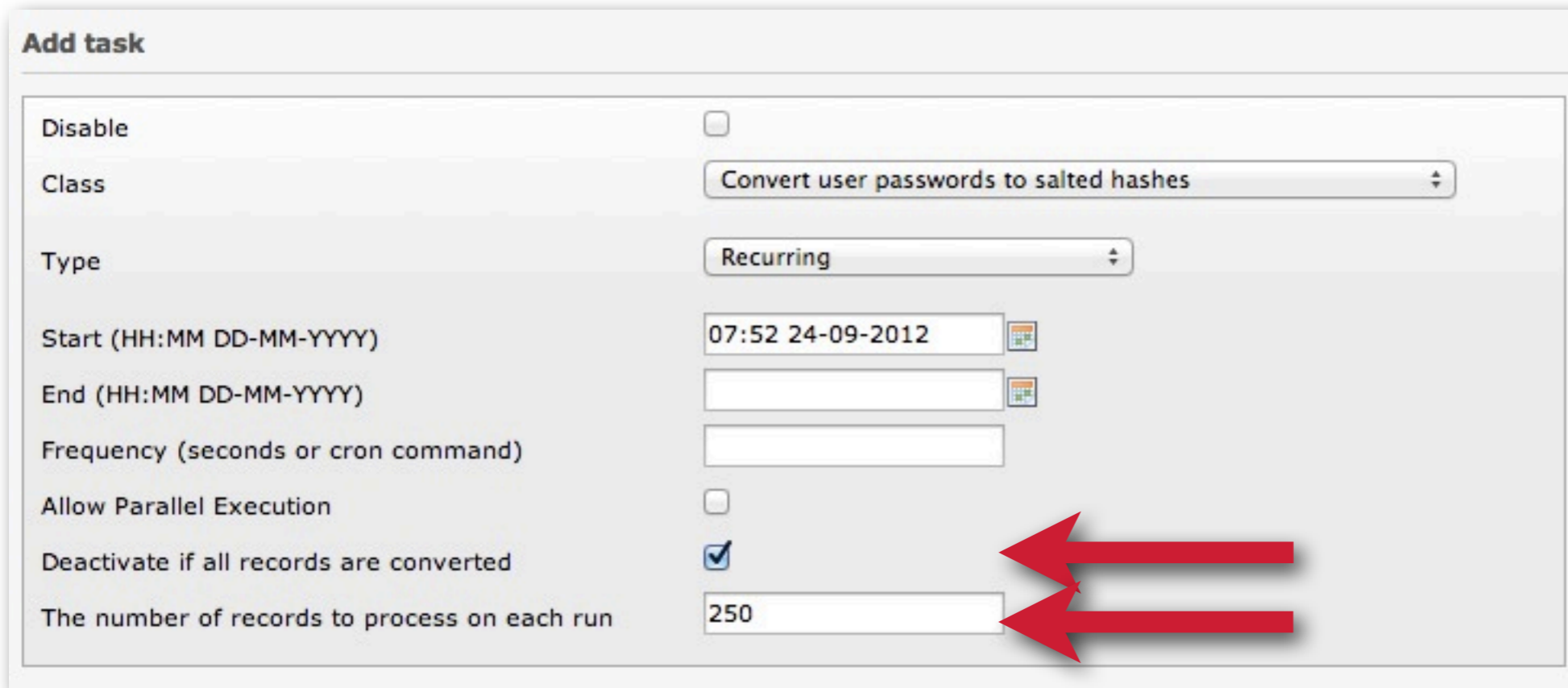
- Es ist nun möglich bei bestimmten TCA-Typen auf einzelne Felder das readOnly-Flag (keine Bearbeitung im Backend möglich) via PageTSconfig zu setzen.
- Dies betrifft die Typen input, text, select, group und inline
- Das Setzen kann nun wie folgt geschehen:

```
TCEFORM.[table].[field].config.readOnly = 1
```

- Normalerweise können nur Admin-User Datensätze von der Seite mit der ID = 0 (Root-Level) lesen
- Zudem müssen sich „normale“ BE-User an die eigestellten WebMounts halten - sehen also nur Datensätze von den dort konfigurierten Seiten
- Dies kann nun mit der Eigenschaft **security** umkonfiguriert werden:

```
$TCA['sys_file'] = array(
 'ctrl' => array(
 ...
 'security' => array(
 'ignoreWebMountRestriction' => 1,
 'ignoreRootLevelRestriction' => 1,
),
 ...
)
);
```

- Beim Scheduler Task „Convert user passwords to salted hashes“ ist es nun möglich den Task per Einstellung zu deaktivieren, sobald er alle Passwörter (erfolgreich) konvertiert hat.



