



TYP03

TYP03 CMS 6.2 LTS

Die Neuerungen

typovision 

Patrick Lobacher

CEO typovision GmbH

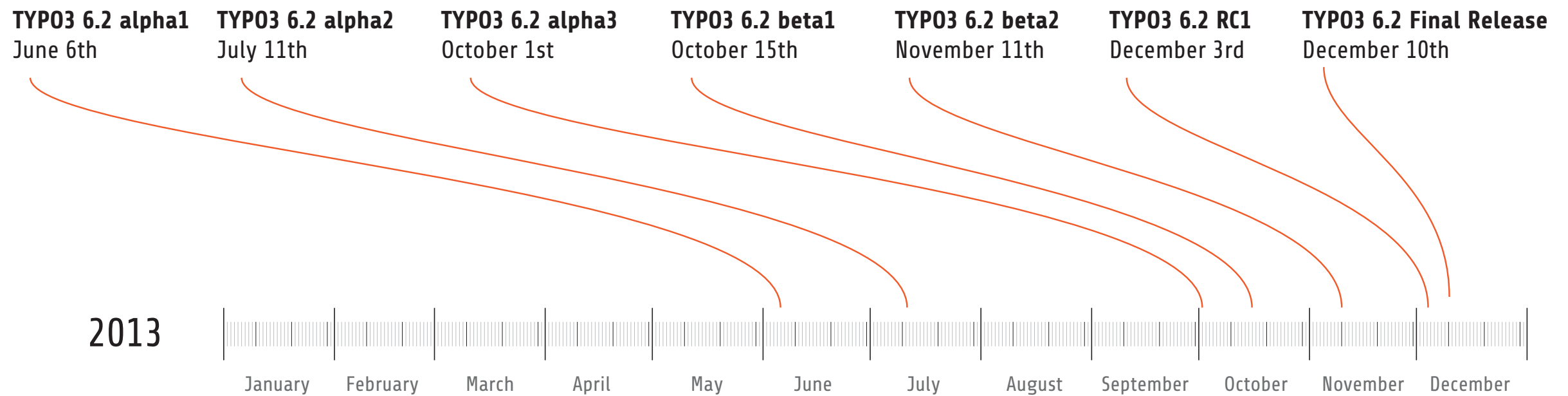
TYPO3 CMS 6.2 LTS - Short facts

- Veröffentlicht: 10.12.2013
- End of Maintenance: 30.12.2016
(3 Jahre Support)
- Release Manager: Ernesto Baschny
ernesto.baschny(AT)typo3.org
@baschny
- Hauptziele: LTS
- Systemvoraussetzungen: PHP 5.3.7 - 5.5.x, MySQL 5.1.x-5.5.x
- Download: <http://typo3.org/download/>



Bildquelle: t3con13de.typo3.org

TYPO3 CMS 6.2 LTS Release



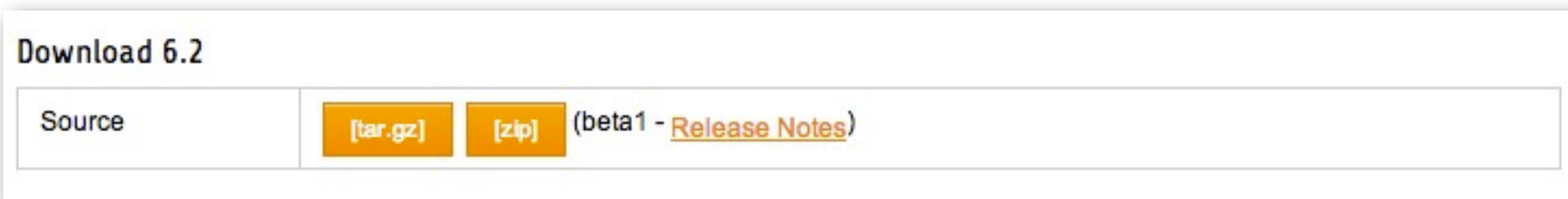
Inhaltsverzeichnis

- **Install-Tool Rewrite**
- **Distribution Management**
- **TYPO3 CMS goes Responsive**
- **Änderungen im Backend**
- **TSconfig & TypoScript**
- **Package Management**
- **Änderungen im System**
- **API**
- **Extbase & Fluid**
- **Entfernte und veraltete Funktionen**
- **Upgrade von TYPO3 CMS 4.5 LTS**
- **Quellen & Ressourcen**
- **Über den Autor**

Install-Tool **Rewrite**

Install Tool Rewrite

- Das Install Tool wurde komplett neu auf Basis von Extbase & Fluid geschrieben
- Wichtigste Änderung im Install-Prozess ist, dass es nur noch ein Paket gibt: TYPO3 Source (kein Dummy Package, Blank Package, ...) - Größe ca. 20 MB
- Zum Installieren einfach das zip oder tar.gz herunterladen, entpacken und in das Document Root legen
- Anschließend einfach die URL aufrufen



Dummy Paket ist ab sofort obsolet

- Der Installer sorgt nun dafür, dass alle benötigten Dateien an der richtigen Stelle sind
- Insbesondere kann er auch alle Dateien anlegen, die für eine persönliche Instanz notwendig sind
- Damit ist das Dummy Paket komplett obsolet
- Es ist lediglich wichtig, dass folgende Dateien und Verzeichnisse richtig verlinkt sind:

typo3_src (Verzeichnis, dies muss auf die Sourcen zeigen - sofern externe Sourcen verwendet werden)

typo3 (Verzeichnis, zeigt auf **typo3_src/typo3**)

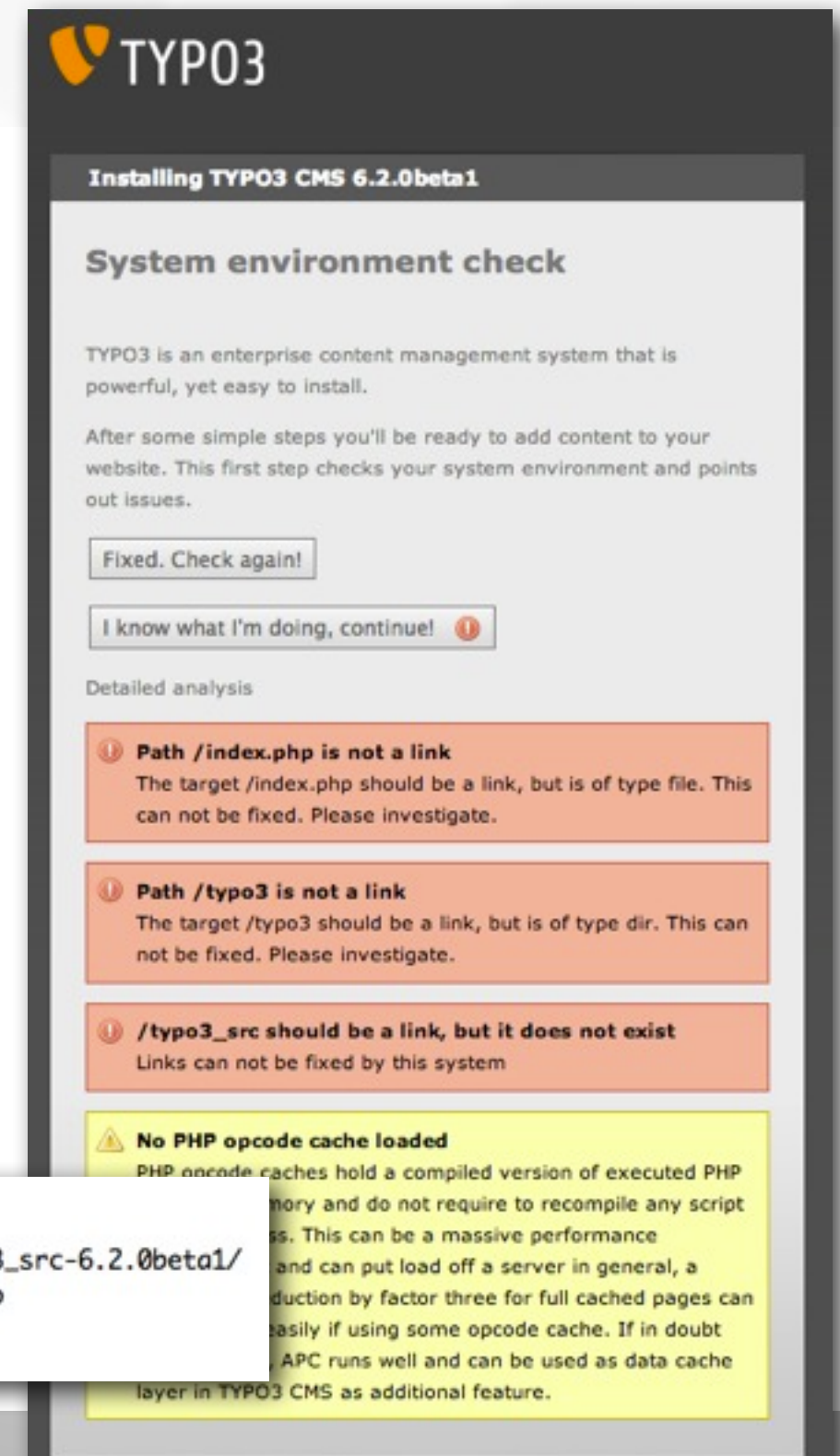
index.php (Datei, zeigt auf **typo3_src/index.php**)

- Weitere Dateien sind **nicht** notwendig zur Installation von TYPO3

Install Tool Rewrite

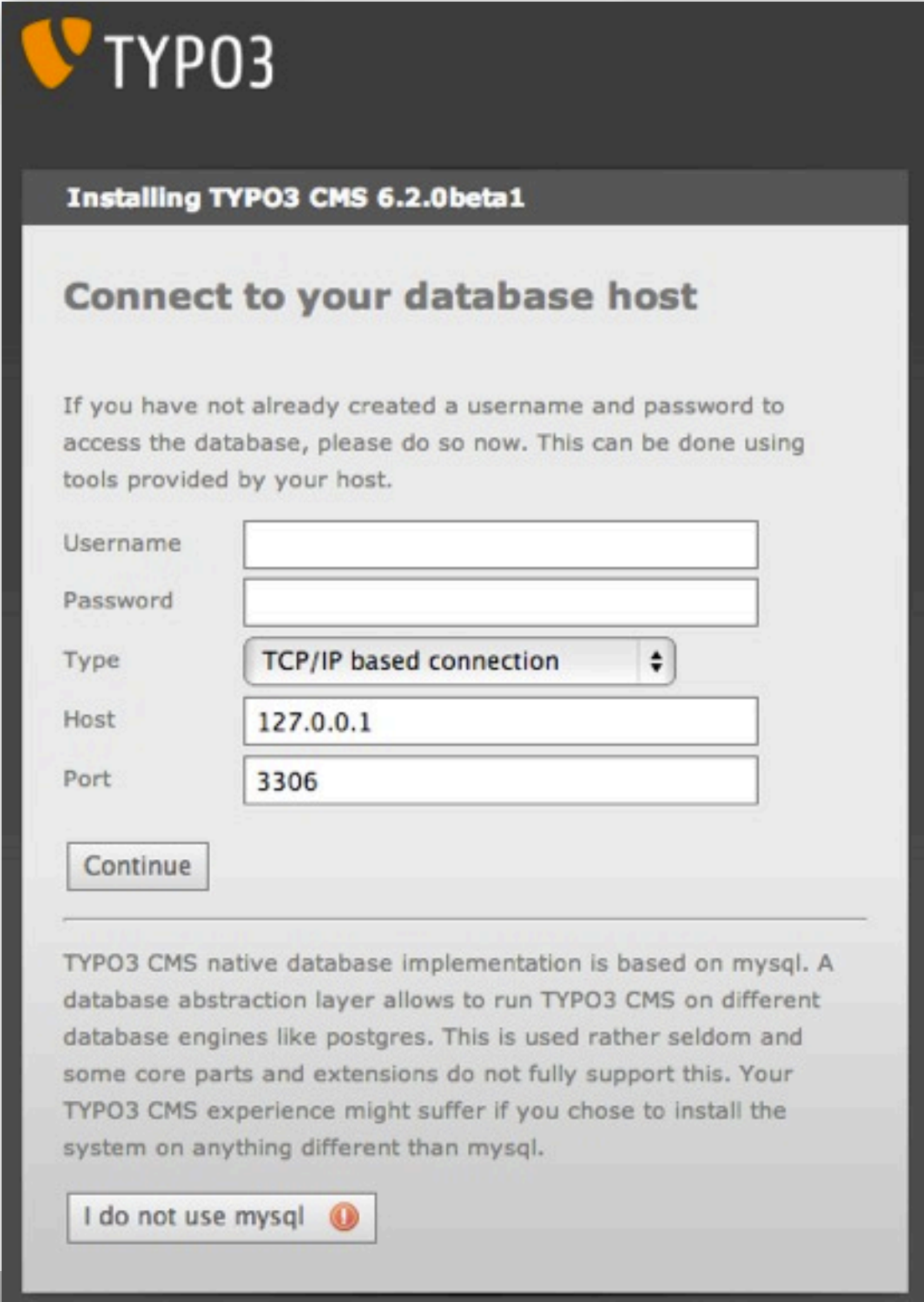
- Im ersten Schritt wird nun die Systemumgebung getestet
- Hier kann man nun die beanstandeten Einstellungen beheben und erneut testen oder die Tests übergehen
- Hat man den Core nicht als typo3_src Verzeichnis angelegt und darauf verlinkt, erhält man eine Fehlermeldung (unten stehende Verlinkung ist optimal)

```
-rwxrwxrwx@ 1 patricklobacher staff 5495 15 Okt 18:42 .htaccess
drwxr-xr-x 3 patricklobacher staff 102 20 Okt 12:30 typo3-sources
lrwxr-xr-x 1 patricklobacher staff 35 20 Okt 12:32 typo3_src -> typo3-sources/typo3_src-6.2.0beta1/
lrwxr-xr-x 1 patricklobacher staff 19 20 Okt 12:32 index.php -> typo3_src/index.php
lrwxr-xr-x 1 patricklobacher staff 15 20 Okt 12:33 typo3 -> typo3_src/typo3
drwxrwx--- 3 _www staff 102 20 Okt 12:33 typo3temp
```



Install Tool Rewrite

- Im zweiten Schritt kann man die Datenbank-Parameter eingeben
- Hier ist es möglich als Typ entweder „TCP/IP based connection“ (empfohlen!) oder „Socket based connection“ (voreingestellt) auszuwählen
- Benutzt man kein MySQL, so kann man dies über den entsprechenden Button ganz unten angeben und im nächsten Schritt einen anderen Datenbank-Treiber auswählen (dafür wird der Database Abstraction Layer geladen)



TYPO3

Installing TYPO3 CMS 6.2.0beta1

Connect to your database host

If you have not already created a username and password to access the database, please do so now. This can be done using tools provided by your host.

Username


Password

Type

Host

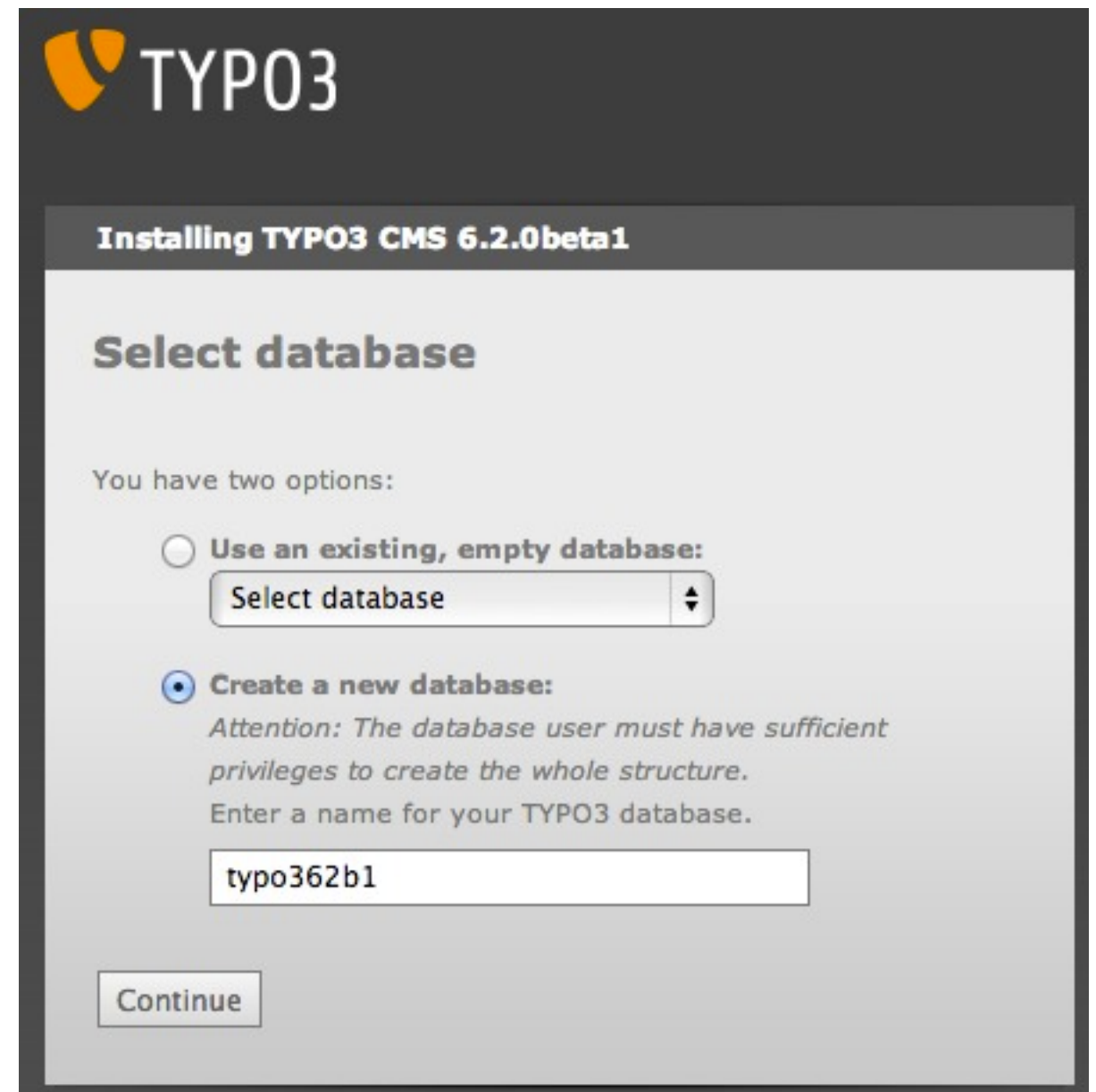
Port

TYPO3 CMS native database implementation is based on mysql. A database abstraction layer allows to run TYPO3 CMS on different database engines like postgres. This is used rather seldom and some core parts and extensions do not fully support this. Your TYPO3 CMS experience might suffer if you chose to install the system on anything different than mysql.



Install Tool Rewrite

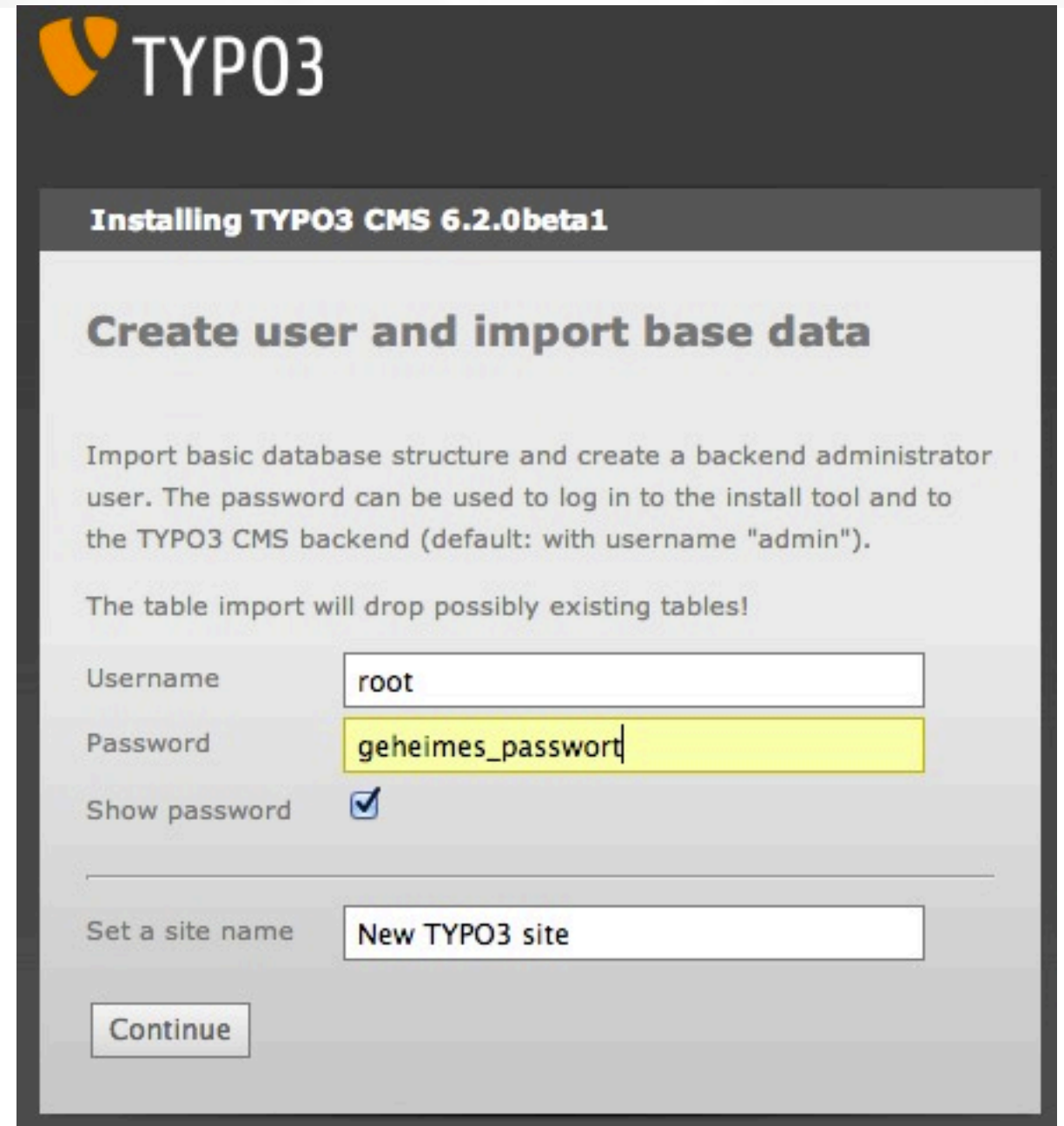
- Im dritten Schritt kann man die Datenbank selektieren
- Dazu wählt man entweder eine der vorhandenen leeren oder gefüllten Datenbanken aus
- Oder man legt eine neue Datenbank an



The screenshot shows the 'Installing TYPO3 CMS 6.2.0beta1' window. The title is 'Select database'. Below the title, it says 'You have two options:'. There are two radio button options. The first option is 'Use an existing, empty database:', which is currently unselected. Below it is a dropdown menu labeled 'Select database'. The second option is 'Create a new database:', which is currently selected. Below this option, there is a note: 'Attention: The database user must have sufficient privileges to create the whole structure. Enter a name for your TYPO3 database.' Below the note is a text input field containing the text 'typo362b1'. At the bottom left of the window is a 'Continue' button.