**Add task**

|                                              |                                         |
|----------------------------------------------|-----------------------------------------|
| Disable                                      | <input type="checkbox"/>                |
| Class                                        | Convert user passwords to salted hashes |
| Type                                         | Recurring                               |
| Start (HH:MM DD-MM-YYYY)                     | 07:52 24-09-2012                        |
| End (HH:MM DD-MM-YYYY)                       |                                         |
| Frequency (seconds or cron command)          |                                         |
| Allow Parallel Execution                     | <input type="checkbox"/>                |
| Deactivate if all records are converted      | <input checked="" type="checkbox"/>     |
| The number of records to process on each run | 250                                     |



- **Ab sofort gibt es einen neuen Mechanismus um XCLASSES zu laden:**

- **ext\_autoload.php:**

```
<?php
...
$extensionPath = \TYPO3\CMS\Core\Extension
\ExtensionManager::extPath('my_extension');
return array(
 'ux_t3lib_userauth' => $extensionPath . 'xclasses/
class.ux_t3lib_userauth.php',
);
?>
```

- Dies weist den Autoloader an, nach einer XCLASS mit dem Namen t3lib\_userauth in der Datei "path-to-extension/xclasses/class.ux\_t3lib\_userauth.php" zu suchen.
- (allerdings sollte man grundsätzlich - sofern möglich - auf XCLASS zu Gunsten von Hooks oder Signal/Slots verzichten)

- Sollte es beim **BE-Login** einen **Fehler** geben, wird nun ein Fehler mit der **Severity 2 (Warning)** und nicht mehr 0 (Info) geloggt. Dies ist insbesondere für Tools wie „fail2ban“ wichtig.
- Das **BE-Login Popup** hat zwei neue Klassen „**error**“ und „**refresh**“ bekommen, mit denen man die beiden Fälle entsprechend stylen kann.
- Für jedes Laden einer Klasse via `\TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance()` wird überprüft, ob es eine Klasse mit dem Namen „ux\_klassenname“ gibt und dann diese Klasse anstelle geladen - dies ist der (neue) XCLASS-Mechanismus von TYPO3. Gibt es diese allerdings nicht, wird dies nun im Autoloader Cache File vermerkt (Caching) und sorgt so für einen Performanceschub.

- Der **Suggest-Wizzard** (TCA) sucht bislang ausschließlich in den Feldern „label“ und „label\_alt“. Nun kann über die Option „additionalSearchFields“ **in weiteren Feldern** gesucht werden.
- Die **Cache-Dateien** unter `typo3conf/temp_CACHED_xxx.php` **existieren ab sofort nicht mehr**. Der Cache wird über das Caching-Framework geschrieben und befinden sich hier: `typo3temp/Cache/`  
Damit wird `TYPO3_CONF_VARS['SYS']['extCache']` überflüssig
- Alle wichtigen **Bootstrap** relevanten **Cache Dateien** sind jetzt in `typo3temp/Cache/Code/cache_core` zu finden und können bei Problemen entsprechend in einem Rutsch gelöscht werden. Die extCache Setting im install Tool sind nun endgültig entfernt.
- **In** den Dateien `ext_tables.php` und `ext_localconf.php` muss man nun (wegen der Einführung des neuen Bootstrapper) auf globale Variablen auch **global zugreifen**, d.h. z.B. das `$TCA` erreicht man über `$GLOBALS['TCA']`.
- PHP: `register_globals` muss nun auf **off** stehen, sonst bricht der Bootstrapper seine Arbeit ab.

- Die Systemextension „**form**“ kann nun individuell **gestyled** werden - dafür wurde der folgende TypoScript-Schlüssel eingeführt:  
`plugin.tx_form._CSS_DEFAULT_STYLE`  
Von nun an wird auch `config.removeDefaultCss` respektiert.
- Auf **Windows-Server** steht nun das **Caching-Backend** „wincache“ (`t3lib_cache_backend_WincacheBackend`) zur Verfügung (wenn die gleichnamige PHP-Extension geladen wurde).
- **IRRE** Felder innerhalb von **FlexForms** sind nun möglich
- Die Länge des Feldes **CType** innerhalb der Tabelle **tt\_content** wurde nun von 30 auf 255 Zeichen erhöht

- **Der Core prüft** an verschiedenen Stellen, ob Klassennamen bestimmte **Prefixe** haben (beispielsweise bei User Functions) - so wird hier nur `tx_`, `Tx_`, `t3lib_`, `user_` oder `User_` als Prefix zugelassen. **Diese Einschränkung wurde aufgehoben** - der Prefix muss mindestens 2 Zeichen lang sein und darf nicht „Core“ lauten. Zudem ist die Install Tool Einstellung `[sys]` `[additionalAllowedClassPrefixes]` entfernt worden.
- Ab sofort ist es möglich sowohl `ext_icon.gif` als auch **`ext_icon.png`** (neu) zu verwenden um das Icon für eine Extension festzulegen.
- Weiteres wurden **entfernt**: `$GLOBALS['TYPO_VERSION']`, `t3lib_div::intInRange()`
- **Entfernung** der **TypoScript-Optionen**: `hover`, `hoverStyle`, `noLinkUnderline`, `smallFormFields`, `minifyJS`, `minifyCSS`
- **Entfernung** der **PHP\_SCRIPT\*** TypoScript cObjects

- Es wurde **Post Prozessoren** (PostProcessors )beim **FORM** cObject eingeführt
- PostProcessors müssen nun das Interface  
`\TYPO3\CMS\Form\PostProcess\PostProcessorInterface`  
implementieren, und können damit per TS geladen werden
- Das **Manual** des Form Wizards wurde um den Abschnitt „Postprocessors“ (Verarbeitung der Formulardaten nach dem Abschicken des Formulars) ergänzt. Das Manual befindet sich unter `typo3/sysex/ext/form/Documentation/en/manual.sxw`

# EXTBASE & FLUID

Neue Funktionen

- Bislang konnte man nur entweder alle Datensätze (sogar gelöschte) oder eben jene die nicht versteckt, innerhalb von Start/Stop liegen und nicht gelöscht sind, aus dem Repository auslesen. Damit man alle bekommt, musste man `$querySettings->setEnableFields(FALSE)`; im Query Manager setzen.
- **Nun gibt es folgende neue Optionen**
  - **setIgnoreEnableFields**  
Bei TRUE werden alle „enable fields“ ignoriert. Wenn zusätzlich die Option `enableFieldsToBeIgnored` gesetzt ist, werden nur diese dort gesetzten Felder ignoriert. Wenn FALSE werden alle enable fields berücksichtigt, unabhängig von `enableFieldsToBeIgnored`.
  - **setEnableFieldsToBeIgnored**  
Setzt ein Array aus zu ignorierenden „enable fields“ welche in `$GLOBALS['TCA'][$table]['ctrl']['enablecolumns']` definiert sein müssen.
  - **setIncludeDeleted**  
Flag, ob gelöschte Dateien im Query enthalten sein sollen.
- Zudem gibt es die TS-Option: `config.tx_extbase.features.ignoreAllEnableFieldsInBe` welche angibt, ob die „enable fields“ im BE ignoriert werden sollen oder nicht.



- Es wurden **Models** für folgende File Abstraction Layer (FAL) Objekte eingeführt:
  - **File**
  - **FileReference**
  - **Folder**
  - **FolderBasedFileCollection**
  - **StaticFileCollection**
- Zusätzlich wurden entsprechende Type-Konverter zugefügt
- **Dieses Feature ist ausdrücklich als „Experimental“ gekennzeichnet!**
  - \* `@api experimental! This class is experimental and subject to change!`

- Der Bootstrapper **Tx\_Extbase\_Core\_Bootstrap** kann nun via **Dependency Injection** (DI) überschrieben werden
- Die **Persistence-Schicht** wurde durch die Einführung von **Caches** beschleunigt  
<http://git.typo3.org/TYPO3v4/CoreProjects/MVC/extbase.git?a=commit;h=0a00500998053d111bb6ee57d609e83eb30d5364>
- Es wurden **Models** für **Backend-User**, **Backend-Usergroup** und **Filemounts** (`Classes/Domain/Model/BackendUser.php`, `Classes/Domain/Model/BackendUserGroup.php` und `Classes/Domain/Model/FileMount.php`) inkl. einem Repository (in welchem `$querySettings->setRespectStoragePage(FALSE)` ; gesetzt wurde) dafür sowie für die eingeführten **Kategorien** (`Classes/Domain/Model/Category.php`) hinzugefügt.

- Im ViewHelper **format.currency** gibt es zwei neue Argumente:
  - **prependCurrency**  
Gibt an, ob das Währungssymbol vorne angestellt wird
  - **separateCurrency** (Default TRUE)  
Gibt an ob das Währungssymbol mit Leerzeichen zur Zahl notiert wird

## Eingabe:

```
<f:format.currency currencySign="$" decimalSeparator="." thousandsSeparator="," prependCurrency="TRUE" separateCurrency="FALSE">54321</f:format.currency>
```

## Ausgabe:

\$54,321.00

- Im ViewHelper **translate** kann man nun die Extension spezifizieren, von der das Sprachlabel verwendet wird:
  - **extensionName**  
UpperCamelCased Extension Key (z.B. BlogExample)

## Eingabe:

```
{f:translate(key: 'someKey', extensionName: 'SomeExtensionName')}
```

## Ausgabe:

```
// Wert des Schlüssels "someKey" in der eingestellten Frontend-Sprache
// Benutzt wird die locallang Datei der extension "some_extension_name"
```

- Im ViewHelper **groupedFor** kann man nun beim Argument **groupBy** auch einen Eigenschaftspfad (PropertyPath) verwenden (vorher funktionierte nur eine Eigenschaft).