Install Tool Rewrite

- Im vierten Schritt legt man den Usernamen und das Passwort für den Admin-User fest
- Hier kann man sich über eine Checkbox das Passwort zur Kontrolle auch anzeigen lassen
- Zusätzlich gibt man den Website-Namen an



TYPO3

Installing TYPO3 CMS 6.2.0beta1

Create user and import base data

Import basic database structure and create a backend administrator user. The password can be used to log in to the install tool and to the TYPO3 CMS backend (default: with username "admin").

The table import will drop possibly existing tables!

Username

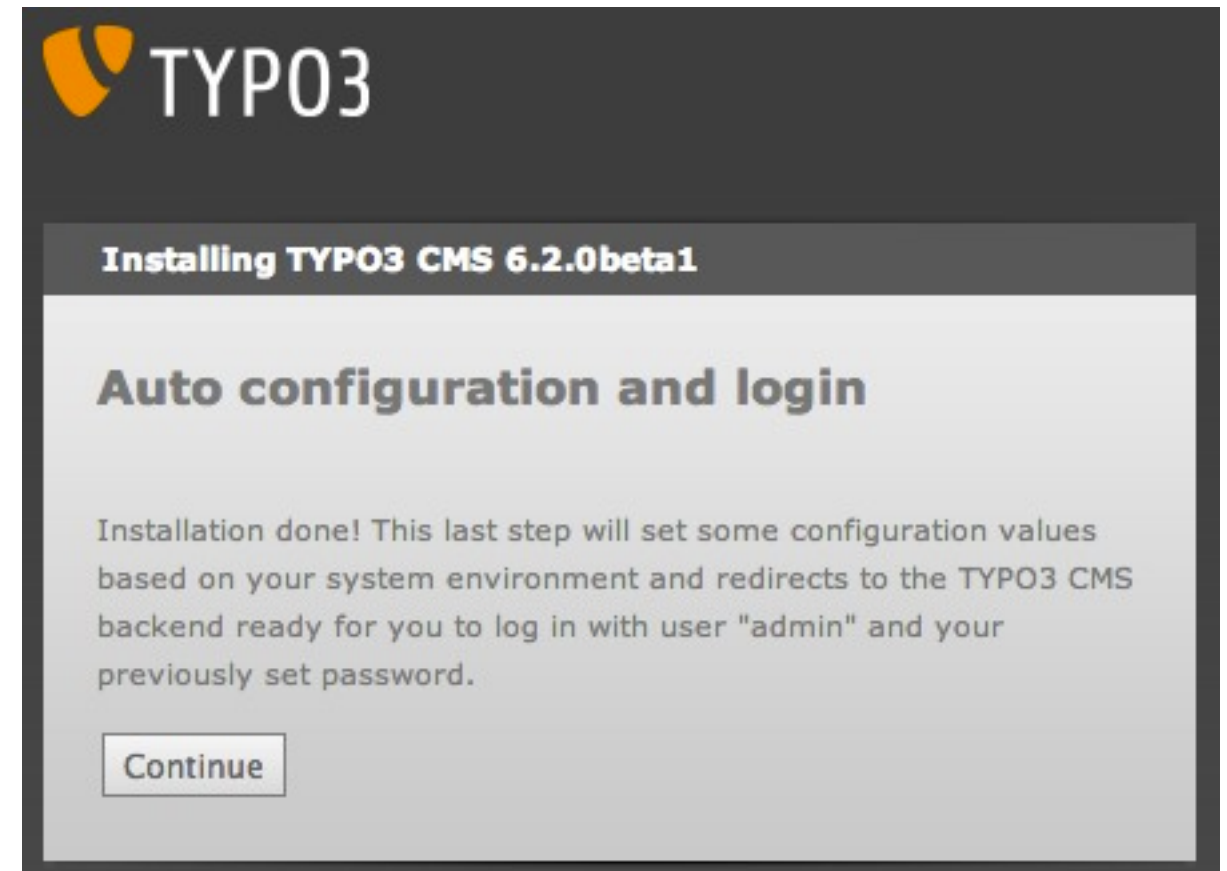
Password

Show password ☒

Set a site name

Install Tool Rewrite

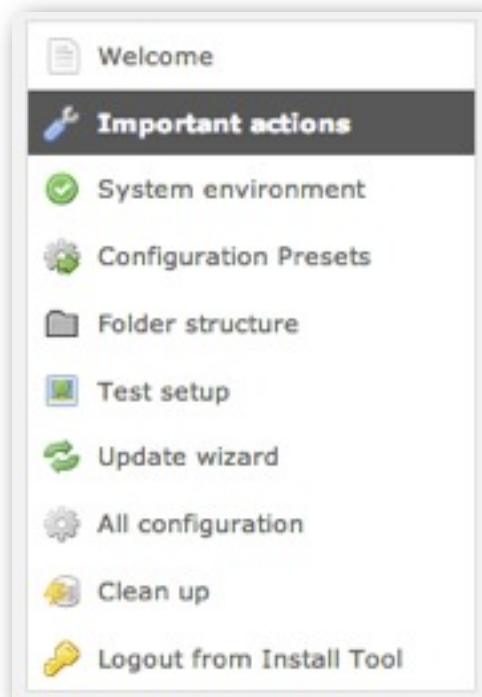
- Nun ist die Installation beendet
- Nach Klick auf „Continue“ wird man direkt zum Backend weitergeleitet, in welches man sich mit dem vorher festgelegten Usernamen und Passwort einloggen kann
- Dort sollte man dann weiter zum Install Tool gehen. Das Passwort ist das selbe wie für den Admin-User



Install Tool Rewrite - Cache löschen

- Im Bereich „Important Actions“ wurde die Möglichkeit geschaffen, den Cache komplett zu löschen
- Dies funktioniert auch, wenn im Cache fehlerhafte PHP-Dateien vorhanden sind, die das TYPO3 ansonsten unbenutzbar machen. Hierfür muss man das Install Tool direkt aufrufen:

<http://www.domain.de/typo3/install>



Zuerst wird das Verzeichnis typo3temp/Cache gelöscht, dann alle Tabellen cf_* geleert. Nun werden ext_localconf und ext_tables der Extensions geladen und der übliche flushCaches() ausgeführt.

Clear all cache

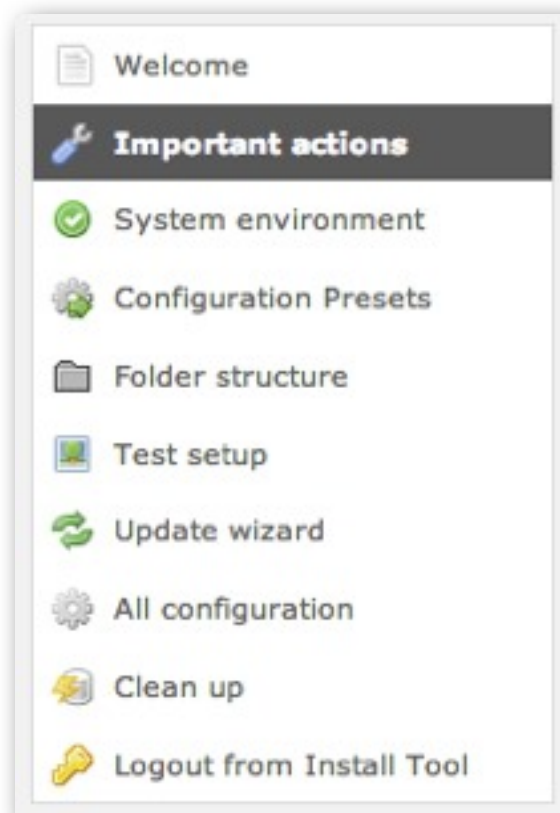
This clear all cache function works similar to the cache clearing in the backend but follows a more straight ahead approach and the according backend hooks are not executed.

This method can throw a fatal error if some broken extension is loaded. If you get a white page or a PHP error message, check your system with the broken extension test below.

Clear all cache

Install Tool Rewrite - Laden der Extensions prüfen

- Im Bereich „Important Actions“ wurde die Möglichkeit geschaffen, zu überprüfen, ob Extensions ohne Fehler geladen werden können (Wichtig z.B. für die Migration 4.5 -> 6.2)




Check for broken extensions

Checks whether the local extensions `ext_tables` and `ext_localconf` files can be included without errors, meaning they can be loaded without breaking the system.

Check extensions

Check for broken extensions

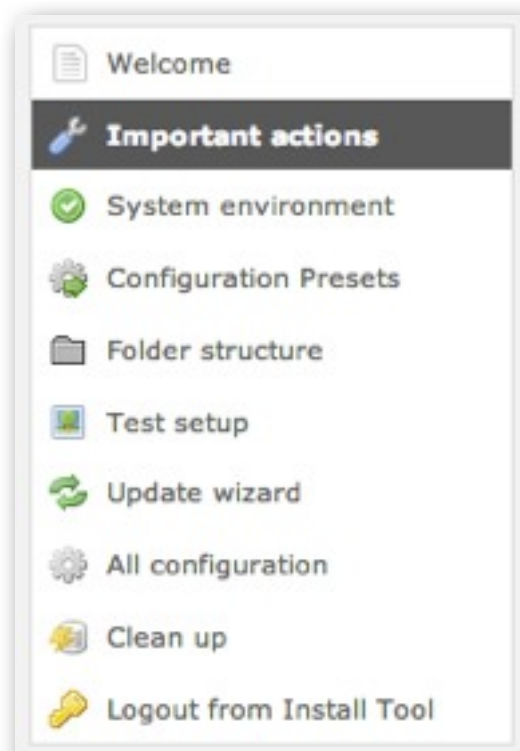
Checks whether the local extensions `ext_tables` and `ext_localconf` files can be included without errors, meaning they can be loaded without breaking the system.

 **All local extensions can be loaded!**

Install Tool Rewrite - BE User mit Salted Password


- Wenn ein Backend-User im Install Tool angelegt wird (entweder explizit im „Important actions“ Menü oder im Verlauf der Installation, geschieht dies nun mit einem Salted Password. Dafür muss die Extension `saltedpasswords` installiert sein (per Default) und der Schlüssel `[BE]` `[loginSecurityLevel] = rsa` gesetzt sein (ebenfalls Default).

Vor TYPO3 CMS 6.2 war `md5()` die Default Encryption.

A screenshot of the 'Create backend administrator user' form in the TYPO3 Install Tool. The form has a title 'Create backend administrator user' and a paragraph of instructions: 'You should use this function only if there are no admin users in the database, for instance if this is a blank database. After you've created the user, log in and add the rest of the user information, like email and real name.' Below the text are three input fields: 'Username:', 'Password:', and 'Password again:'. At the bottom of the form is a button labeled 'Create administrator'.

Install Tool Passwort ist gesalzen

- Das Install-Tool Password ist nun auch „salted“ (also gesalzen und damit nicht mehr nur MD5)
- Bestehende MD5-Passwörter werden beim ersten Login umgewandelt



The screenshot shows the TYPO3 6.2-dev Install Tool login interface. At the top, it says "Login to TYPO3 6.2-dev Install Tool". Below this, it says "Site: New TYPO3 site". There is a "Password" label and an input field. A "Login" button is below the input field. Below the login form, there is an orange error message box that says "Login failed" and "Given password does not match the install tool login password. Calculated hash: \$P\$CzyVrKM5PXXwRGoB9SYkpXwqZxhe64." Below this, there is a blue information box that says "The Install Tool Password is *not* the admin password of TYPO3. The default password is *joh316*. Be sure to change it!" and provides instructions on how to set a new password by modifying the `$TYPO3_CONF_VARS['BE']['installToolPassword']` in `typo3conf/LocalConfiguration.php`.

Kontext-Auswahl und Konfiguration

- Der (aus Flow bekannte) Kontext kann im Install Tool eingestellt und konfiguriert werden
- Damit kann man beispielsweise die Debug-Einstellungen, devIPmask und andere Presets einstellen

▼ Development / Production settings

TYPO3 can be run in a specific application context by using one of the built-in contexts "Production" (default), "Development" or "Testing". This can be used to provide specific configuration sets for each context. The context can be defined with the environment variable "TYPO3_CONTEXT" which is usually set through your webserver configuration (e.g. in htaccess).

However, if you don't set a context environment variable, you can still use the install tool to select a configuration preset for "Production" or "Development" context. The goal is to configure a production instance with maximum performance and no debug output that is possibly shown to users, while development instances should enable error output. The configuration preset for "Production" is set by default. As a third alternative, you can enter a custom configuration.

☒ **Production**
 Production settings turn off debug output, deprecation logs and set logging to warnings and errors only.

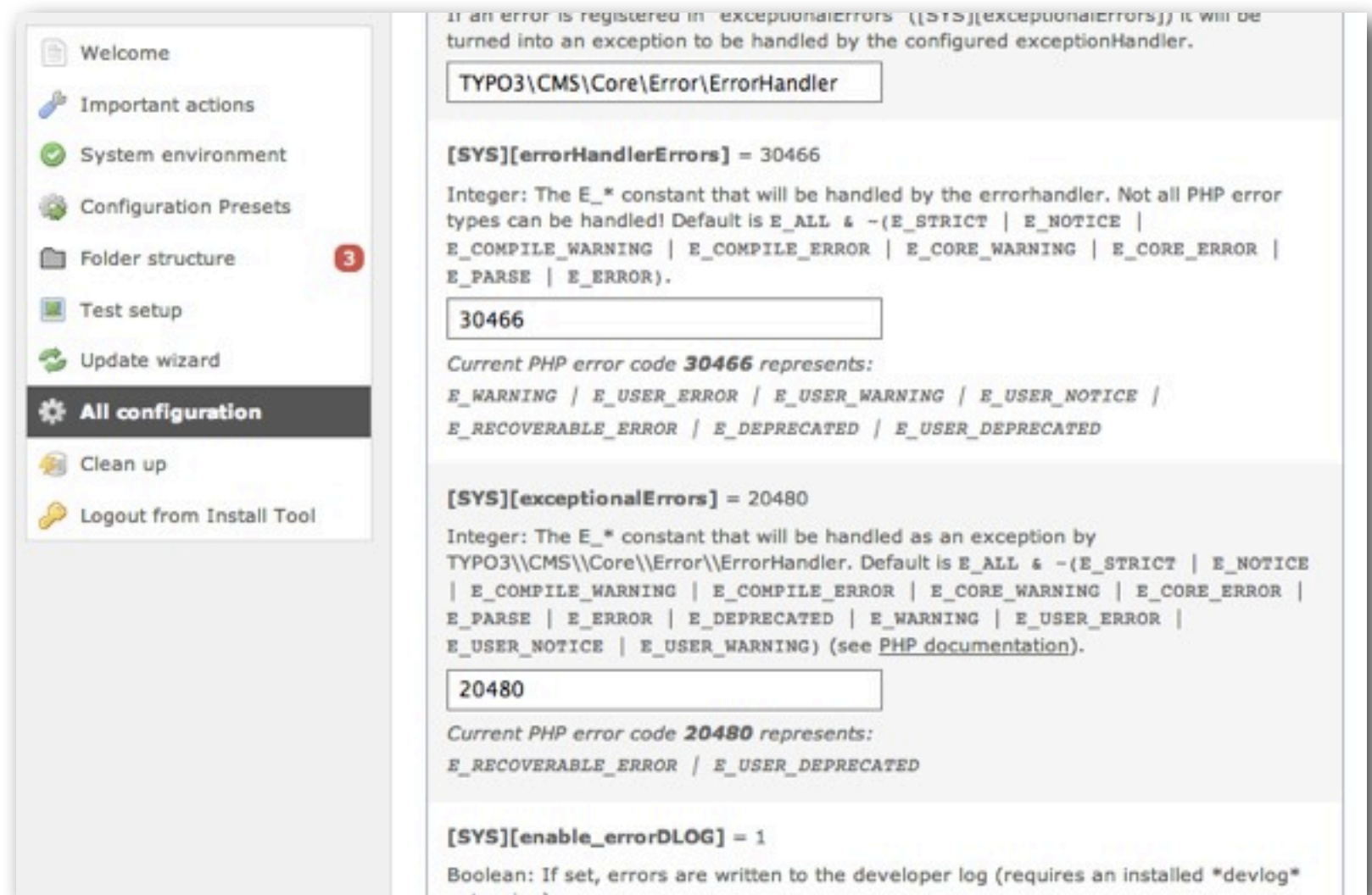
☒ **Development**
 Development settings enable debug output, deprecation logs and set logging to info level.

☒ **Custom configuration [Active]**
 Custom configuration mixture if no other preset fits.

<input type="text"/>	BE/debug
<input type="text"/>	FE/debug
<input type="text" value="127.0.0.1,::1"/>	SYS/devIPmask
<input type="text" value="-1"/>	SYS/displayErrors
<input type="text" value="file"/>	SYS/enableDeprecationLog
<input type="text" value="0"/>	SYS/sqlDebug
<input type="text" value="0"/>	SYS/systemLogLevel

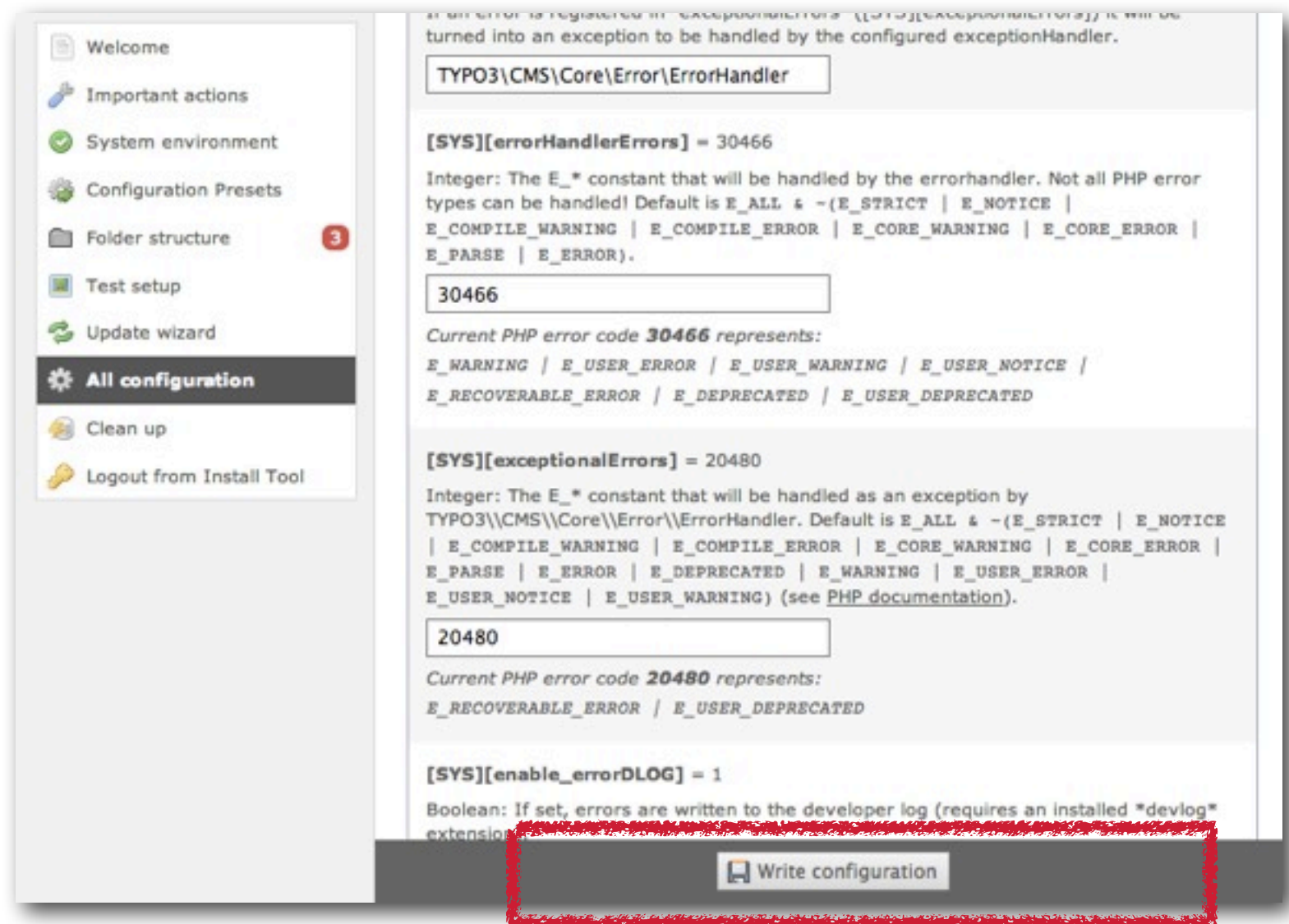
Install-Tool Menü scrollt nicht mit

- Das Menü auf der linken Seite ist nun fixiert und scrollt daher nicht mit
- So kann immer schnell auf eine andere Funktion umgeschaltet werden



Install-Tool Einträge direkt speichern

- Der Button „Write configuration“ ist permanent sichtbar und scrollt nicht mit
- So können getätigte Änderungen sofort und unmittelbar gespeichert werden



Error-Codes als Strings

- In den folgenden Install-Tool Optionen wurden bisher lediglich die Fehler-Codes ausgegeben:
`[SYS][errorHandlerErrors]`
`[SYS][exceptionalErrors]`
`[SYS][syslogErrorReporting]`
`[SYS][belogErrorReporting]`
- Hierfür wurde nun ein ViewHelper (`format.phpErrorCode`) hinzugefügt, um diese Codes in aussagekräftige Strings umzuwandeln

```
[SYS][errorHandlerErrors] = 30466
```

Integer: The E_* constant that will be handled by the errorhandler. Not all PHP error types can be handled! Default is E_ALL & ~(E_STRICT | E_NOTICE | E_COMPILE_WARNING | E_COMPILE_ERROR | E_CORE_WARNING | E_CORE_ERROR | E_PARSE | E_ERROR).

30466

Current PHP error code **30466** represents:

E_WARNING | E_USER_ERROR | E_USER_WARNING | E_USER_NOTICE |
E_RECOVERABLE_ERROR | E_DEPRECATED | E_USER_DEPRECATED

```
[SYS][exceptionalErrors] = 20480
```

Integer: The E_* constant that will be handled as an exception by TYPO3\CMS\Core>Error\ErrorHandler. Default is E_ALL & ~(E_STRICT | E_NOTICE | E_COMPILE_WARNING | E_COMPILE_ERROR | E_CORE_WARNING | E_CORE_ERROR | E_PARSE | E_ERROR | E_DEPRECATED | E_WARNING | E_USER_ERROR | E_USER_NOTICE | E_USER_WARNING) (see [PHP documentation](#)).

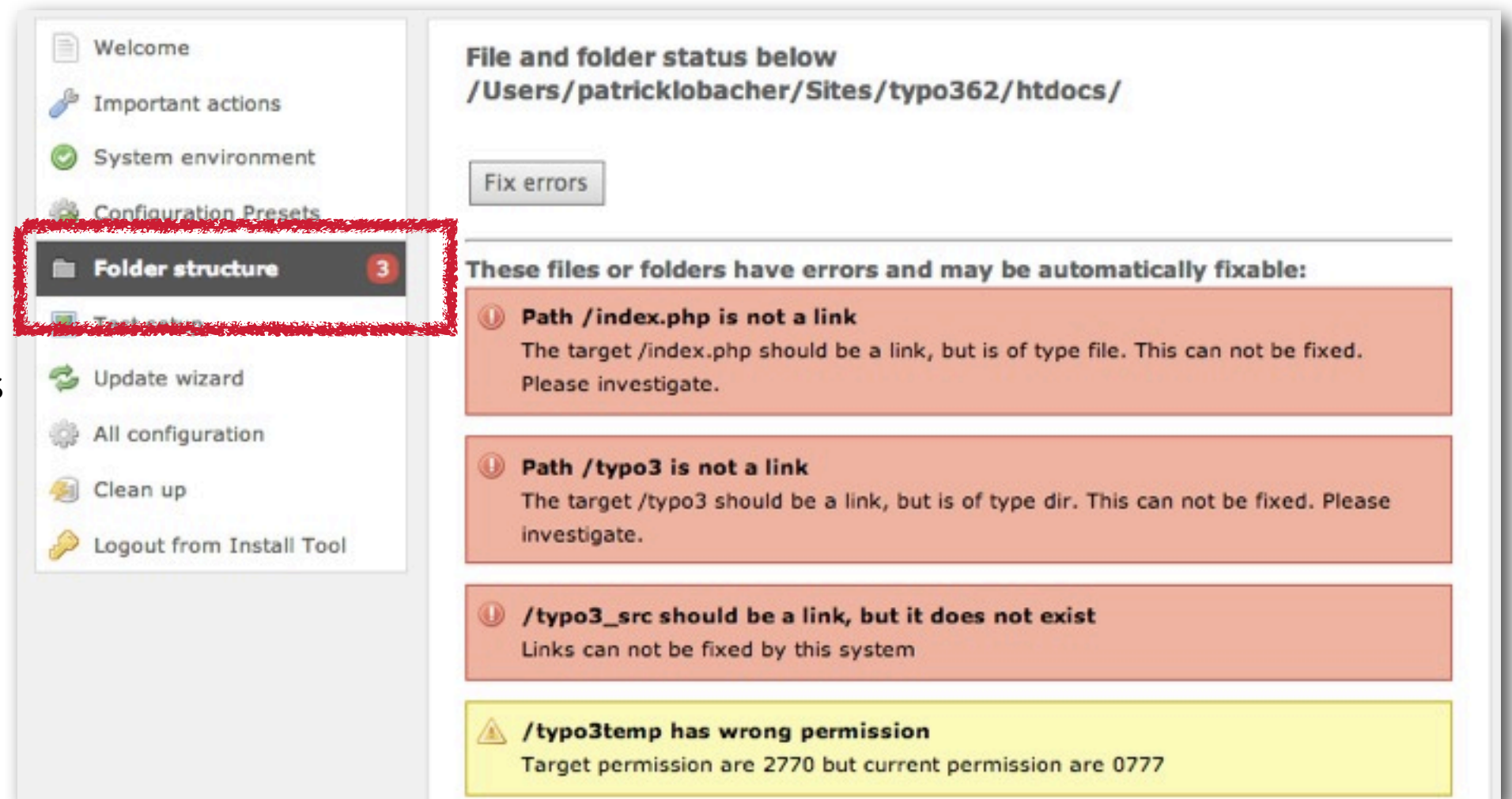
20480

Current PHP error code **20480** represents:

E_RECOVERABLE_ERROR | E_USER_DEPRECATED

Verzeichnis-Fehler werden angezeigt

- Im Install Tool werden nun Fehler in der Verzeichnis-Struktur direkt mit einem Hinweis angezeigt




Automatisches Core-Update


- Man kann nun Minor-Core-Updates (inkl. Security-Versionen) direkt aus dem Install Tool ausführen
- Durch die Umgebungsvariable **TYPO3_DISABLE_CORE_UPDATER = 1** kann der Core-Updater deaktiviert werden


Core update


The install tool can automatically update the TYPO3 CMS core to its latest minor release if certain criteria are met.


Check for core updates

 **Fetching list of released versions from typo3.org**

 **Fetched list of released versions**

 **No regular update available**

 **Fetched list of released versions**

 **Update to development release 6.2.0beta1 is available!**

Update now

Automatisches Core-Update

✓ Fetched list of released versions

! Update to development release 6.2.0beta1 is available!

Update now

! **Automatic core update not possible: Folder structure has errors or warnings**
To perform an update the folder structure of this TYPO3 CMS instance must stick to the conventions, or the update process could lead to unexpected results and may commit hazard to your system

```
sources/:
total 0
drwxr-xr-x 12 _www _www 408 15 Okt 18:42 typo3_src-6.2.0beta1
drwxr-xr-x 13 _www _www 442 23 Okt 14:06 typo362

.:
total 40
-rw-r--r--@ 1 _www _www 5141 11 Jul 15:29 .htaccess
lnwxr-xr-x 1 patricklobacher _www 15 23 Okt 14:08 typo3 -> typo3_src/typo3
lnwxr-xr-x 1 patricklobacher _www 19 23 Okt 14:08 index.php -> typo3_src/index.php
drwxrws---@ 9 _www _www 306 23 Okt 14:10 uploads
drwxrws---@ 112 _www _www 3888 23 Okt 14:10 typo3temp
drwxrws---@ 15 _www _www 510 23 Okt 14:10 typo3conf
drwxr-xr-x 4 _www _www 136 23 Okt 14:10 sources
drwxrws---@ 10 _www _www 340 23 Okt 14:10 fileadmin
lnwxr-xr-x 1 _www _www 72 23 Okt 14:11 typo3_src -> /Users/patricklobacher/Sites/typo362/htdocs/sources/typo3_src-6.2.0beta1
```

Core update

The install tool can automatically update the TYPO3 CMS core to its latest minor release if certain criteria are met.

✓ Fetched list of released versions

! Update to development release 6.2.0beta1 is available!

✓ System can be updated

✓ Core download finished

✓ Checksum verified

✓ Unpacking core successful

✓ Moved core to final location

✓ Caches cleared

✓ Core updated - please reload your browser

Gruppierung und Sortierung: All Configuration

- Der Bereich „All Configuration“ ist nun gruppiert
- Durch Klick auf die Überschrift klappt der zugehörige Bereich auf
- Die Überschriften sind zudem alphabetisch aufsteigend sortiert

Change configuration values

▼ \$TYPO3_CONF_VARS['BE']

[BE][unzip_path]
Path to "unzip".

[BE][diff_path] = diff
Path to "diff". For Windows this program can be downloaded here:
unxutils.sourceforge.net

Change configuration values

► \$TYPO3_CONF_VARS['BE']

► \$TYPO3_CONF_VARS['DB']

► \$TYPO3_CONF_VARS['EXT']


► \$TYPO3_CONF_VARS['FE']

► \$TYPO3_CONF_VARS['GFX']

► \$TYPO3_CONF_VARS['HTTP']

► \$TYPO3_CONF_VARS['MAIL']

► \$TYPO3_CONF_VARS['SYS']

 Write configuration

Install Tool - MISC 1

- Alle Install Tool Formulare sind CSRF geschützt
- Das Install Tool arbeitet mit einem vereinfachten Fluid Standalone View
- Der Startpunkt ist nun nicht mehr `typo3/install/index.php` sondern `typo3/sysextnl/install/Start/Install.php` (es gibt aber eine Weiterleitung)
- Es werden lediglich die minimal wichtigsten TYPO3-Funktionen geladen - kaputte `ext_localconf.php` oder `ext_tables.php` von Extensions führen nicht mehr zum Abbruch des Install Tools
- Das Install Tool verzichtet komplett auf Caching, somit können Fehler im Cache dem Install Tool nichts anhaben
- Die PHP-Option `xdebug.max_nesting_level` wird überprüft, da es mit dem Default-Wert von 100 unter TYPO3 CMS zu Problemen kommen kann. Empfohlen wird ein Wert von 250 oder höher.
- Im Install-Tool kann man nun bestehende (gefüllte) Datenbanken auswählen oder eine leere Datenbank anlegen bzw. auswählen
- Es wurde ein „Distribution management“ zugefügt (um Distributionen wie „Introduction Package“ oder „Government Package“ während der Installation zu laden und zu konfigurieren)

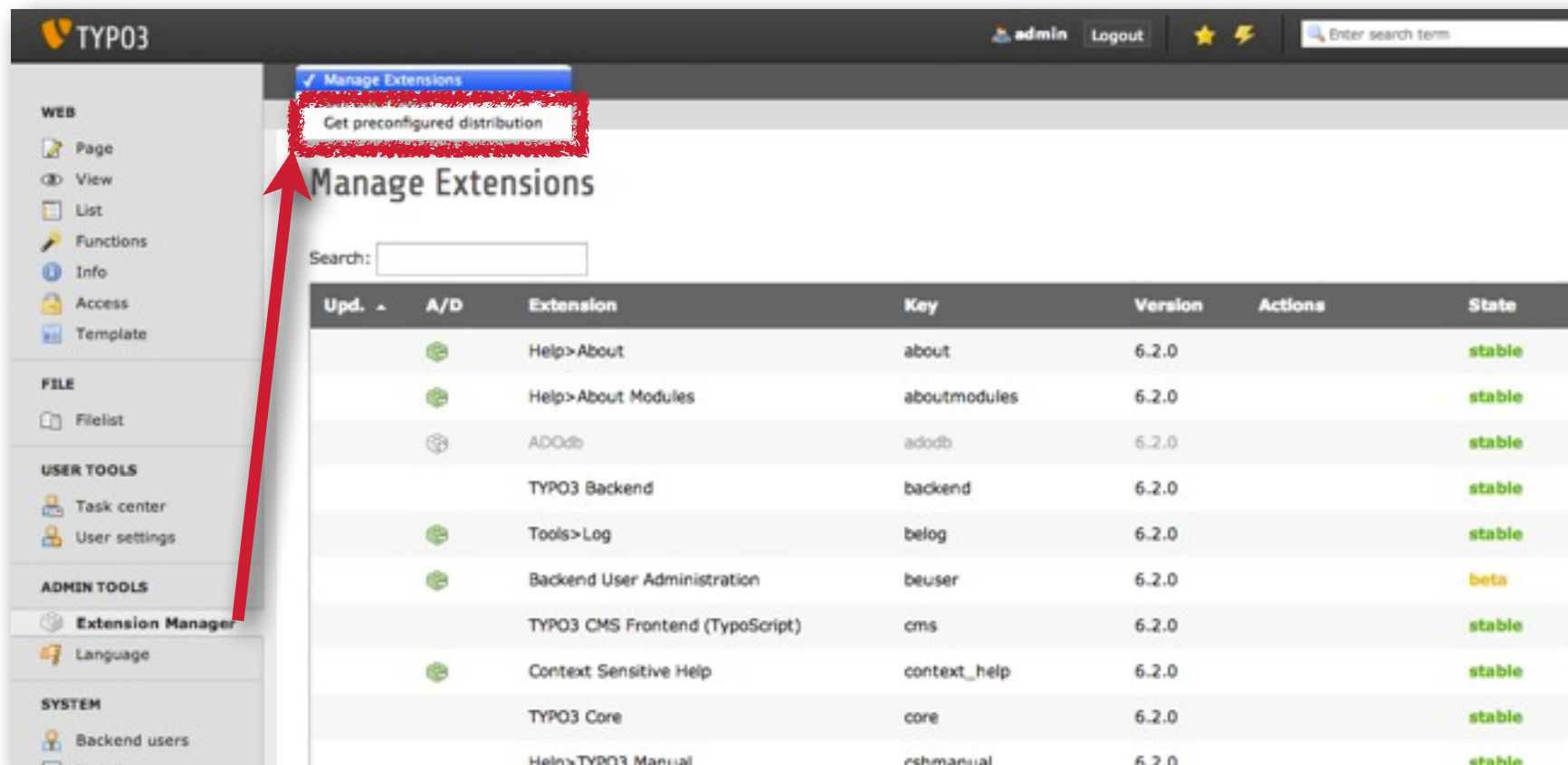
Install Tool - MISC 2

- Während normalerweise die Berechtigung für das Root-Verzeichnis der Installation 2770 sein muss und der Ordner dem Web-User gehören muss, um TYPO3 zu installieren, wurde nun eine Option „targetPermissionRelaxed“ eingeführt, bei der dieser Check für den Root-Folder außer Kraft gesetzt wurde, sofern es trotzdem möglich ist, die benötigten Unterverzeichnisse anzulegen

Distribution **Management**

Distribution Management

- Das Distribution Management Modul ist über den Extension Manager zu erreichen
- Dieser ist für das Laden von Distributionen wie „Introduction Package“ oder „Government Package“ zuständig



The screenshot shows the TYPO3 CMS 6.2 LTS Extension Manager interface. The left sidebar contains the following menu items: WEB (Page, View, List, Functions, Info, Access, Template), FILE (Filelist), USER TOOLS (Task center, User settings), ADMIN TOOLS (Extension Manager, Language), and SYSTEM (Backend users). The 'Extension Manager' item is highlighted. The main content area shows the 'Manage Extensions' page. A red box highlights the 'Get preconfigured distribution' button, and a red arrow points to it from the sidebar. The table below lists the installed extensions:

Upd.	A/D	Extension	Key	Version	Actions	State
		Help>About	about	6.2.0		stable
		Help>About Modules	aboutmodules	6.2.0		stable
		ADODB	adodb	6.2.0		stable
		TYPO3 Backend	backend	6.2.0		stable
		Tools>Log	belog	6.2.0		stable
		Backend User Administration	beuser	6.2.0		beta
		TYPO3 CMS Frontend (TypoScript)	cms	6.2.0		stable
		Context Sensitive Help	context_help	6.2.0		stable
		TYPO3 Core	core	6.2.0		stable
		Help>TYPO3 Manual	eshmanual	6.2.0		stable

Distribution Management

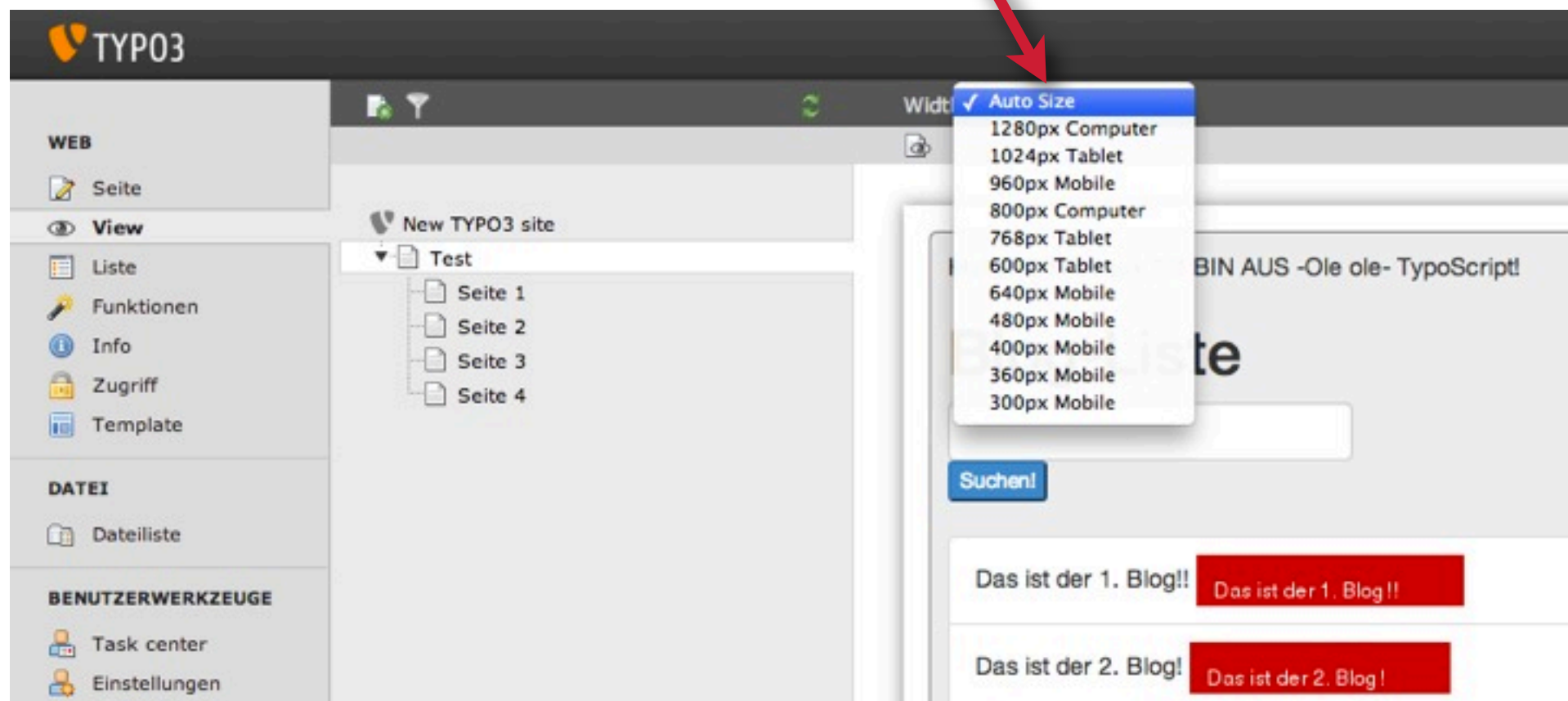
- Zum Start kann man das „Responsive Introduction Package“ (erstellt von Sven Wolfermann) und das „Government Package“ laden