- Alle Tag-based ViewHelpers können nun mit einem optionalen Default-Value ausgestattet werden

```
function registerTagAttribute($name, $type, $description, $required = FALSE,
$default = NULL) {
```

- Der ViewHelper **translate** verwendet nun (wie unter Fluid in TYPO3 Flow) das Attribut **id** für die Referenz auf die LocalLang. Das bisher verwendete Attribut **key** wird mit Version 6.2 entfernt. Werden beide Attribute verwendet, bekommt **id** Vorrang.

```
<f:translate id="..." />
```

- Extbase & Fluid werden nun IMMER geladen und können auch nicht deinstalliert werden.
- Die Sprachlabels im Widget paginate sind nun nicht mehr hartkodiert (sondern in `Resources/Private/Language/locallang.xlf`)

# HOOKS

Hooks & Signal/Slots

- Damit das Backend-Login Formular komplett angepasst werden kann, wurde ein Hook in Form eines Signal/Slots integriert:

```
protected function emitRenderLoginFormSignal(array &$markers) {
 $this->getSignalSlotDispatcher()->dispatch(
 // Signal class name
 'TYPO3\CMS\Backend\Controller>LoginController',
 // Signal name:
 // const SIGNAL_RenderLoginForm = 'renderLoginForm';
 self::SIGNAL_RenderLoginForm,
 //
 array($this, $markers)
);
}
```



- Wenn man den File Abstraction Layer (FAL), verwendet, können Ressourcen auf einem Remote Storage wie Amazon S3 oder WebDAV liegen. Jedoch darf in diesem Fall die geschützte URL (welche die Benutzer Zugangsdaten enthält) für den Upload nicht zugänglich für den normalen User sein.
- Mit diesem Hook kann die Generierung der öffentlichen URL beeinflusst werden:

```
$this->getSignalSlotDispatcher()->connect(
 '\TYPO3\CMS\Core\Resource\ResourceStorage',
 '\TYPO3\CMS\Core\Resource\ResourceStorage::SIGNAL_PreGeneratePublicUrl',
 'Tx_MyExtension_HandlerSlot',
 'preGeneratePublicUrl'
);
```

- Wenn Seiten in einem zusätzlichen System (wie z.B. nc\_staticfilecache) gecached werden, müssen diese nicht zusätzlich noch (in welchem Cache Backend auch immer) gecached werden. Durch diesen neuen Hook kann die Speicherung einer Seite im Cache selektiv deaktiviert werden.

```
['SC_OPTIONS']['tslib/class.tslib_fe.php']['usePageCache']
```

```
$usePageCache = $_procObj->usePageCache($this,
$usePageCache = TRUE);
```

- Hook vor der Bestimmung der Page ID - TSFE `determineId()`  
`['SC_OPTIONS']['tslib/class.tslib_fe.php']`  
`['determineId-PreProcessing']`
- Hook nach der Bestimmung der Page ID - TSFE `fetch_the_id()`  
`['SC_OPTIONS']['tslib/class.tslib_fe.php']`  
`['fetchPageId-PostProcessing']`
- Hook in der Extension „felogin“ um Login-Error weiter zu bearbeiten  
`['EXTCONF']['felogin']['login_error']`

# DEPRECATED FUNCTIONS

Veraltete Funktionen

- Im Modul „DB check > Full search“ kann man ab sofort nur noch ein CSV exportieren, aber kein XML mehr. Daher wurde auch die Klasse `t3lib_xml` als deprecated klassifiziert und wird in TYPO3 6.2 entfernt.
- Das Feature, jeden Page-Request entweder in die Datenbank zu schreiben oder in ein Apache-Style ähnliches Logfile wurde kaum benutzt und daher aus TYPO3 entfernt. Die Funktionalität kann gut über Google Analytics oder Piwik und ähnliche Tools ersetzt werden.

- **Flag doNotLoadInFE wurde entfernt**

Das Flag, welches in TYPO3 4.3 eingeführt wurden, diente dazu in der Datei `ext_emconf.php` kennzuzeichnen, dass die Extension keinerlei Frontend-Funktionalität hat. Das Flag und das korrespondierende Setting `extList_FE` in der `LocalConfiguration.php` wurden entfernt.

- **Die Extension `simulatestatic` wurden entfernt**

Beim Update wird die Extension aus dem TER nachgeladen, wenn diese vorher installiert war.