TYPO3 CMS goes **Responsive**

Responsive Preview im Backend - 1

- Die Vorschau-Fläche im Modul „View“ kann nun per Dropdown in der Breite begrenzt werden, um so responsive Websites besser testen zu können



Responsive Preview im Backend - 2

- Die Framegrößen sind zudem konfigurierbar
- Dafür kann man folgenden Code per PageTSconfig verwenden

```
mod.web_view.previewFrameWidths {  
    1780.label = <any LLL or string>  
    1780.height = 145  
}
```

- Der Schlüssel (hier 1780) gibt dabei die Breite an
- Die Höhe (height) ist optional
- Die bereits enthaltenen Größen sind in der folgenden Datei definiert:

`typo3/sysext/core/Configuration/DefaultConfiguration.php`

```
mod.web_view.previewFrameWidths {  
    1280.label = LLL:EXT:viewpage/Resources/Private/Language/locallang.xlf:computer  
    1024.label = LLL:EXT:viewpage/Resources/Private/Language/locallang.xlf:tablet  
    ...  
}
```

Neue Attribute für responsive Bildergalerien

- Um responsive Bildergalerien zu erstellen, kann es helfen, wenn im Rendering von Bildern zusätzliche Attribute ausgegeben werden.
- Dafür wurde das CSC (CSS styled content) Rendering angepasst
- Im Falle des Doktypes html5 (`config.doctype = html5`), wird zusätzlich noch die Anzahl der Spalten und die Anzahl der Bilder ausgegeben

- Frühere Ausgabe:

```
<div class="csc-textpic-imagewrap">...</div>
```

Jetzige Ausgabe:

```
<div class="csc-textpic-imagewrap" data-csc-images="{register:imageCount}"  
data-csc-cols="{field:imagecols}">...</div>
```

Responsive Image Rendering - 1

- Das IMAGE cObject kann nun eine sogenannte **sourceCollection** rendern, um damit verschiedene Display-Auflösungen und Bildschirmgrößen zu unterstützen
- Um das Responsive-Rendering für die Content-Objekte „Bild“ und „Text/Bild“ einzuschalten, muss man zwei Einstellungen im Constant-Editor treffen:

Responsive Rendering verwenden:

`styles.content.imgtext.responsive`

Rendering-Type:

`styles.content.imgtext.layoutKey`

Default img-tag = **default**

img-tag with alternate sources as srcset-attribute = **srcset**

picture-tag with source-child-tags = **picture**

img-tag with alternate sources as data-attributes = **data**

Responsive Image Rendering - 2

- **Eigenschaft: `layoutKey`**
 - Definiert das Render-Layout des IMAGE
 - Das Render-Layout ist der HTML-Code, der für das IMAGE verwendet wird
 - Mögliche ("out of the box") Werte sind: default, srcset, picture, data
 - Jede Möglichkeit repräsentiert eine unterschiedliche Lösung um HTML-Code für das IMAGE zu rendern
 - Der Default-Code rendert das ``-Tag auf herkömmliche Weise mit verschiedenen Attributen
 - Wenn man ein Responsive Layout implementiert, benötigt man unterschiedliche Bild-Größen für unterschiedliche Auflösungen und Bildschirmgrößen
 - Abhängig vom verwendeten HTML-Framework, den Fähigkeiten der gewünschten Browser und dem verwendeten JavaScript für das progressive enhancement, kann man entweder eines der vordefinierten Layouts verwenden oder ein neues erstellen, indem man einen anderen `layoutKey` benutzt
 - Wenn man kein Responsive Layout benötigt, sollte man das Default-Layout verwenden

Responsive Image Rendering - 3

- Eigenschaft: **layout**

- layoutKey** = {\$styles.content.imgtext.layoutKey}
 - layout** {
 - default** {
 - element =
 - // <http://www.w3.org/html/wg/drafts/srcset/w3c-srcset/>
 - srcset** {
 - element =
 - source = |*|###SRC### ###SRCSETCANDIDATE###,|*|###SRC### ###SRCSETCANDIDATE###
 - // <http://www.w3.org/TR/html-picture-element/>
 - picture** {
 - element = <picture>###SOURCECOLLECTION###</picture>
 - source = <source src="###SRC###" media="###MEDIAQUERY###"###SELCLOSINGTAGSLASH###>
 - data** {
 - element =
 - source = data-###DATAKEY###="###SRC###"

Responsive Image Rendering - 4

- Eigenschaft: **layout.[layoutKey].element**
 - **###SRC###**
URL für das src-Attribut
 - **###WIDTH###**
Breite des Bildes für das width-Attribut
 - **###HEIGHT###**
Höhe des Bildes für das height-Attribut
 - **###PARAMS###**
Zusätzliche Parameter (wie im IMAGE object definiert)
 - **###ALTPARAMS###**
Zusätzliche alternative Parameter (wie im IMAGE object definiert)
 - **###BORDER###**
Border-Attribut des -Tags
 - **###SELF-CLOSING-TAG-SLASH###**
Rendert das Tag-Abschlußzeichen (abhängig von config.xhtmlDoctype bzw. config.Doctype)
 - **###SOURCECOLLECTION###**
Damit werden die zusätzlichen Quellen des Bildes abhängig von der unterschiedlichen Verwendung im Responsive Web Design definiert. Die Definition selbst wird im Schlüssel **layout.[layoutKey].source** vorgenommen.

Responsive Image Rendering - 5

- Eigenschaft: **sourceCollection.[dataKey]**
 - Dies ist die Default-SourceCollection (der CSS Styled Content Extension)
 - Es ist unbedingt zu empfehlen, eine eigene zu schreiben
 - **sourceCollection** {
 - small** {
 - width = 200
 - srcsetCandidate = 600w
 - mediaQuery = (max-device-width: 600px)
 - dataKey = small
 - smallRetina** {
 - if.directReturn = 1
 - width = 200
 - pixelDensity = 2
 - srcsetCandidate = 600w 2x
 - mediaQuery = (max-device-width: 600px) AND (min-resolution: 192dpi)
 - dataKey = smallRetina
- **Optionen:** dataKey, if, pixelDensity, width, height, maxW, maxH, minW, minH, linkWrap, srcsetCandidate, mediaQuery

Responsive Image Rendering - 6a - Beispiel

```

page.30 = IMAGE
page.30 {
    file = fileadmin/rocky.jpg
    file.width = 300
    layoutKey = default
    layout {
        default {
            element = 
            source =
        }
        srcset {
            element = 
            source = |*|###SRC### ###SRCSETCANDIDATE###,|*|###SRC### ###SRCSETCANDIDATE###
        }
        picture {
            element = <picture>###SOURCECOLLECTION###</picture>
            source = <source src="###SRC###" media="###MEDIAQUERY###" ###SELCLOSINGTAGSLASH###>
        }
        data {
            element = 
            source.noTrimWrap = | data-###DATAKEY###="###SRC###"|
        }
    }
}

```

Responsive Image Rendering - 6b - Beispiel

```
sourceCollection {
  small {
    width = 200
    srcsetCandidate = 800w
    mediaQuery = (min-device-width: 800px)
    dataKey = small
  }
  smallHires {
    if.directReturn = 1
    width = 300
    pixelDensity = 2
    srcsetCandidate = 800w 2x
    mediaQuery = (min-device-width: 800px) AND (min-resolution: 192dpi)
    dataKey = smallHires
    pictureFoo = bar
  }
}
page.40 < page.30
page.40.layoutKey = data
page.50 < page.30
page.50.layoutKey = picture
page.60 < page.30
page.60.layoutKey = srcset
```

Responsive Image Rendering - 6b - Beispiel

HTML-Ausgabe:

```




<picture>
  <source src="fileadmin/_processed_/csm_rocky_ccc7ea1d33.jpg" media="(min-device-width: 800px)" >
  <source src="fileadmin/_processed_/csm_rocky_29113e55a7.jpg" media="(min-device-width: 800px) AND (min-resolution:
192dpi)" >
  
</picture>


```

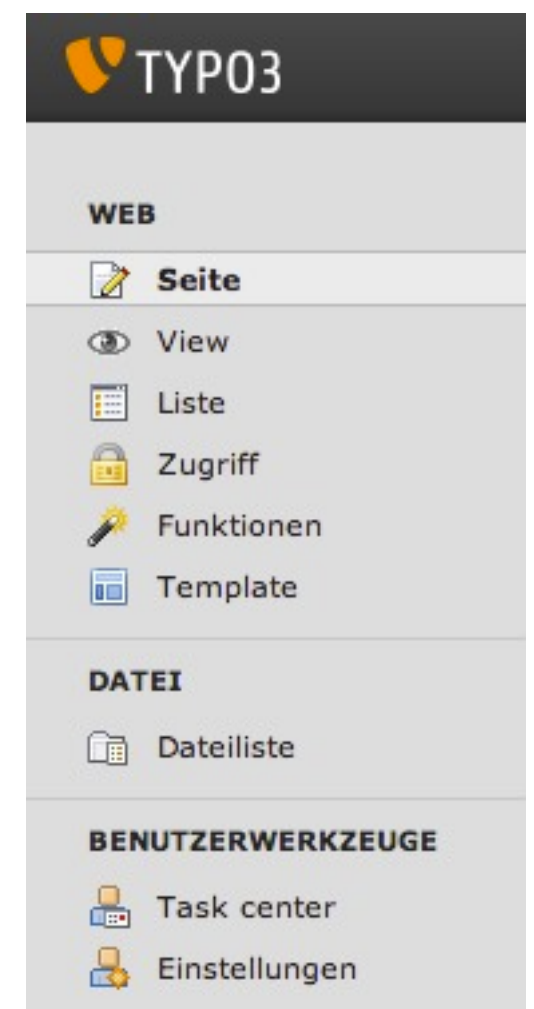
Änderungen im **Backend**

Mehr Platz im Backend

- Im Backend wurden viele Bereiche vom Platz her aufgelockert, um damit für eine bessere Usability zu sorgen
- Dafür wurden horizontale und vertikale Abstände vieler Elemente vergrößert
- Basis ist ein 12px Raster, welches verdoppelt wurde



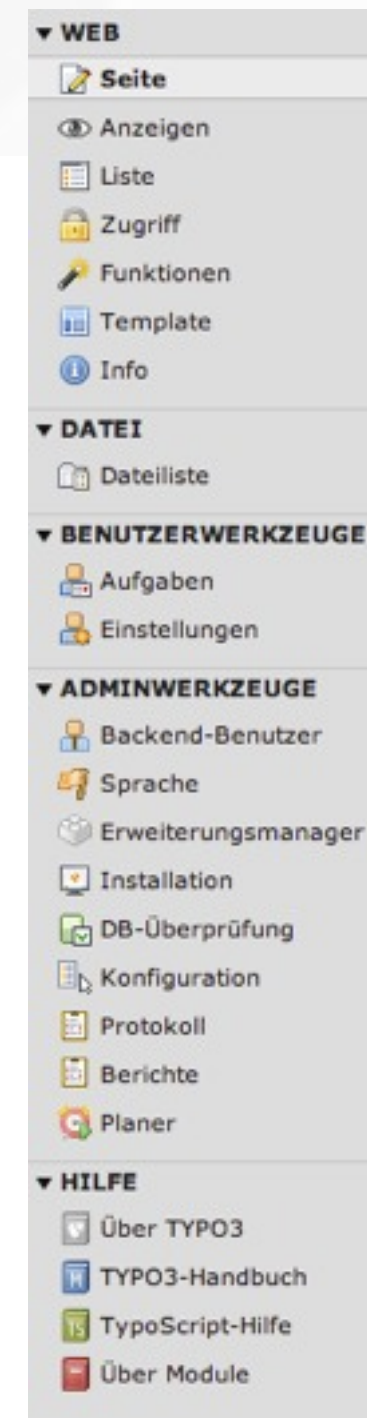
TYPO3 < 6.2



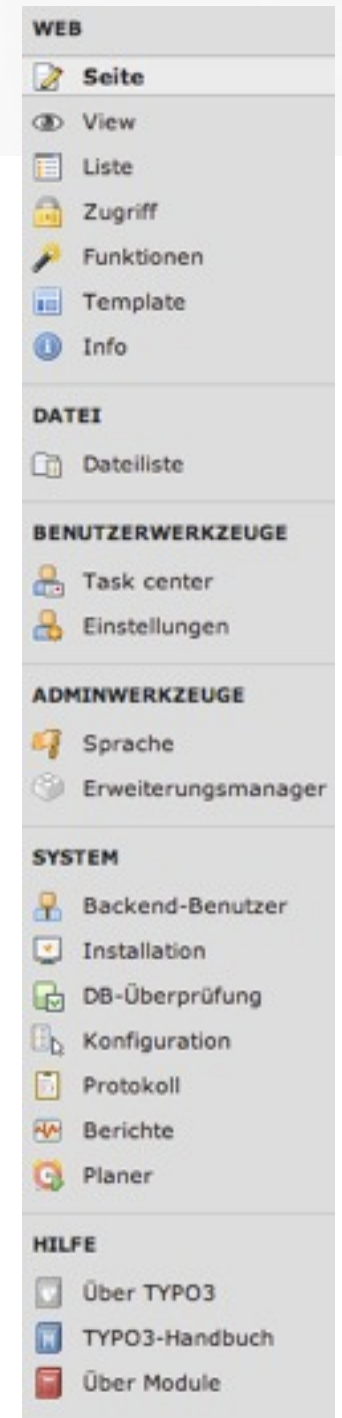
TYPO3 >= 6.2

Neue Anordnung der Module

- Die Module auf der linken Seite wurden neu angeordnet
- Insbesondere wurde das Modul „SYSTEM“ eingeführt, welches die meisten „LowLevel-Statustools“ beherbergt, die zudem keinen Seitenbaum benötigen
- Unter „ADMINWERKZEUGE“ wurden nun die Module „Sprache“ und „Erweiterungsmanager“ einsortiert
- Das Modul „TypoScript-Hilfe“ wurde (da veraltet) komplett entfernt



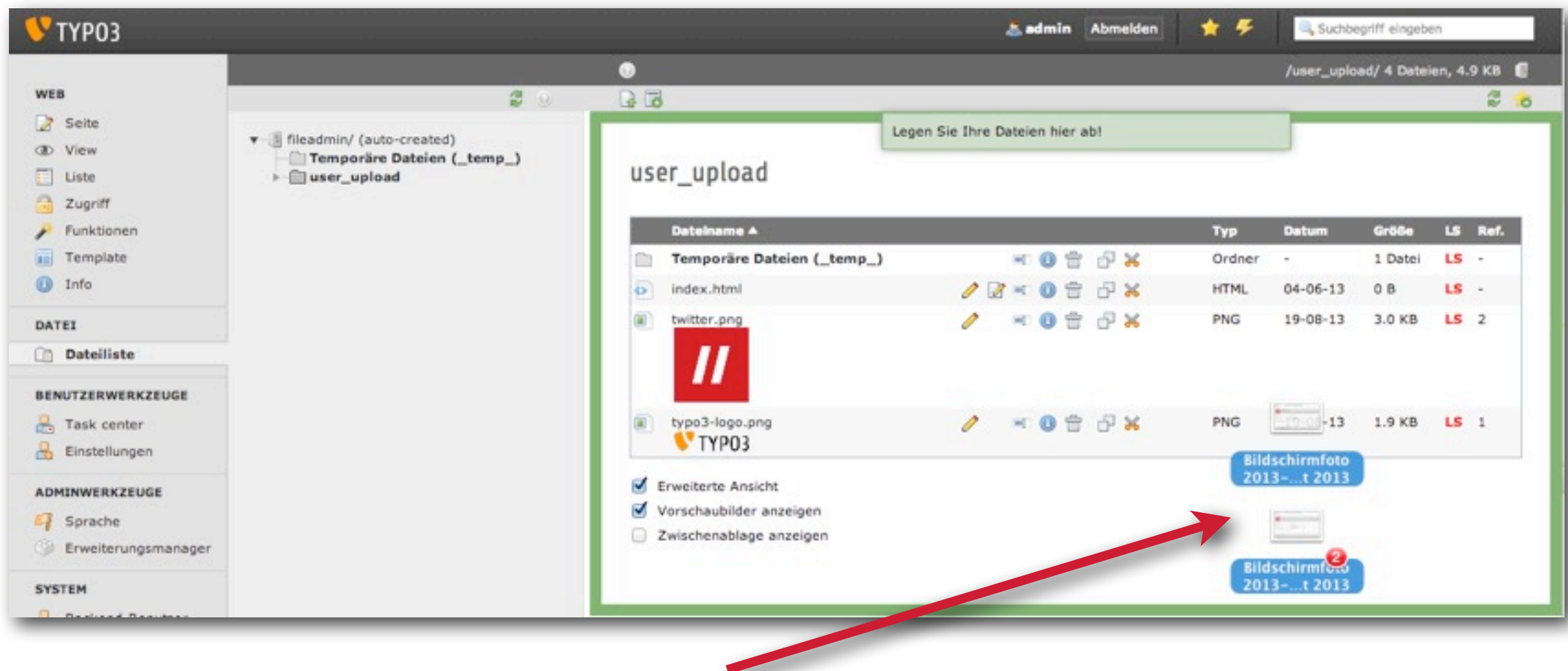
TYPO3 < 6.2



TYPO3 >= 6.2

Drag&Drop in der Datei-Liste

- Dateien können nun einfach per Drag & Drop in die Datei-Liste gelegt werden (Frage nach Überschreibung)













Bessere Usability im Modul Backend Benutzer

- Einerseits wurde der Benutzername und der Name nun auf der linken Seite gruppiert und andererseits sind diese nun klickbar (und führen zum Editier-Formular). Zudem gibt es nun einen Delete-Button in der Liste.

Benutzername	Name			Letzte Anmeldung
 admin	Patrick Lobacher	 Vergleichen	  	20-08-13 07:28
 advanced_editor	Advanced McEditor	 Vergleichen	    	Nie

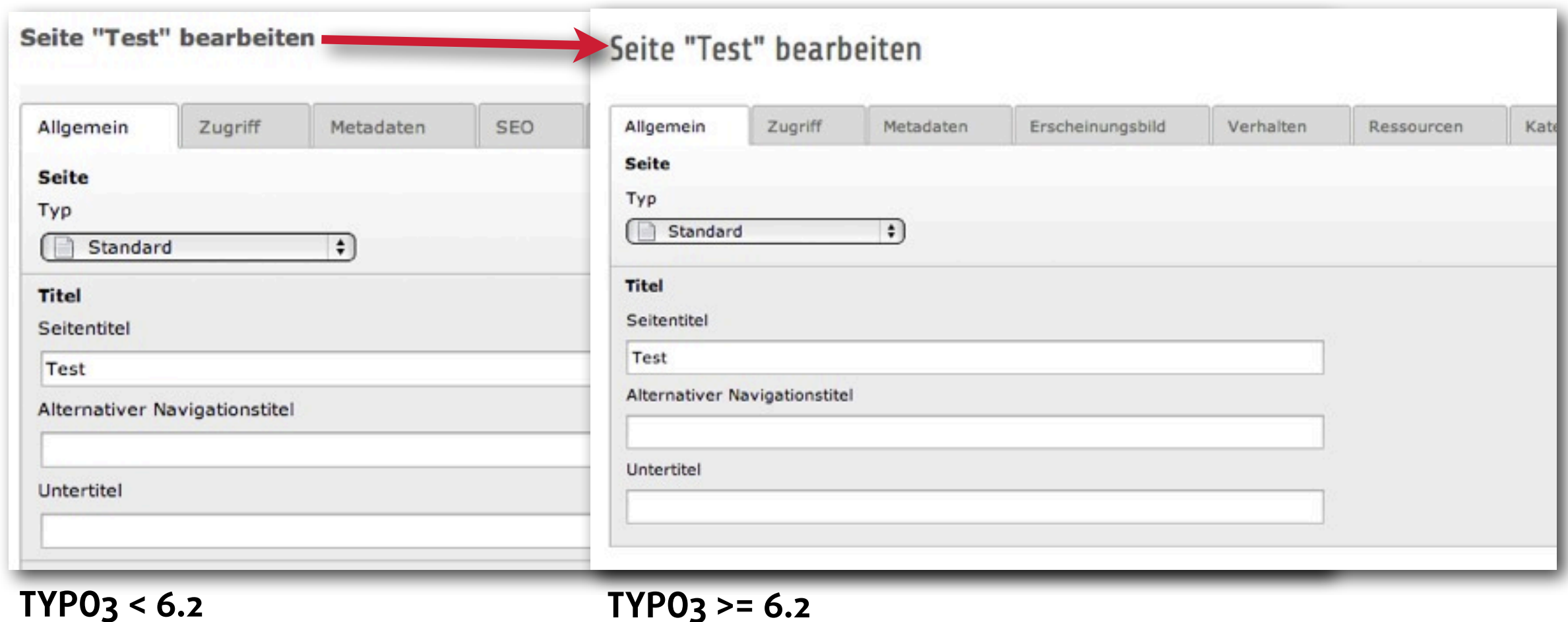
TYPO3 < 6.2

Benutzername / Name				Letzte Anmeldung
 <u>admin</u>	<u>Patrick Lobacher</u>	 <u>Vergleichen</u>	  	20-08-13 07:23
 advanced_editor	Advanced McEditor	 Vergleichen	    	Nie
2 Benutzer				

TYPO3 >= 6.2

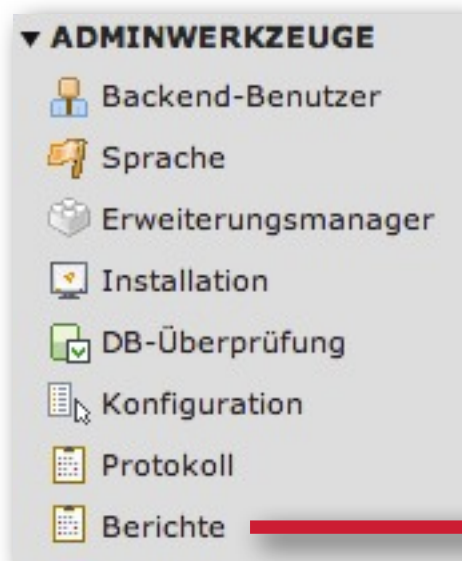
H1 im CI-Font Share

- Die H1-Überschriften im rechten Bereich werden nun durchwegs mit dem CI-Font „Share“ dargestellt

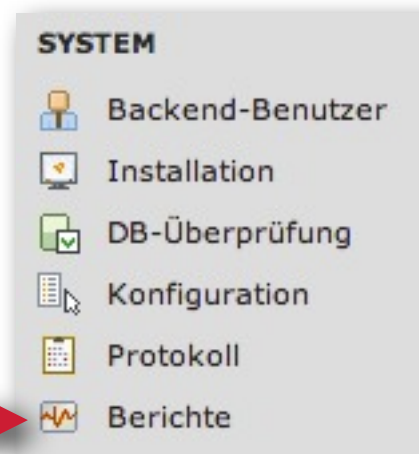


Neues Symbol für Modul „Reports“

- Das Modul „Reports“ hat ein eigenes Symbol erhalten
- Bislang wurde dafür das Symbol des Moduls „Log“ verwendet.