- **Die TypoScript Option `noBlur` wurde entfernt**

- **Historische Daten werden nicht mehr gelöscht**

TYPO3 löscht die historischen Daten, welche in der Systemtabelle `sys_history` gesammelt werden, nicht mehr selbstständig. Dies kann zu großen Datenmengen in dieser Tabelle führen. Daher sollte man den Garbage collection scheduler task verwenden um alte Einträge zu löschen.

- **Removed compression of javascript files with jsmin**

Komprimierung mittels jsmin wurde aufgrund von Lizenzproblemen entfernt. Stattdessen wird GZIP-Komprimierung Serverseitig empfohlen

# UPGRADE

von einer früheren TYPO3 Version

- **Download eines Packages von <http://typo3.org/download/>**
- **Upgrade-Anleitung**
  - Überprüfen Sie die Systemvoraussetzungen - diese sind **nicht** mit 4.7 identisch!
  - Kontrollieren Sie das „Deprecation Log“, ob Ihre aktuelle Installation veraltete Funktionen verwendet, welche mit TYPO3 6.0 entfernt werden.
  - Verlinken Sie die neuen Sourcen
  - Gehen Sie ins Install Tool:
    - Verwenden Sie den "Update wizard" (Durch alle Schritte mittels "Next" weitergehen)
    - Verwenden Sie "Database COMPARE" um alle relevanten Datenbank-Änderungen zu übernehmen.



- **Liste der mit TYPO3 6.0 inkompatiblen Extensions**

- Nahezu die Hälfte aller vorhandenen Extensions (über 2200) sind **NICHT** kompatibel mit TYPO3 6.0.

Eine Liste aller Extensions und der Grund der Inkompatibilität (meist statische Aufrufe veralteter Funktionen) sind hier zu finden:

[http://wiki.typo3.org/TYPO3\\_6.0\\_Incompatible\\_Extensions\\_A](http://wiki.typo3.org/TYPO3_6.0_Incompatible_Extensions_A)

...

[http://wiki.typo3.org/TYPO3\\_6.0\\_Incompatible\\_Extensions\\_Z](http://wiki.typo3.org/TYPO3_6.0_Incompatible_Extensions_Z)

- **Migration-Tipps für Extension-Autoren**

- Im folgenden Dokument befinden sich Hinweise, wie veraltete statische Aufrufe ersetzt werden können:

[http://wiki.typo3.org/TYPO3\\_6.0\\_Extension\\_Migration\\_Tips](http://wiki.typo3.org/TYPO3_6.0_Extension_Migration_Tips)

# QUELLEN

Weitere Quellen

- **NEWS:**

<http://typo3.org/news/article/typo3-60-released/>

- **Release Notes**

<http://typo3.org/download/release-notes/typo3-6-0-release-notes/>

- **Technische Release Notes**

[http://wiki.typo3.org/TYPO3\\_6.0](http://wiki.typo3.org/TYPO3_6.0)

- **TYPO3 GIT**

<http://git.typo3.org/TYPO3v4/Core.git>

- **TYPO3 Bugtracker**

<http://forge.typo3.org/projects/typo3v4-core/issues>

- **NEWS.txt** (im Rootverzeichnis der Sourcen)

- **Logging API:**

<http://forge.typo3.org/projects/typo3v4-logging>

- **File Abstraction Layer (FAL):**

<http://www.slideshare.net/fullscreen/sritterkh1/the-file-abstraction-layer-in-typo3-60/>

<http://preview.docs.typo3.org/TYPO3/FileAbstractionLayerReference/>

- **Wiki:**

[http://wiki.typo3.org/TYPO3\\_6.0](http://wiki.typo3.org/TYPO3_6.0)

- **XCLASS und Autoloader:**

<http://wiki.typo3.org/Autoload>

- **Liste der inkompatiblen Extensions:**

[http://wiki.typo3.org/TYPO3\\_6.0\\_Incompatible\\_Extensions\\_A \(bis \\_Z\)](http://wiki.typo3.org/TYPO3_6.0_Incompatible_Extensions_A_(bis_Z))

- **Details zu den Namespaces:**

<http://wiki.typo3.org/Namespaces>

# NEUE TYPO<sub>3</sub> MARKENWELT

TYPO<sub>3</sub> Branding

- **Neues Branding ab 01.10.2012**

- Auf [typo3.org](http://typo3.org) wurde am 01.10.2012 eine News mit dem Titel „A Brand New Way“ veröffentlicht, welche das Rebranding offiziell gemacht hat

<http://typo3.org/news/article/a-brand-new-way/>

- In ihrem 15 jährigen Bestehen wollte man die Marke harmonisieren, homogenisieren und stärken, sowie die aktuellen und zukünftigen Produkte zu einer Markenfamilie zusammenführen.

- **Das Logo-Symbol selbst ist ab sofort einfarbig und zwar im bisher verwendeten Orange - grün fällt weg.**
- **Das Symbol selbst steht für „TYPO3“ und drückt damit die Zugehörigkeit zur Dachmarke aus.**



**altes Logo-Symbol**



**neues Logo-Symbol**

Pantone Orange 021C

R: 255, G: 135, B:0

#FF8700

- TYPO3 wird zeitgleich zur **Dachmarke** erhoben



- Mit oben gezeigtem Logo wird das TYPO3-Projekt ansich gekennzeichnet und **zeitgleich das CMS TYPO3 4.x/6.x!**



- Für alle TYPO3 Produkte wird die Marke wie folgt aufgebaut:  
**[Logosymbol] + PRODUKTNAME** (alles Großbuchstaben!)



# PRODUKT

- Dabei wird das Produkt wie folgt ausgesprochen, bzw. notiert:  
**TYP03 Produkt** (Kleinbuchstaben mit erstem Buchstaben groß!)
- Tägliche Benutzung: **Produkt**

- Das Application Framework **FLOW<sub>3</sub>** welches seit ca. 6 Jahren in Entwicklung ist, heißt ab sofort **TYPO3 Flow**.



- **TYPO3 Flow** wurde am 24.08.2012 in der Version 1.1.0 veröffentlicht.
- Tägliche Benutzung: **Flow**

- Der neueste Sproß aus dem Hause TYPO3 heißt **TYP03 Neos** und war vorher unter dem Namen „**Phoenix**“ bzw. davor als TYPO3 5.x bekannt. Dies soll der Nachfolger des klassischen CMS TYPO3 werden.



- **TYP03 Neos** wurde am 05.10.2012 in der Version Alpha 1 auf der T3CON12 veröffentlicht. Basis für dieses CMS ist TYPO3 Flow.

- Weitere Produkte werden jeweils wie folgt benannt:

- **TYPO3 Fluid**

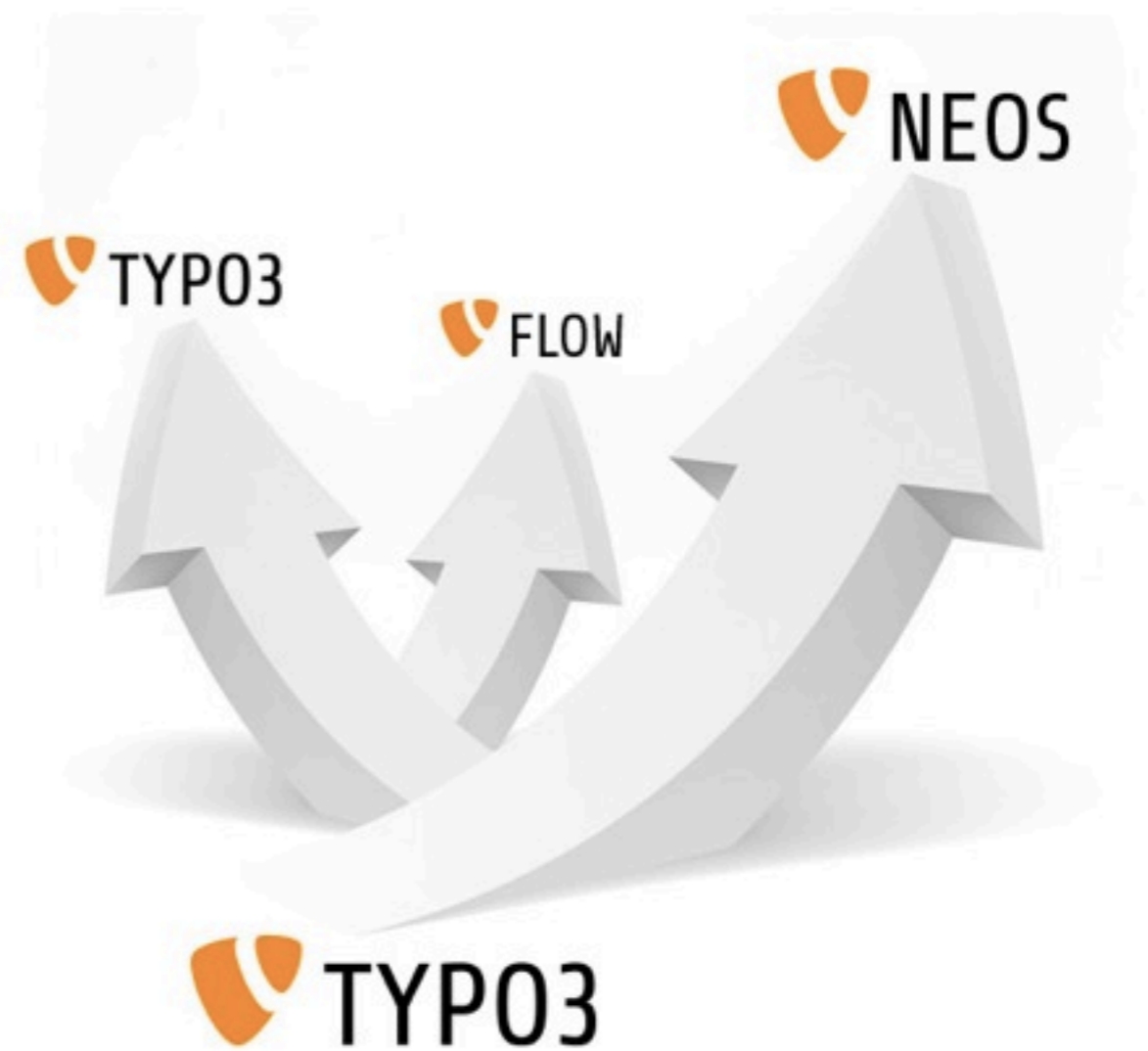
(wird in Kürze als Produkt zur Verfügung stehen)