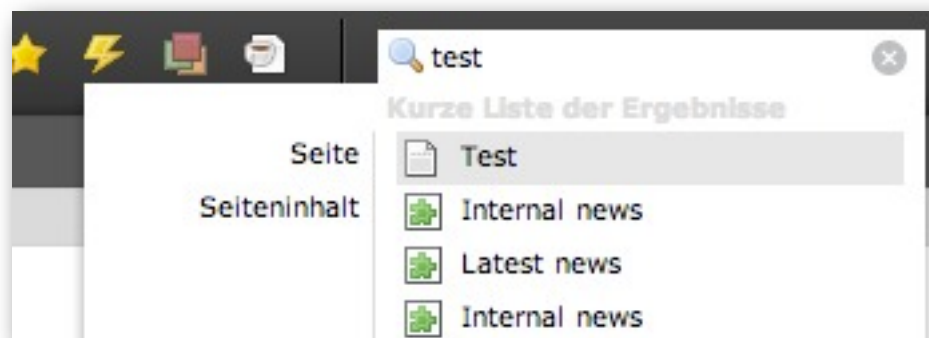
TYPO3 < 6.2



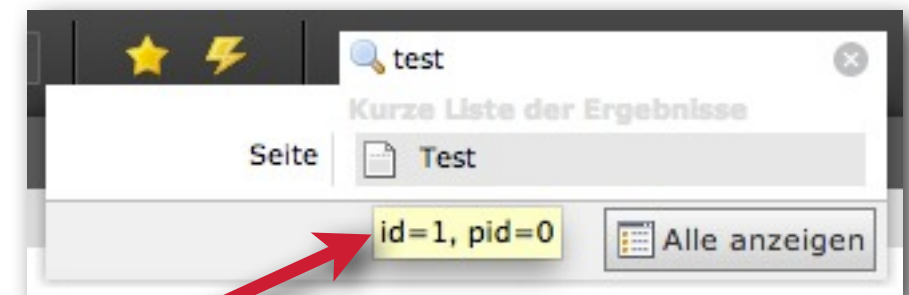
TYPO3 >= 6.2

Verbesserte Suche - UID und PID im Ergebnis

- Die Live-Suche rechts oben kann nun im Tooltip sowohl UID wie auch PID anzeigen
- Sobald man nach einer Suche das Editierformular wieder schließt, landet man in der Listen-Ansicht der PID (und nicht wie früher in einer leeren Seite)



TYPO3 < 6.2

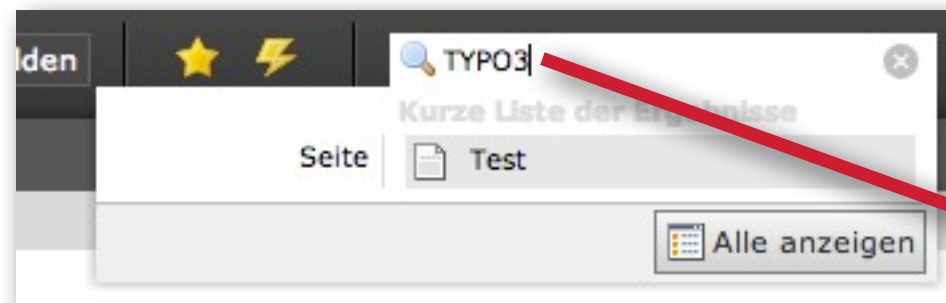


TYPO3 >= 6.2

Verbesserte Suche - Alias kann durchsucht werden

- Bislang durchsuchte die Suche rechts oben für Seiten nur die Felder `title` und `uid`
- Zusätzlich kann man nun das Feld `alias` durchsuchen, wenn dies per `UserTSconfig` konfiguriert wurde:

`options.pageTree.searchInAlias = 1`

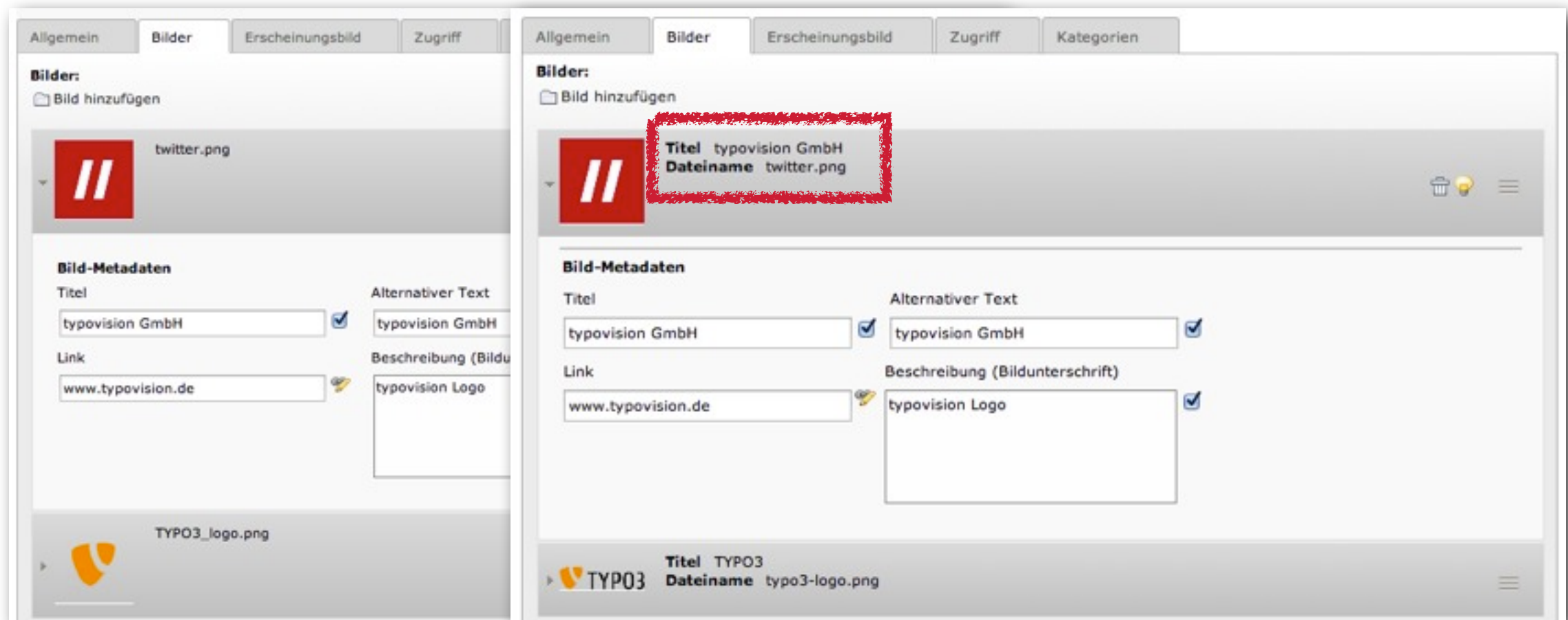


Seite "Test" bearbeiten

Allgemein	Zugriff	Metadaten	Erscheinungsbild	Verhalten
Links zu dieser Seite				
URL-Alias				
<input type="text" value="typo3"/>				
Linkziel				
<input type="text"/>				
Benutze Protokoll				
<input type="button" value="Standard"/> ▾				

Meta-Informationen im FAL-Element Header

- Dateiname und Titel werden nun im FAL-Element Header inkl. Überschrift angeben

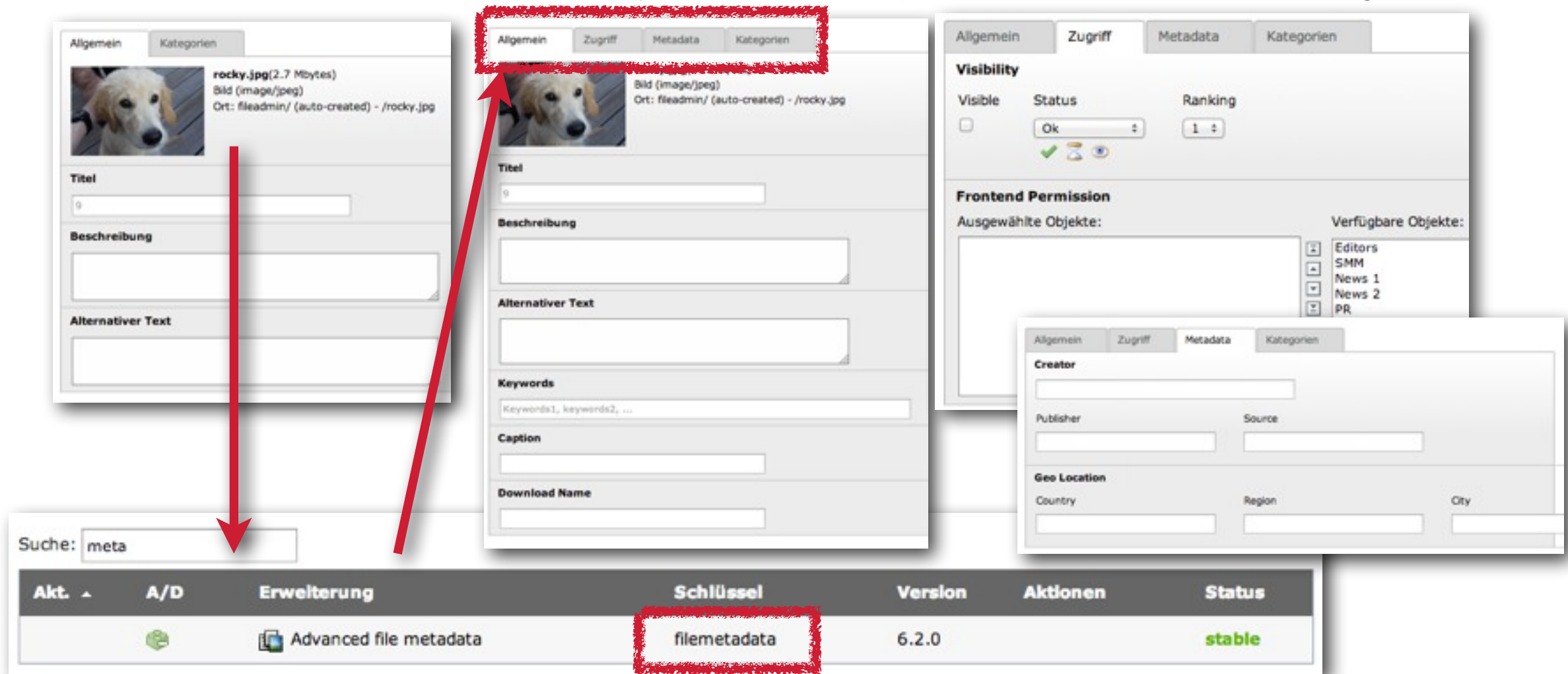


TYPO3 < 6.2


TYPO3 >= 6.2

FAL - erweiterte Meta-Informationen

- Über die Extension „filemetadaten“ (nicht per Default installiert) werden zusätzliche Meta-Daten eingebracht

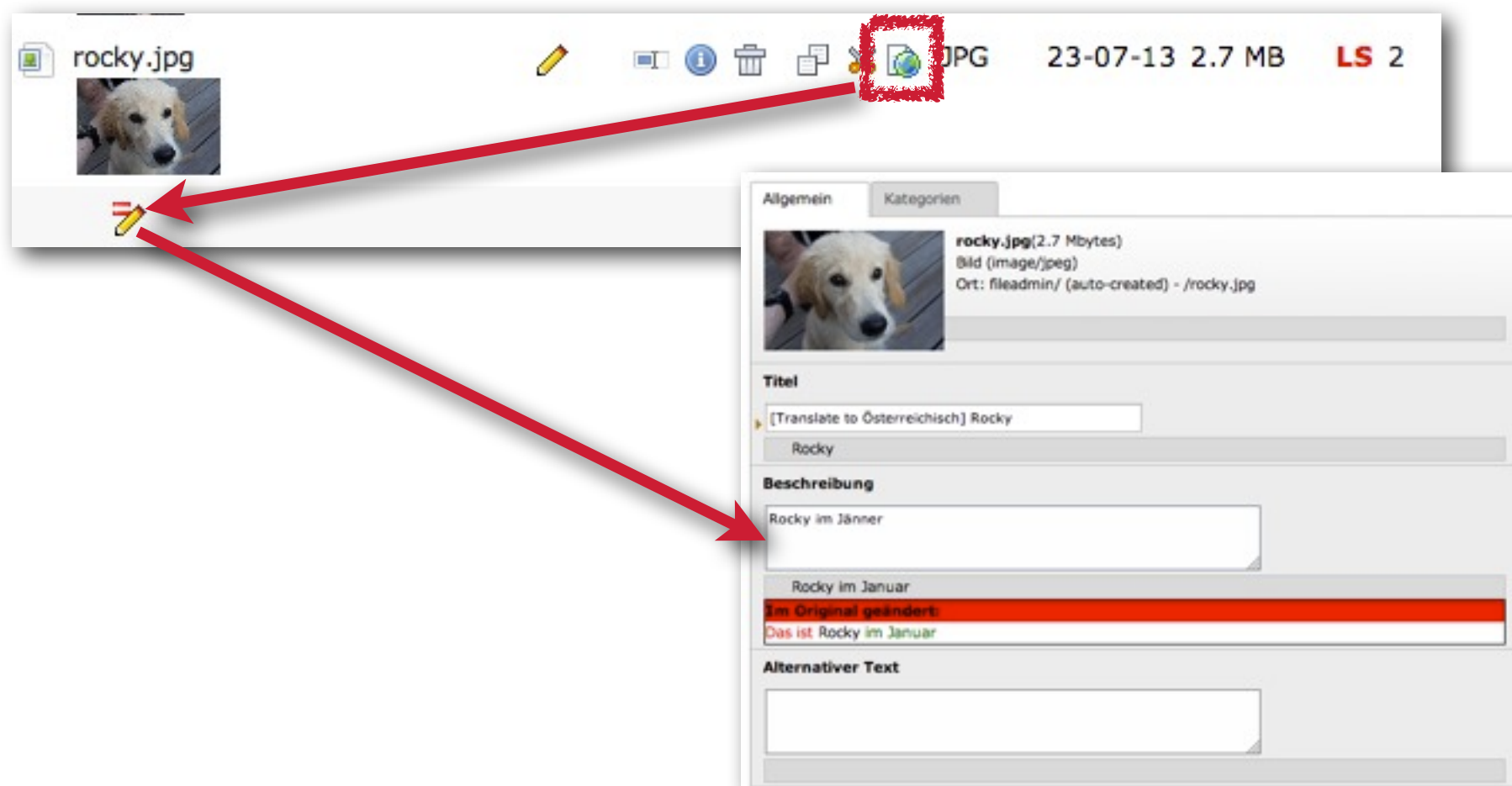


The screenshot illustrates the FAL (File Abstraction Layer) interface in the TYPO3 CMS 6.2 LTS Backend. The main window shows the 'Allgemein' (General) tab for a file named 'rocky.jpg'. The 'Allgemein' tab is highlighted with a red box. Below the main window, a search bar shows 'Suche: meta'. At the bottom, a table lists installed extensions, with 'filemetadaten' highlighted by a red box. To the right, a 'Frontend Permission' dialog is visible, showing a list of available objects including 'Editors', 'SMM', 'News 1', 'News 2', and 'PR'.

Akt.	A/D	Erweiterung	Schlüssel	Version	Aktionen	Status
		Advanced file metadata	filemetadaten	6.2.0		stable

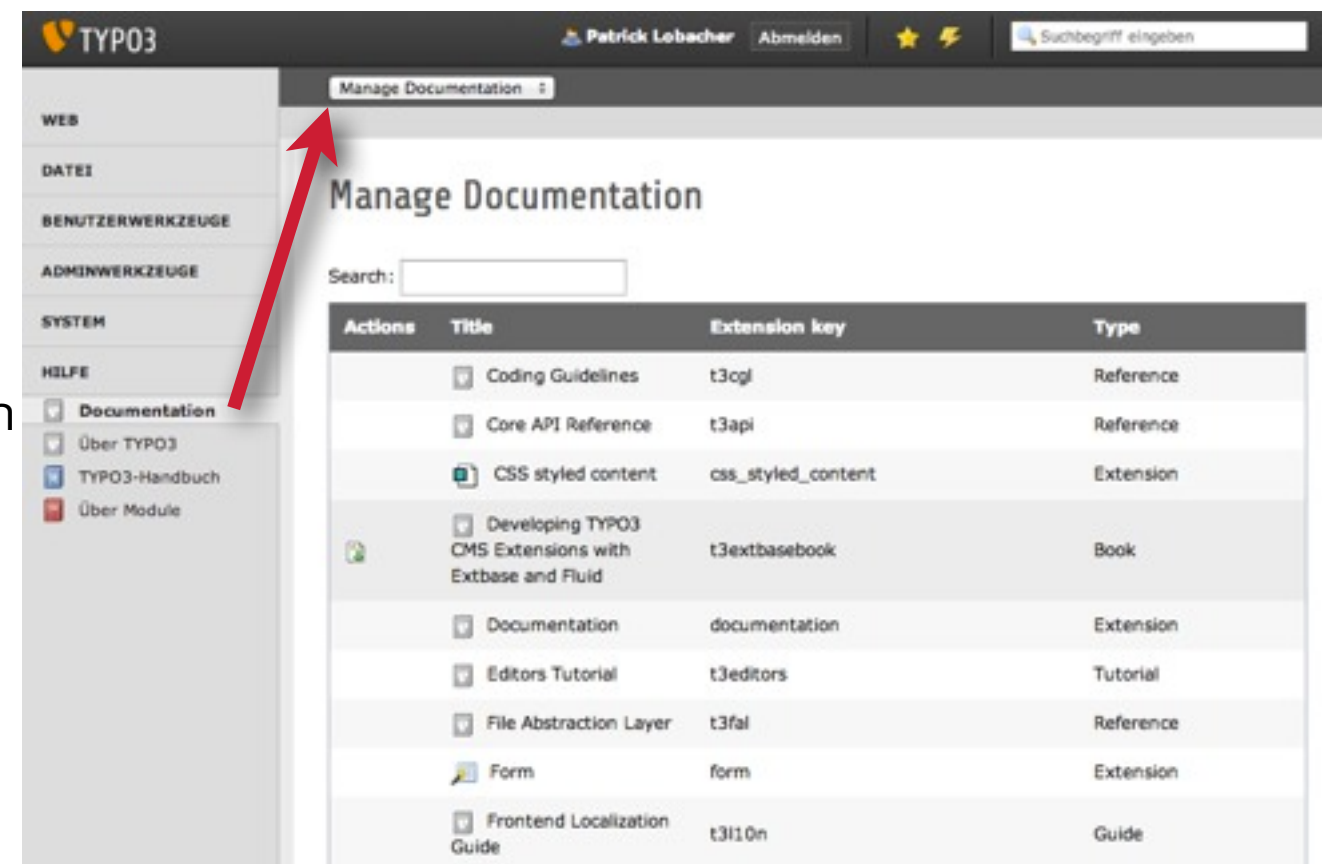
FAL Metadaten können übersetzt werden

- Die FAL Metadaten können nun in die konfigurierten Frontend-Sprachen übersetzt werden