- **TYPO3 Extbase**

(Produktinformation steht aus)

- **TYPO3 Surf**

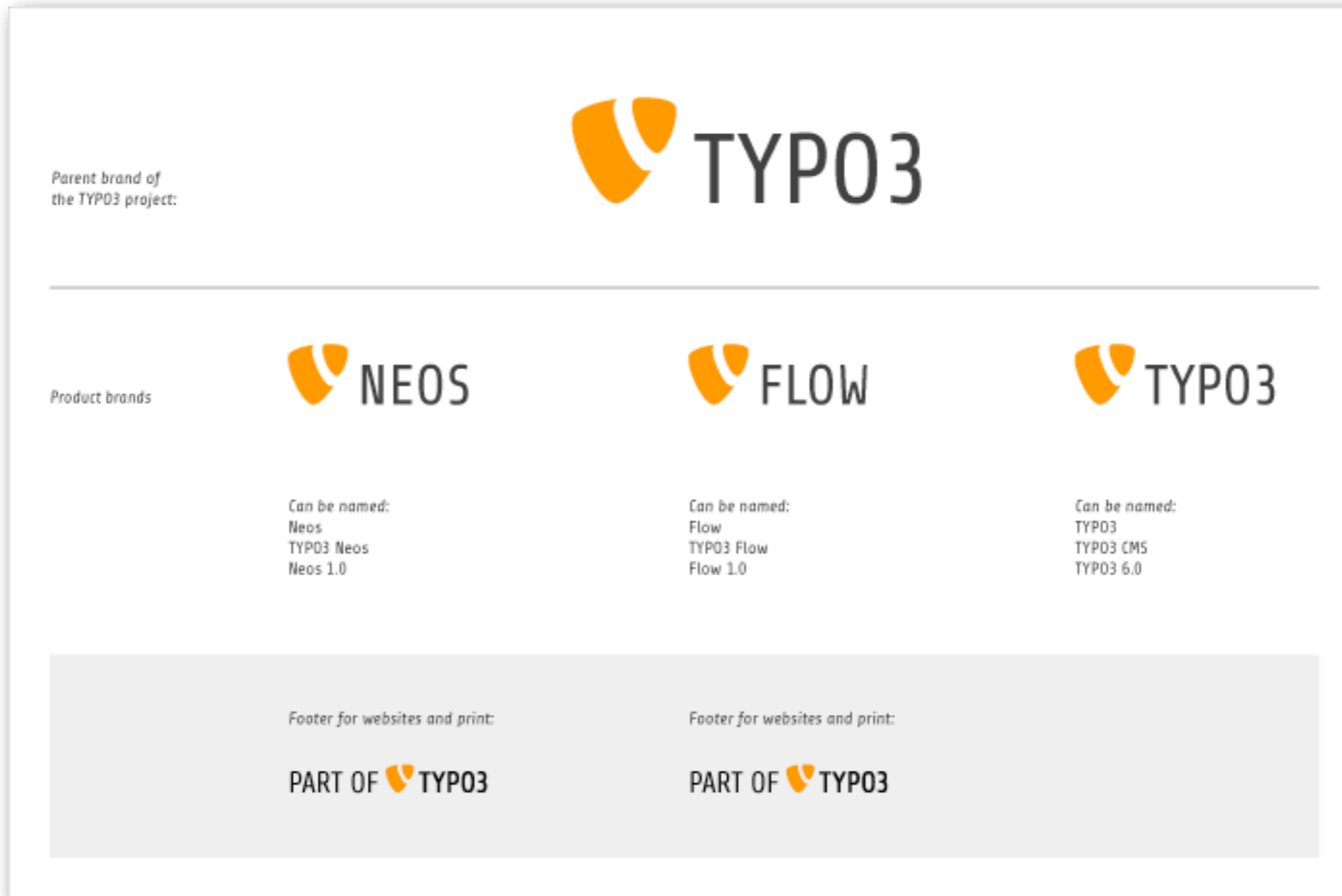
(wird in Kürze als Produkt zur Verfügung stehen)



- Das bisherige TYPO3 4.x bzw. 6.x wird ab sofort mit vollem Namen wie folgt geschrieben:  
**TYPO3 CMS** 4.x bzw. **TYPO3 CMS** 6.x - für den vollen Namen gibt es **KEINE** Logo-Entsprechung



- In der Kurzform wird man aber dazu weiterhin **TYPO3** 4.x bzw. **TYPO3** 6.x sagen.



<http://de.wikipedia.org/wiki/Dachmarke>

<http://typo3.org/news/article/a-brand-new-way/>

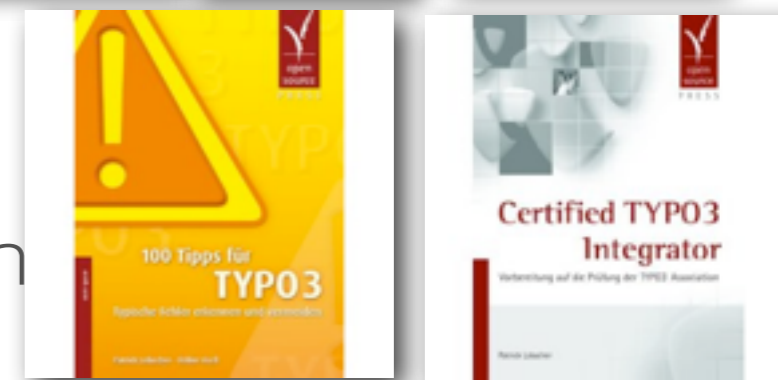
<http://typo3.org/the-brand/a-brand-new-way/>

<http://t3n.de/news/typo3-cms-neos-flow-ubersicht-427253/t3n30-infografik-typo3/>

VIELEN DANK!  
FRAGEN?

# ÜBER PATRICK LOBACHER

- Patrick Lobacher (geb. Schuster) - Geschäftsführer
- 42 Jahre alt, glücklich verheiratet, wohnhaft in München
- Autor von 7 Fachbüchern und 35 Fachartikeln zum Thema TYPO3 und Webentwicklung
- Certified TYPO3 Integrator seit 2009
- Bis 2012 Mitglied im Team Extbase (Leader), Certification und im EAB (Expert Advisory Board)
- Mitveranstalter des TYPO3camp Munich und TYPO3 Developer Days (T3DD12 und T3DD13)
- Speaker auf nationalen und internationalen Kongressen
- Dozent für führende Schulungsinstitute und die MVHS





- Münchner Fullservice-Agentur für digitale Kommunikation
- Platinum Member der TYPO3 Association
- über 30 Mitarbeiter (+ 10 aus festem Freelancer Pool)
- Geschäftsführer: Sebastian Böttger, Patrick Lobacher
- Spezialisiert auf TYPO3 seit 10 Jahren (Extbase/Fluid seit 2009)
- **Agenturpräsentation unter: [www.typovision.de/dieagentur](http://www.typovision.de/dieagentur)**
- Über 500 TYPO3-Projekte jeglicher Größenordnung - für Kunden wie:



## Adresse:

typovision GmbH

Landshuter Allee 8  
80637 München

**Fon:** +49 89 45 20 59 3 - 0

**Fax:** +49 89 45 20 59 3 - 29

**Email:** [info@typovision.de](mailto:info@typovision.de)

**Web:** <http://www.typovision.de>

## Twitter:

[www.twitter.com/typovision](http://www.twitter.com/typovision)

[www.twitter.com/PatrickLobacher](http://www.twitter.com/PatrickLobacher)

## Facebook:

[www.facebook.com/typovision](http://www.facebook.com/typovision)

## XING:

[www.xing.com/profile/Patrick\\_Lobacher](http://www.xing.com/profile/Patrick_Lobacher)

## Slideshare:

[www.slideshare.net/plobacher](http://www.slideshare.net/plobacher)

## Amazon:

[www.amazon.de/Patrick-Lobacher/e/Boo45AQVEA](http://www.amazon.de/Patrick-Lobacher/e/Boo45AQVEA)