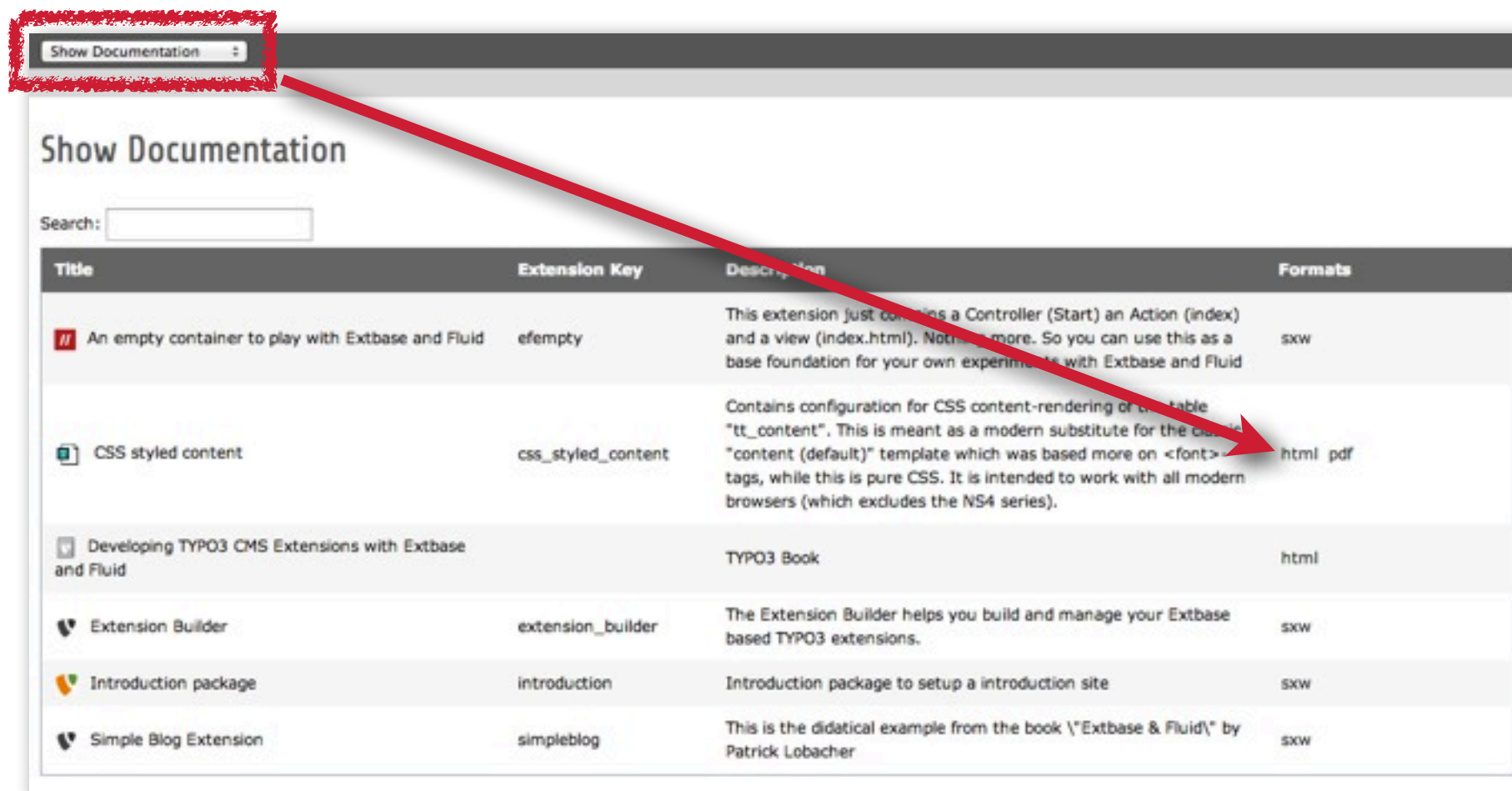
Modul Documentation hinzugefügt - 1

- Es wurde ein Modul „Documentation“ hinzugefügt, mit dem es möglich ist, Manuals herunterzuladen und anzusehen
- Das Modul wird bei neuen Installation per Default geladen - ansonsten kann man es im Erweiterungsmanager installieren
- Im Modul „Manage Documentation“ kann man Dokumentationen auf das System herunterladen
- Im Modul „Show Documentation“ kann man diese dann ansehen



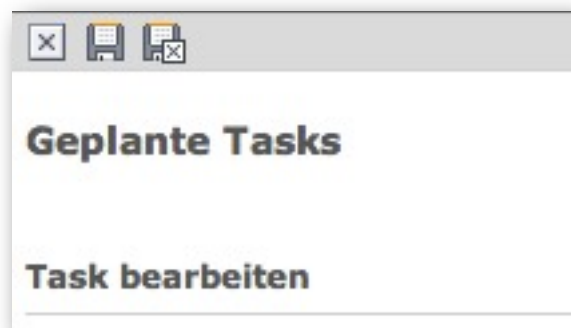
Modul Documentation hinzugefügt - 2

- Im Modul „Show Documentation“ kann man die Dokumentationen der Erweiterungen dann ansehen

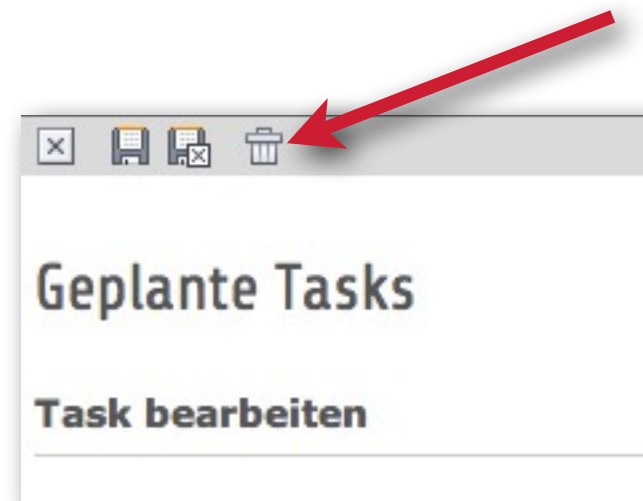


Scheduler-Task löschen im Edit-Fenster

- Bislang konnte man Scheduler-Tasks nur aus der Liste der Tasks heraus löschen
- Nun gibt es auch eine Möglichkeit zum Löschen aus dem Edit-Fenster heraus



TYPO3 < 6.2



TYPO3 >= 6.2

Beschreibung bei Scheduler Tasks

- Scheduler Tasks kann man nun mit einer Beschreibung ausstatten (als Subheadline oder Hover)

Task bearbeiten

Deaktivieren ☐

Klasse Erweiterungsliste aktualisieren (extensionmanager)

Typ

Start (HH:MM TT-MM-YYYY)

Ende (HH:MM TT-MM-YYYY)



Häufigkeit (Sekunden oder Cron-Befehl)

Parallele Ausführung erlauben ☐

Beschreibung

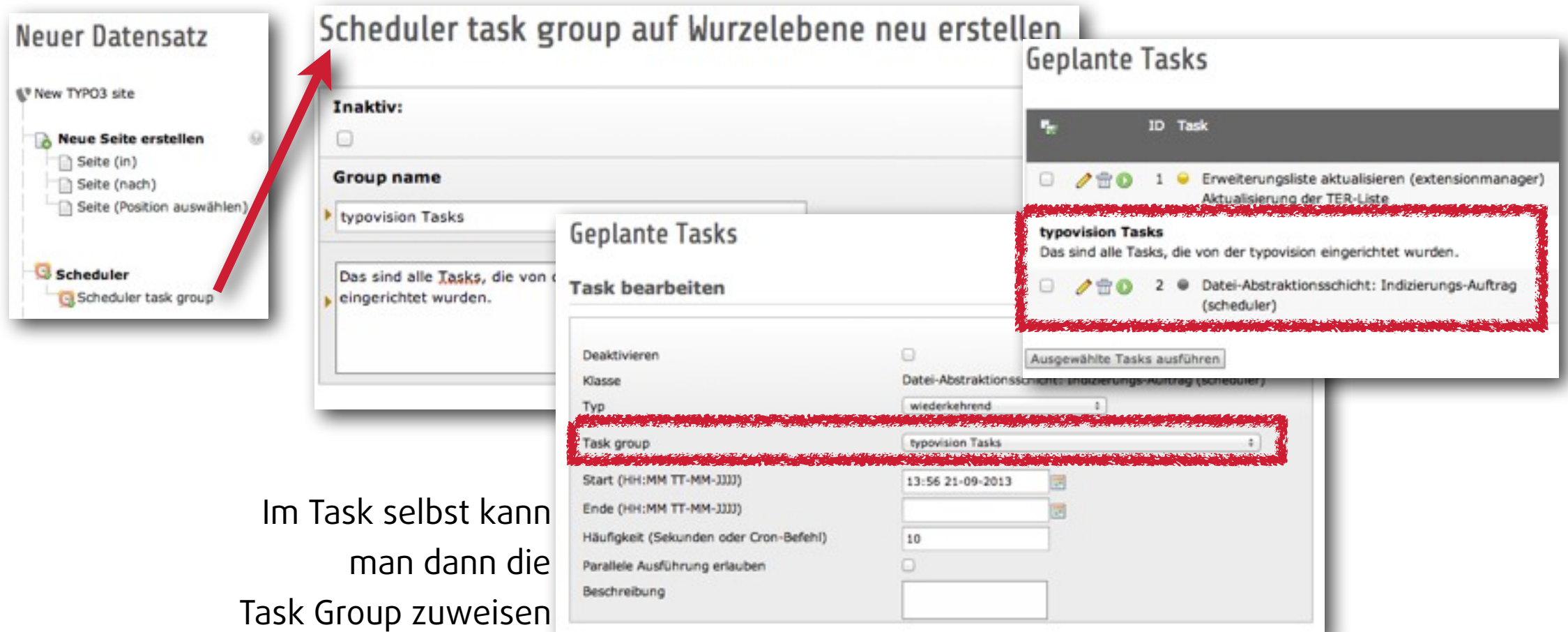
In der Extension
Konfiguration
kann eingestellt werden,
ob die Beschreibung als
Hover angezeigt wird

Show description of tasks as hover [basic.listShowTaskDescriptionAsHover]
If disabled descriptions of tasks will be shown below the job-name in the scheduler-list
☒ (bookesch)

ID	Task	Typ	Häufigkeit	Parallele Ausführung	Letzte Ausführung	Nächste Ausführung
<input type="checkbox"/>	2  Datei-Abstraktionsschicht: Indizierungs-Auftrag	wiederkehrend	10	Nein	21-09-13 13:57 (manuell)	21-09-13 13:57
<input type="checkbox"/>	1  Erweiterungsliste aktualisieren (extensionmanager) Aktualisierung der TER-Liste	wiederkehrend	5	Nein	-	16-10-13 07:17

Scheduler-Task Gruppen

- Man kann nun Scheduler-Tasks in Gruppen anordnen
- Dafür muss man einen Datensatz vom Typ „Scheduler task group“ auf der Root-Page (UID=0) anlegen



The image displays three overlapping screenshots from the TYPO3 CMS 6.2 LTS Backend:

- Left Screenshot:** The 'Neuer Datensatz' (New Record) screen for the 'Scheduler' module. It shows a tree structure with 'Scheduler task group' selected. A red arrow points from this selection to the middle screenshot.
- Middle Screenshot:** The 'Scheduler task group auf Wurzelebene neu erstellen' (Create new Scheduler task group on root level) form. It shows the 'Group name' field with 'typovision Tasks' entered. Below, a message states: 'Das sind alle Tasks, die von der typovision eingerichtet wurden.' (These are all tasks set up by typovision).
- Right Screenshot:** The 'Geplante Tasks' (Scheduled Tasks) overview. It shows a table with two tasks:

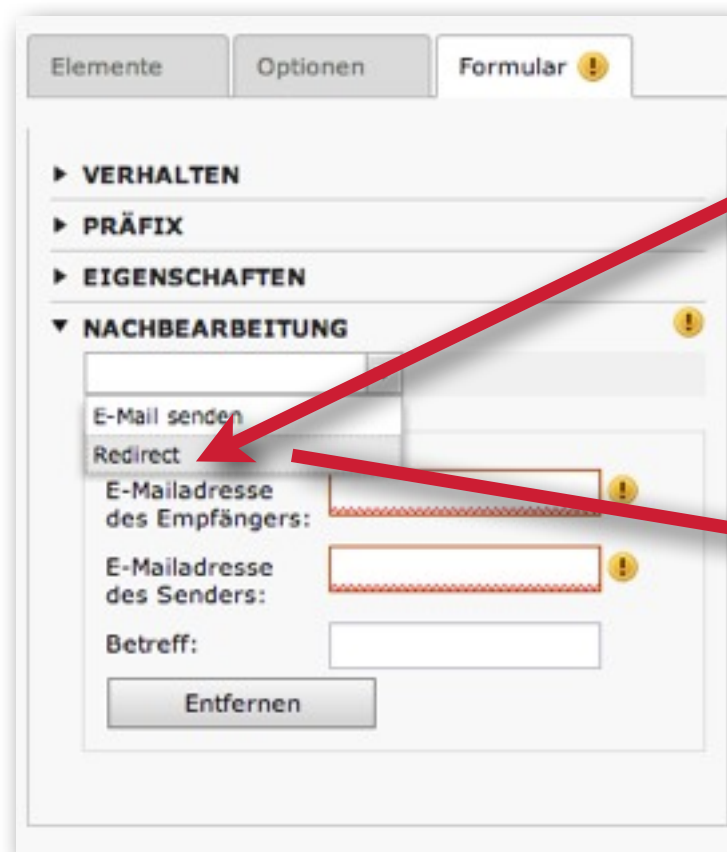
ID	Task
1	Erweiterungsliste aktualisieren (extensionmanager) Aktualisierung der TER-Liste
2	Datei-Abstraktionsschicht: Indizierungs-Auftrag (scheduler)

The task with ID 2 is highlighted with a red dashed box. Below the table, a button 'Ausgewählte Tasks ausführen' (Execute selected tasks) is visible.

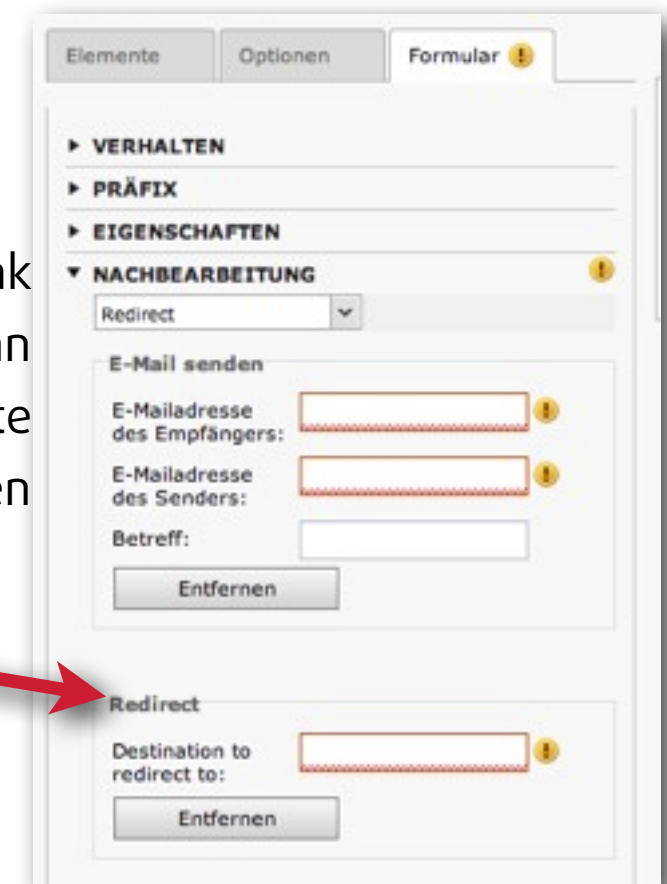
Below the middle screenshot, a text box explains: 'Im Task selbst kann man dann die Task Group zuweisen' (In the task itself, you can then assign the task group).

Postprozessor Redirect für Formulare

- Bislang gab es nur einen Postprozessor Email für mit dem FORM Content Objekt erstellte Formulare
- Ab sofort gibt es noch einen weiteren Postprozessor: **Redirect (Weiterleitung nach Abschicken der Mail)**

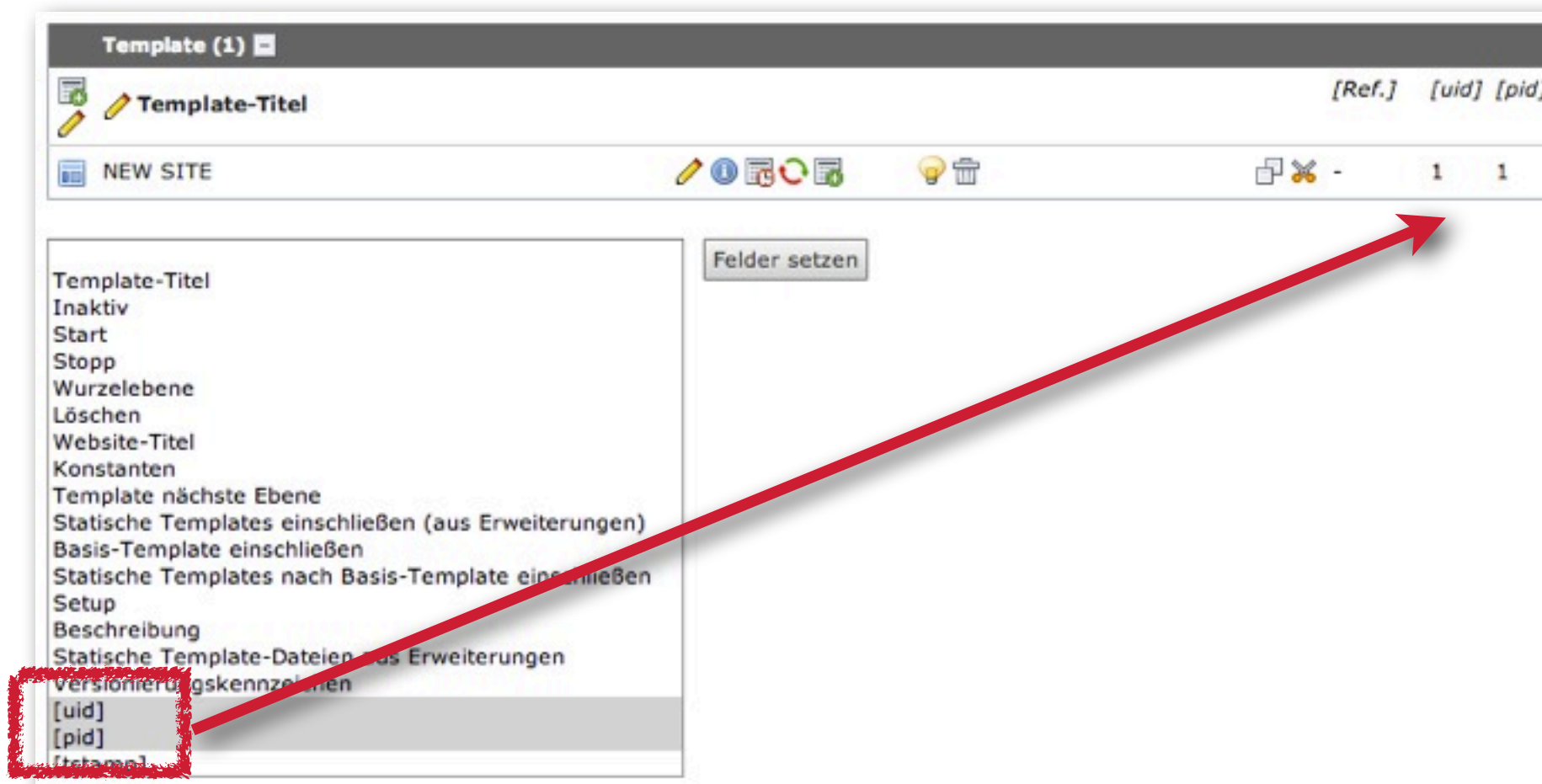


Das Feld wird über typolink
ausgewertet - d.h. man kann
z.B. eine ID für eine interne Seite
oder einen URL angeben



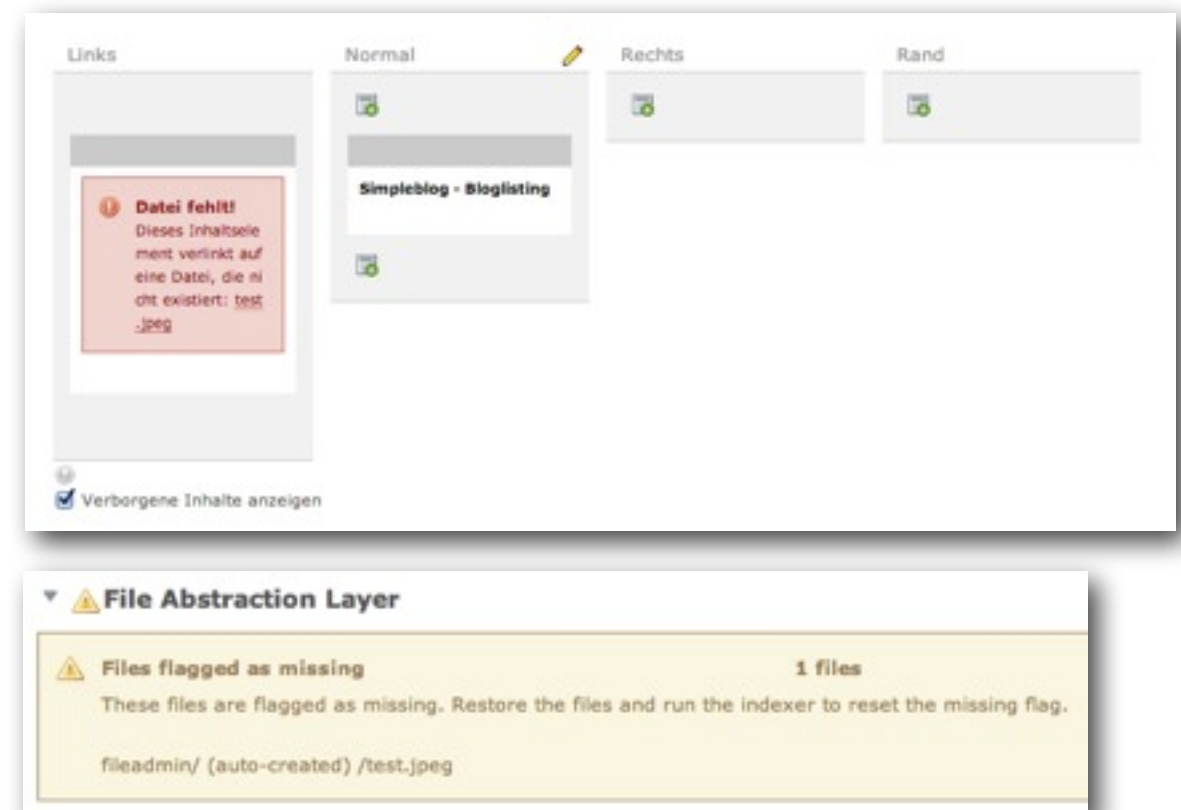
UID und PID auswählbar in List-View für User

- Im Modul List kann nun auch der normale User (Nicht-Admin) die Spalten [uid] und [pid] auswählen



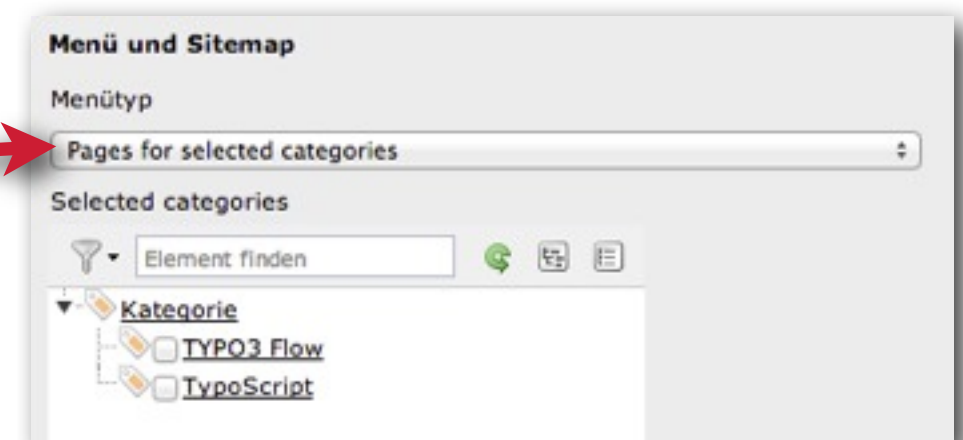
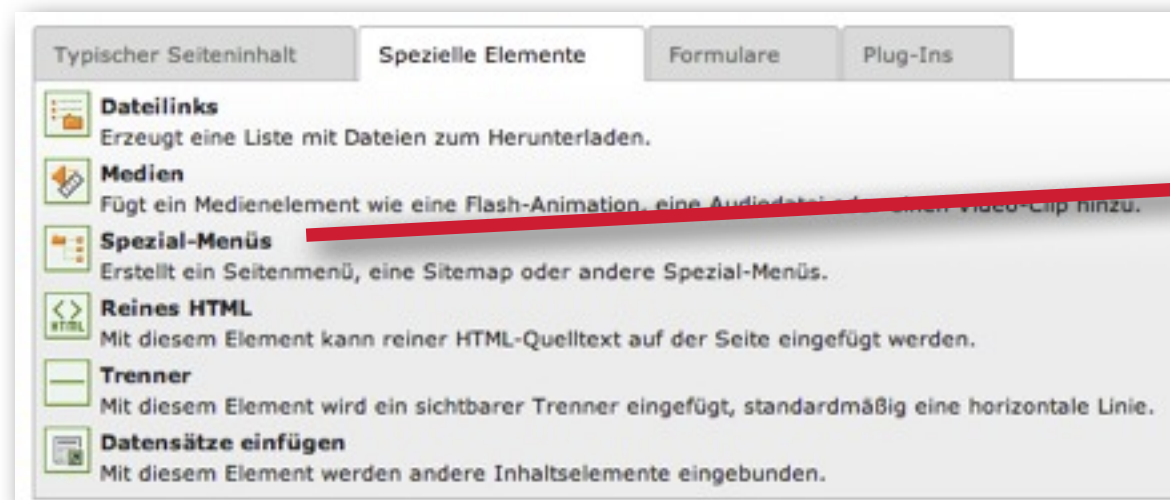
Fehlende Dateien werden angezeigt

- Sobald der FAL-Indexer auf eine Datei trifft, die nicht mehr vorhanden ist, wird ein entsprechender Hinweis im Backend ausgegeben. Zusätzlich wird das Flag „missing“ in der Datenbank „sys_file“ gesetzt. Sobald die Datei wieder vom Indexer gefunden wird, verschwinden Flag und Hinweis. Reports Modul dito.

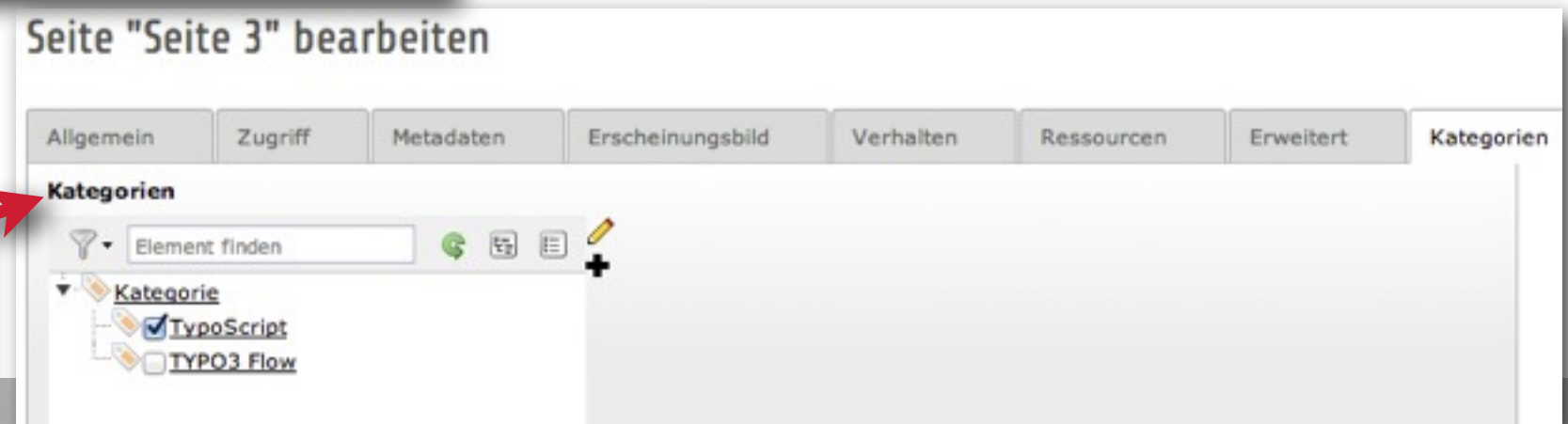
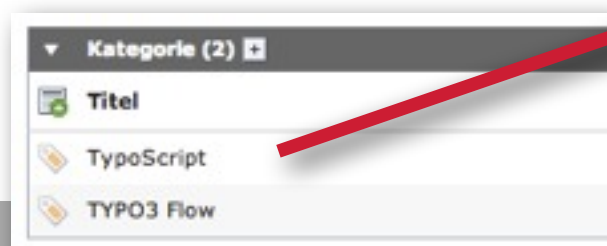


Neuer Menü-Typ: categories

- Man kann nun ein Menü - bestehend aus Elementen aus Kategorien - erstellen.
- Dafür gibt es ein spezielles Menü-Content-Objekt

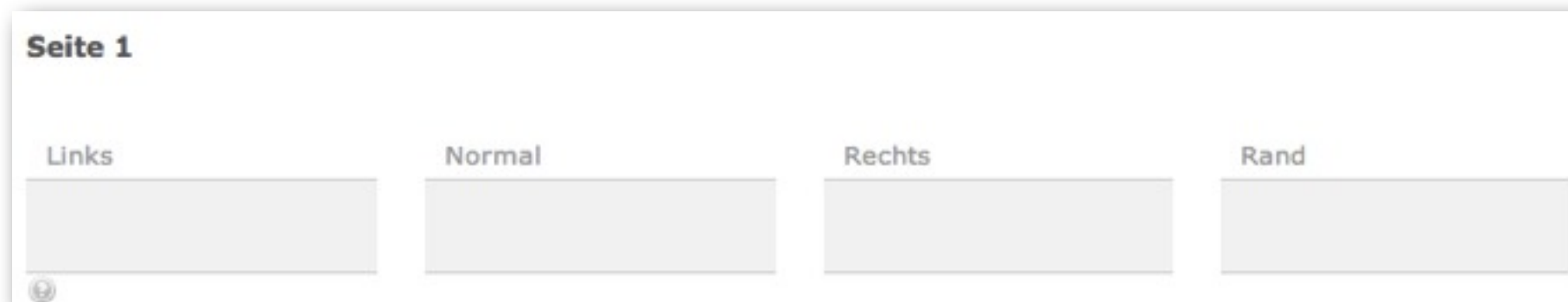


Kategorien zuweisen



Icon „Neues Inhaltselement“ immer sichtbar

- Das Icon ist nun bei leeren Spalten immer sichtbar (und nicht nur beim Drüberfahren)



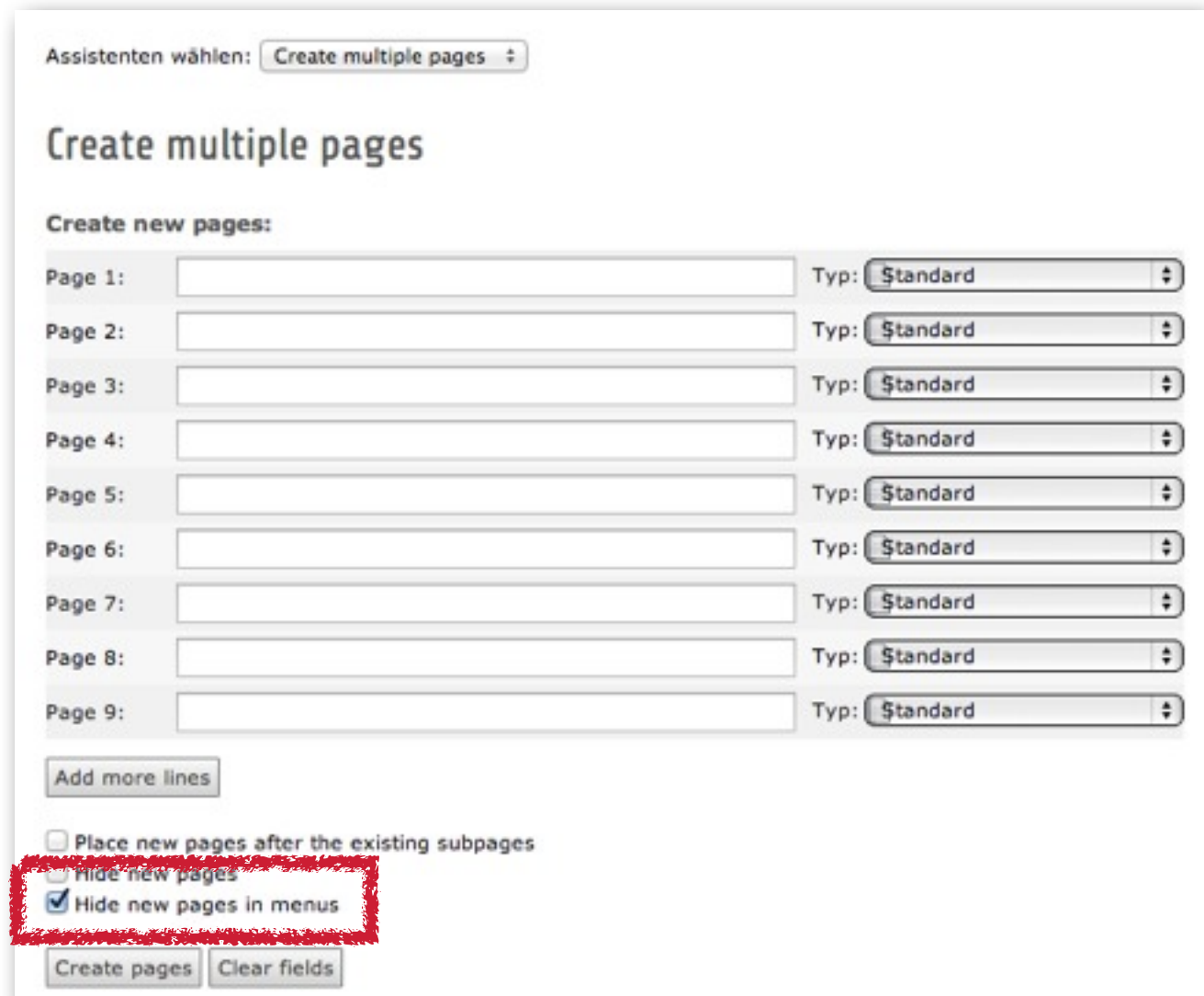
TYPO3 < 6.2



TYPO3 >= 6.2

„Hide in Menü“ im Funktionen-Menü

- Im Funktionen-Menü kann man im „Create multiple pages“ Wizard nun „Hide new pages in menus“ für alle Seiten anwählen



Assistenten wählen: Create multiple pages

Create multiple pages

Create new pages:

Page 1:		Typ: Standard
Page 2:		Typ: Standard
Page 3:		Typ: Standard
Page 4:		Typ: Standard
Page 5:		Typ: Standard
Page 6:		Typ: Standard
Page 7:		Typ: Standard
Page 8:		Typ: Standard
Page 9:		Typ: Standard

Add more lines

☐ Place new pages after the existing subpages

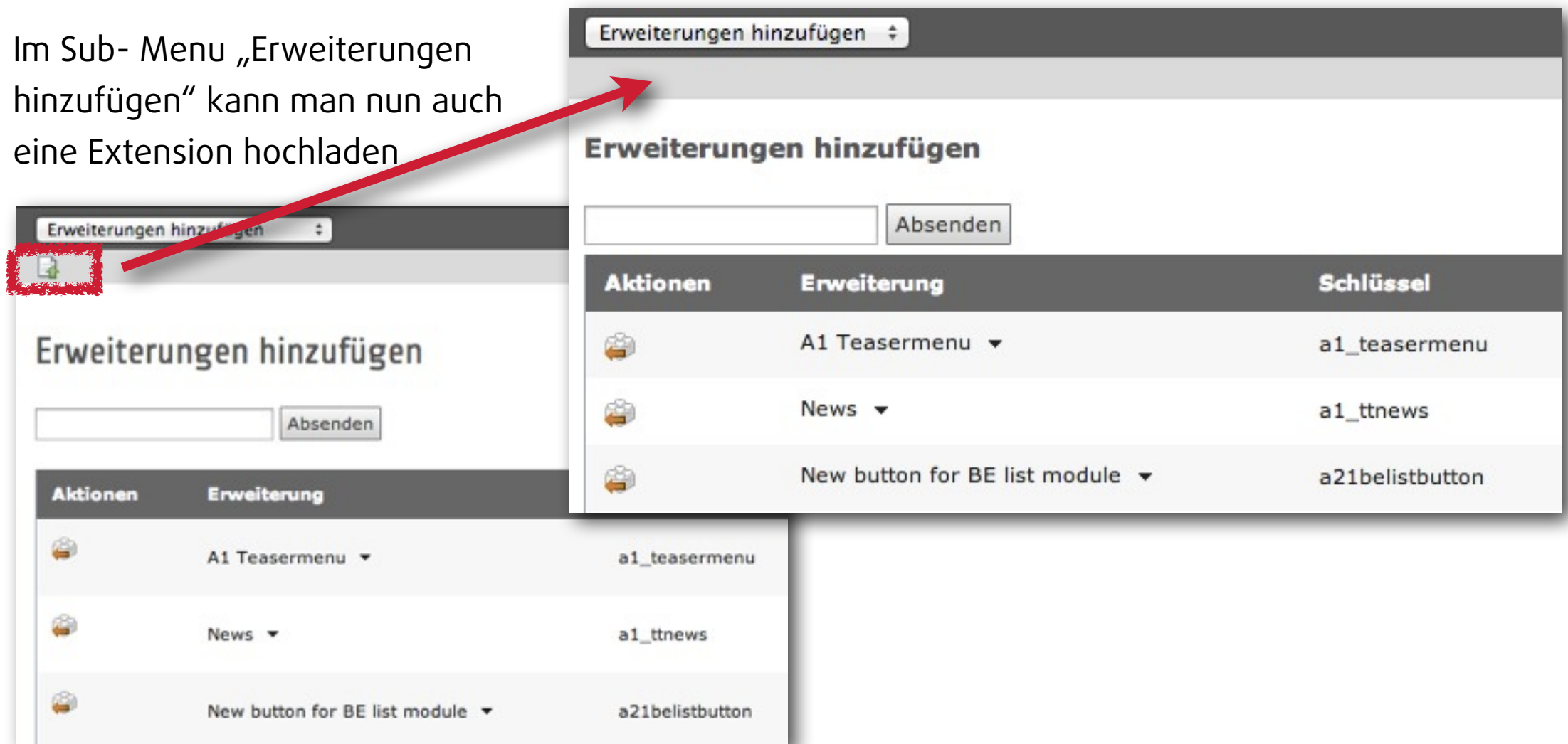
☐ Hide new pages

☒ Hide new pages in menus




Create pages Clear fields

Upload nun auch in „Erweiterungen hinzufügen“

- Im Sub- Menu „Erweiterungen hinzufügen“ kann man nun auch eine Extension hochladen



The screenshot shows the 'Erweiterungen hinzufügen' (Add Extensions) menu in the TYPO3 Backend. A red box highlights the upload icon (a green square with a white document and a plus sign) in the top left corner of the menu. A red arrow points from this icon to the right, towards the main content area of the menu. The main content area shows a search bar, an 'Absenden' (Submit) button, and a table of installed extensions.

Aktionen	Erweiterung	Schlüssel
	A1 Teasermenu ▾	a1_teasermenu
	News ▾	a1_ttnews
	New button for BE list module ▾	a21belistbutton

Kategorien nun auch manuell sortierbar

- Bislang waren Kategorien stets nach Titel sortiert, nun kann man diese auch manuell sortieren

▼ Kategorie (2) +

Titel						[Ref.]
TYPO3 Flow						-
TypoScript						-

TYPO3 < 6.2

▼ Kategorie (2) +

Titel						[Ref.]	[Lokalisierung]	Lokalisieren:
TypoScript							-	Standard
TYPO3 Flow							-	Standard

TYPO3 >= 6.2

Recycler nach letztem Zugriff sortierbar

- Man kann die Einträge im Recycler nun nach dem letzten Zugriff sortieren

TYPO3 < 6.2

Papierkorb

Der Papierkorb erlaubt es, gelöschte Daten auszuwählen und sie wiederherzustellen. Ist auch das übergeordnete Element gelöscht (etwa die Seite, auf der sich das wiederherzustellende Inhaltselement befunden hat), können Sie Datensätze auch rekursiv wiederherstellen.

Suche: Tiefe: Unendlich Tabelle: Alle Tabellen

	UID	PID	Datensätze	Tabelle
<input type="checkbox"/>	210	51	TYPO3 History	Seiteninhalt

TYPO3 >= 6.2

Papierkorb

Der Papierkorb erlaubt es, gelöschte Daten auszuwählen und sie wiederherzustellen. Ist auch das übergeordnete Element gelöscht (etwa die Seite, auf der sich das wiederherzustellende Inhaltselement befunden hat), können Sie Datensätze auch rekursiv wiederherstellen.

Suche: Tiefe: Diese Seite Tabelle: Alle Tabellen

	UID	PID	Datensätze	Tabelle	Letzte Bearbe...
<input type="checkbox"/>	1	1	1	Dateireferenz	22-08-13 10:52
<input type="checkbox"/>	2	1	2	Dateireferenz	19-08-13 14:08
<input type="checkbox"/>	3	1	3	Dateireferenz	22-08-13 10:52
<input type="checkbox"/>	4	1	4	Dateireferenz	22-08-13 10:52
<input type="checkbox"/>	1	1	NEW SITE	Template	22-08-13 14:15
<input type="checkbox"/>	2	1	[Kein Titel]	Seiteninhalt	22-08-13 10:52

Granulare Rechte für Dateien und Ordner

- Seit TYPO3 CMS 6.0 ist es möglich, die Rechte für Dateien und Ordner sehr granular per UserTSconfig zu setzen
- Dies ist nun auch im BE-User und in der BE-Gruppe als Formular möglich

Rechte für Dateioperationen:

☐ **Directory:**

- ☒ Directory: Read
- ☒ Directory: Write
- ☒ Directory: Add
- ☒ Directory: Rename
- ☒ Directory: Move
- ☐ Directory: Copy
- ☒ Directory: Delete
- ☐ Directory: Delete recursively

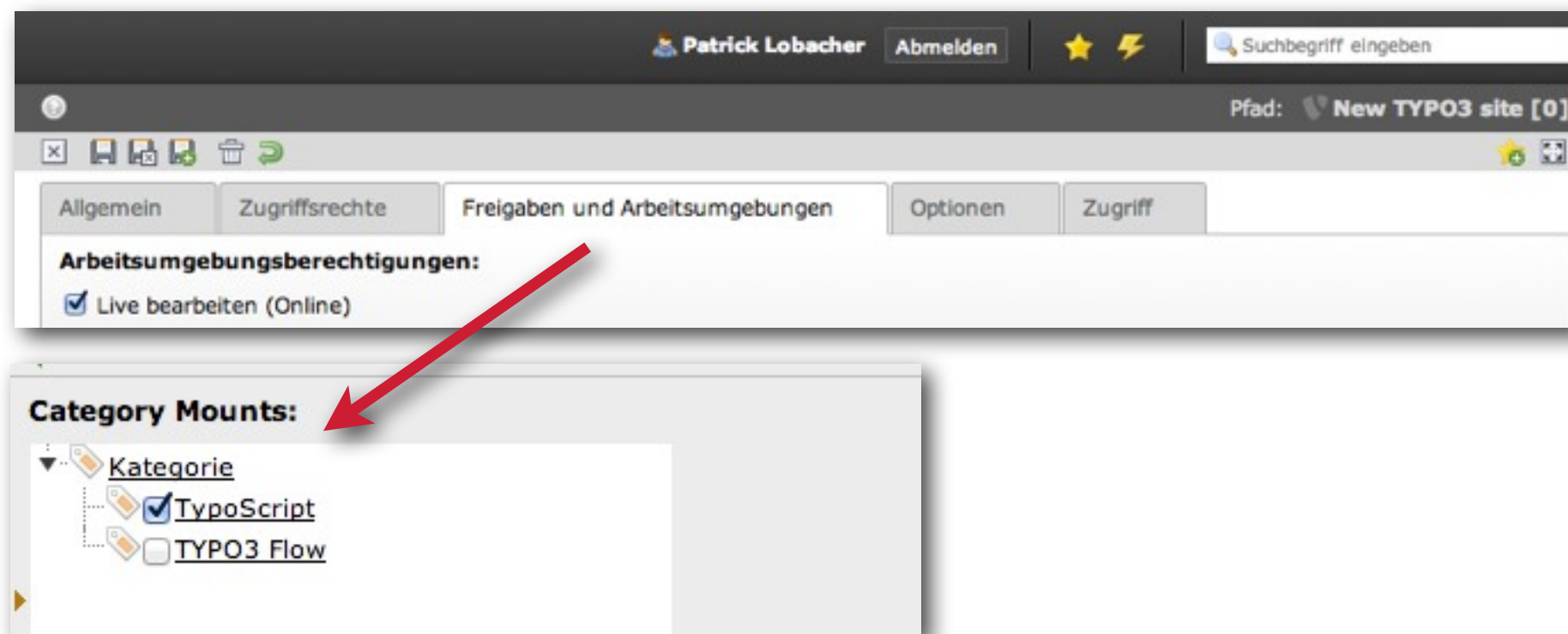
☐ **Files:**

- ☒ Files: Read
- ☒ Files: Write
- ☒ Files: Add

```
permissions.file.default {  
    addFile = 0  
    readFile = 1  
    writeFile = 0  
    copyFile = 1  
    moveFile = 1  
    renameFile = 1  
    unzipFile = 1  
    deleteFile = 1  
    addFolder = 1  
    readFolder = 1  
    writeFolder = 1  
    copyFolder = 1  
    moveFolder = 1  
    renameFolder = 1  
    deleteFolder = 1  
    recursiveDeleteFolder = 1  
}  
  
permissions.file.storage.1 {  
    addFile = 1  
    readFile = 1  
    writeFile = 1  
    copyFile = 1  
    moveFile = 1  
    renameFile = 1  
    unzipFile = 1  
    deleteFile = 1  
    addFolder = 1  
    readFolder = 1  
    writeFolder = 1  
    copyFolder = 1  
    moveFolder = 1  
    renameFolder = 1  
    deleteFolder = 1  
    recursiveDeleteFolder = 1  
}
```

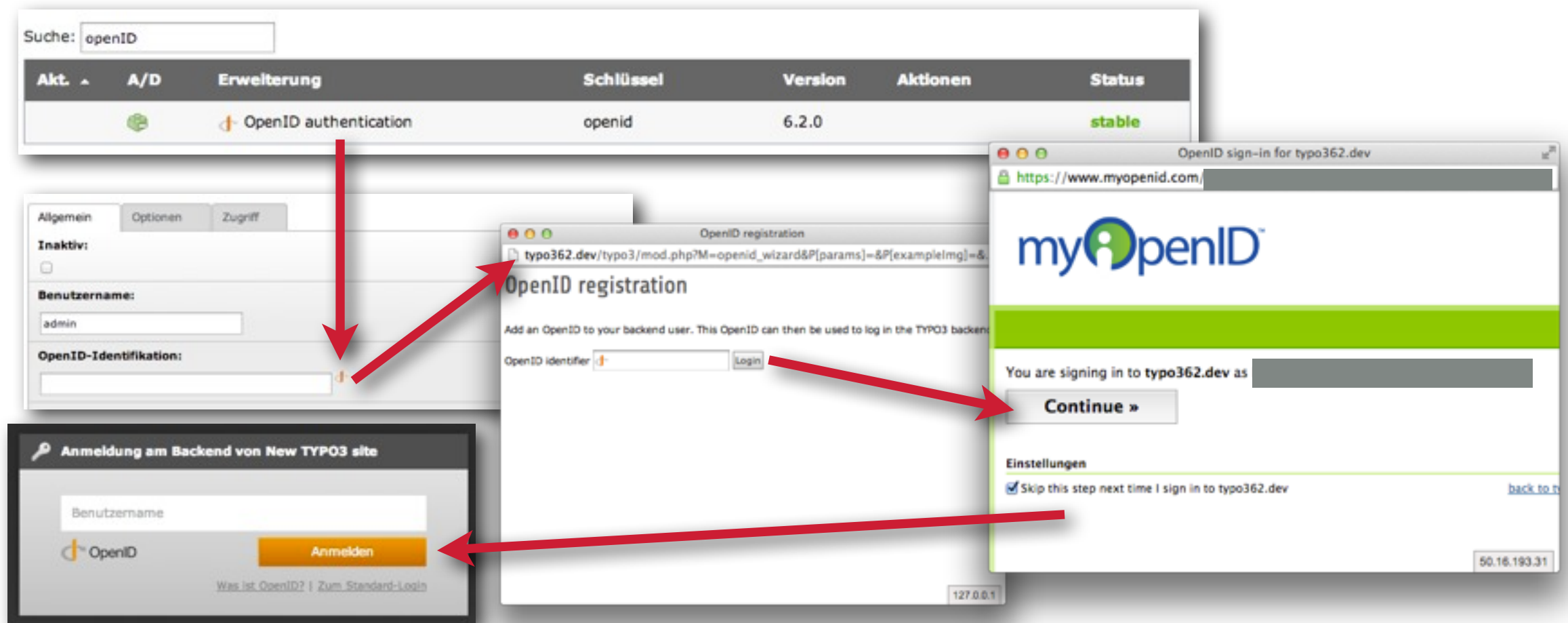
Einschränkung von Kategorien - BE-User/-Gruppen

- Bei Backend-Usern und Backend-Gruppen kann nun definiert werden, welche Kategorien diese zu sehen bekommen
- Dies geschieht im Tab „Freigaben und Arbeitsumgebungen“



OpenID-Wizard für Backend-Benutzer

- Bei der Verwaltung von Backend-Benutzern kann man die OpenID per Wizard angeben
- Hierfür muss die Systemextension „openid“ installiert sein



The image shows a sequence of steps for setting up OpenID authentication in the TYPO3 CMS 6.2 LTS backend:

- Backend Extension List:** A search for "openid" shows the "OpenID authentication" extension (key: openid, version: 6.2.0, status: stable).
- User Settings:** In the user management interface, the "OpenID-Identifikation" field is highlighted for the user "admin".
- OpenID registration:** A wizard window titled "OpenID registration" prompts the user to add an OpenID to their backend user. It includes a field for the "OpenID identifier" and a "Login" button.
- myOpenID sign-in:** An external browser window shows the "myOpenID" sign-in page for "typo362.dev", with a "Continue »" button.

Red arrows indicate the workflow: from the extension list to the user settings, then to the registration wizard, and finally to the external sign-in page.

MISC 1

- Das Feld `username` bekommt den Autofokus, sobald man das Backend-Login aufruft (über das HTML5-Attribut `autofocus="autofocus"`)

TSconfig & **TypoScript**

TS: INCLUDE_TYPOSCRIPT für alle Dateien eines Verzeichnisses

- Man kann nun alle Dateien eines Verzeichnisses per `INCLUDE_TYPOSCRIPT` inkludieren lassen
- Es werden zudem auch Unterverzeichnisse eingelesen
- Gibt man die Option „`extensions`“ an, so werden nur die Dateien mit dieser Extension inkludiert
- Die Reihenfolge ist: erst Dateien, dann Verzeichnisse - beide alphabetisch

```
<INCLUDE TYPOSCRIPT: source="DIR:directorypath" extensions="ts">
```

oder

```
<INCLUDE_TYPOSCRIPT: source="DIR:EXT:myext/Typoscript">
```

- Es können nur Dateien mit den Endungen (ts, t3, t3s, t3c, txt) inkludiert werden. Will man eine andere Extension verwenden, muss man diese in `$TYPO3_CONF_VARS['SYS']['tsfile_ext']` registrieren.

TS: INCLUDE_TYPOSCRIPT auch mit relativen Pfaden möglich

- Ab sofort kann man auch relative Pfade beim Include verwenden
- Der erste Include muss dabei absolut erfolgen
- Mit „.“ wird das aktuelle Verzeichnis des letzten Includes angesprochen
- Mit „..“ wird das Elternverzeichnis des letzten Includes angesprochen
- Beispiele

```
<INCLUDE_TYPOSCRIPT: source="FILE:../filename.ts">
```

oder

```
<INCLUDE_TYPOSCRIPT: source="FILE:./filename.ts">
```

TS: StdWrap für strPad

- Die Optionen **length**, **padWidth** und **type** der stdWrap-Funktion strPad wurden nun ebenfalls mit der Möglichkeit zum stdWrap versehen:

```
page = PAGE
page.10 = TEXT
page.10 {
    value = Hallo Welt!
    strPad {
        length = 5
        length {
            current = 1
            setCurrent.data = TSFE:page|uid
            setCurrent.wrap = | + 80
            prioriCalc = 1
        }
        padWidth = .
    }
}
```

TS: stdWrap für _DEFAULT_PI_VARS

- Die sogenannten _DEFAULT_PI_VARS dienen dazu, Default-Werte für piVars (zu einer Extension gehörige GET/POST-Variablen) zu setzen
- Bislang war es allerdings lediglich möglich, einen Schlüssel und einen dazugehörigen Wert zu setzen.
- Nun kann man auch stdWrap verwenden

- Beispiel vor TYPO3 CMS 6.2

```
plugin.tt_news._DEFAULT_PI_VARS {  
    year = 2013  
}
```

Beispiel ab TYPO3 CMS 6.2

```
plugin.tt_news._DEFAULT_PI_VARS {  
    year.stdWrap.data = date:Y  
}
```


TS: Condition userFunc mit CSV

- Die Condition userFunc erlaubte es bisher lediglich einen einzigen Parameter zu übergeben
- Ab sofort kann man eine kommaseparierte Liste übergeben

```
[userFunc = user_function(argument1, argument2, ...)]
```

// TypeScript Code

```
[userFunc = user_match(checkLocalIP, 192.168)]
```

// PHP Code

```
function user_match($command, $subnet) {  
    switch($command) {  
        case 'checkLocalIP':  
            if (strstr(getenv('REMOTE_ADDR'), $subnet)) {  
                return TRUE;  
            }  
            ...  
            return FALSE;  
        }  
    }  
}
```

TS: Debug Register und Page per TypoScript

- Der Inhalt der Arrays `$GLOBALS['TSFE']->register` und `$GLOBALS['TSFE']->page` kann nun per debug Option innerhalb vom Datentyp `getText` ausgegeben werden

- Beispiele:**

```
10 = LOAD_REGISTER  
10.variable = wert
```

```
20 = TEXT  
20.data = debug:register
```

```
30 = TEXT  
30.data = debug:page
```

SYS_LASTCHANGED	1376804898
variable	wert
uid	1
pid	0
t3ver_old	0
t3ver_id	0
t3ver_wsid	0
t3ver_label	
t3ver_state	0
t3ver_stage	0
t3ver_count	0
t3ver_tstamp	0
t3ver_move_id	0
t3_origuid	0
tstamp	1376804898
sorting	256
deleted	0

TS: Zugriff auf Register für Title&AltText in File Links

- Bei Filelinks konnte man früher nur auf die Beschreibung via Register zugreifen
- Dies wurde nun insofern geändert, dass nun auch der Titel-Text (`register:titleText`) und der Alt-Text (`register:altText`) per Register zugänglich sind
- **Beispiele:**

```
# filelinks
tt_content.uploads.20 {
    # link description instead of filename
    labelStdWrap.data = register:description
    # output alternative text
    itemRendering.20.data = register:titleText
}
```

TS: OptionSplit für stdWrap Funktion replacement

- Die Option `replace` in der `stdWrap` Funktion `replacement` wurde nun mit `optionSplit` ausgestattet

- Beispiele:**

```
10 = TEXT
10.value = There_are_a_cat,_a_dog_and_a_tiger_in_da_hood!_Yeah!
10.replacement.10 {
    search = _
    replace = 1 || 2 || 3
    useOptionSplitReplace = 1
}
```

Ergebnis: There1are2a3cat,3a3dog3and3a3tiger3in3da3hood!3Yeah!

```
20 = TEXT
20.value = There are a cat, a dog and a tiger in da hood! Yeah!
20.replacement.10 {
    search = #(a) (Cat|Dog|Tiger)#i
    replace = ${1} tiny ${2} || ${1} midsize ${2} || ${1} big ${2}
    useOptionSplitReplace = 1
    useRegExp = 1
}
```

Ergebnis: There are a tiny cat, a midsize dog and a big tiger in da hood! Yeah!

TS: Register für FILES cObject zugefügt

- Für das Content Objekt FILES wurden die Register **FILE_NUM_CURRENT** und **FILES_COUNT** zugefügt

Beispiel (Der Index in FILE_NUM_CURRENT startet bei 0!):

```
10 = FILES
10 {
    references {
        table = tt_news
        uid.field = uid
        fieldName = media
    }
    renderObj = COA
    renderObj {
        10 = TEXT
        10.value = Rendert die erste Datei zweimal
        10.if.isFalse.data = register:FILE_NUM_CURRENT
        20 = TEXT
        20.value = file {register:FILE_NUM_CURRENT} of {register:FILES_COUNT}
        20.insertData = 1
    }
}
```


TS: Neuer TypoScript Menü-Typ: categories

- Im TypoScript gibt es nun die Möglichkeit, ein Kategorien-Menü zu erstellen

```
page.20 = HMENU
page.20 {
    special = categories
    special {
        # Kommaseparierte Liste von Kategorien bzw. aus stdWrap
        value = 1
        # Feld aus „pages“ nach der sortiert werden soll (stdWrap)
        sorting = title
        # Sortierung: „asc“ oder „desc“ (stdWrap)
        order = desc
        1 = TMENU
        1.NO {
            allWrap = <li> | </li>
        }
    }
}
```

TS: splitChar für JS- und CSS-Includes

- Die folgenden TypeScript-Optionen wurden nun mit der Eigenschaft **splitChar** beim `allWrap` ausgestattet (`allWrap.splitChar`):

```
includeJSlibs  
includeJSFooterlibs  
includeJS  
includeJSFooter  
includeCSS
```

- Das Default-Zeichen für die Option `splitChar` ist das Pipe-Symbol

TS: Conditions für ApplicationContext

- Der ApplicationContext kann nun auch per Condition abgefragt werden
- Wildcards + und * sowie REGEX mit /PREG_PATTERN/ können angegeben werden

```
[applicationContext = Development/Debugging, Development/Profiling]
    # ist gültig für "Development/Debugging" oder "Development/Profiling"
[global]
```

```
[applicationContext = Production*]
    # Ist gültig für jeden ApplicationContext mit einem RootContext
    # "Production",
    # wie z.B. "Production/Live" oder "Production/Staging"
[global]
```

```
[applicationContext = /^Production\/Staging\/Server\d+$/]
    # Ist gültig für "Production/Staging/Server1",
    # "Production/Staging/Server2"
    # u.s.w.
[global]
```

TS: CONTENT Datensätze ohne Default Sprache

- Wenn man Datensätze über das TypoScript Objekt „CONTENT“ ermittelt (bzw. über `.select`), kann man über die Option `languageField` all jene Datensätze in der aktuellen Sprache ermitteln lassen
- Dabei holt `.select` alle Datensätze in der Default-Sprache ("0,-1") und erstellt einen Overlay mit der aktuellen Sprache
- Damit war es aber nicht möglich, Datensätze zu ermitteln, die keine Entsprechung in der Default-Sprache hatten (`l1on_parent = ""`)
- Dafür wurde nun eine entsprechende Option innerhalb von `.select` eingeführt, die es ermöglicht, auch Datensätze in das Ergebnis zu inkludieren, die keine Entsprechung in der Default-Sprache haben:

```
INHALT = CONTENT
INHALT {
    table = tt_content
    select.includeRecordsWithoutDefaultTranslation = 1
    ...
}
```

TS: begin, maxItems und direction für FILES cObject

- Das FILES Content Objekt verfügt nun über die (aus HMENU bekannten) Optionen
 - `begin` (Angabe des ersten Elements, welches zurückgegeben wird)
 - `maxItems` (Maximale Anzahl Elemente)
- Zusätzlich wurde die Option `direction` als Eigenschaft von `sorting` implementiert (einziger Wert `desc`)
- Beispiel (5 Bilder aus dem Media-Feld, beginnend vom ersten, werden zurückgegeben):

```
page.10 = FILES
page.10 {
    references {
        table = pages
        uid.data = page:uid
        fieldName = media
    }
    maxItems = 5
    begin = 0
    sorting.direction = desc
}
page.10.renderObj = TEXT
page.10.renderObj {
    data = file:current:size
    wrap = <p>File size:<strong>|</strong></p>
}
```


Developer IP in Condition IP

- In der Condition `[IP=...]` kann nun die im Install Tool eingetragene Developer IP verwendet werden
- Beispiel:

```
[IP = devIP]
```

```
page.20 = TEXT  
page.20.value = Hello Developer!
```

```
[end]
```

TSconfig: Doktypes vom Seitenbaum ausschließen

- Über die Einstellung `excludeDoktypes` in der UserTSconfig (beim Benutzer oder der Gruppe) ist es möglich, bestimmte Doktypes vom Rendern im Seitenbaum auszuschließen
- **Beispiele:**

Ausschließen von Seiten vom Typ „Ordner“ (Folder)

`options.pageTree.excludeDoktypes = 254`

Ausschließen von Seiten vom Typ „Ordner“ (Folder) und dem Seitentyp „Normal“

`options.pageTree.excludeDoktypes = 254,1`

TSconfig: Module im BE ausblenden

- Man kann nun einzelne Module im Backend ausblenden
- Der Zugriff darauf wird aber nicht eingeschränkt - hier muss man nach wie vor mit den ACL für BE-User bzw. BE-Group arbeiten
- **Beispiele:**

```
options.hideModules = file, help
```

```
options.hideModules.web := addToList(func,info)
```

```
options.hideModules.system = BelogLog
```

TSconfig: Preview Domain setzen (PageTS)

- Im PageTSconfig ist es nun möglich eine Preview-Domain zu setzen, die für die Anzeige verwendet wird
- Dies ist insbesondere für Multi-Domain-Installationen nützlich und sinnvoll
- **Beispiel:**

```
TCEMAIN.previewDomain = example.com
```

Conditions in Backend-Layouts

- Innerhalb von Backend-Layouts können nun auch Conditions verwendet werden, z.B.:

```
backend_layout {  
    colCount = 2  
    rowCount = 1  
    rows {  
        1 {  
            columns {  
                1 {  
                    name = Main  
                    colPos = 0  
                }  
                2 {  
                    name = right  
                    colPos = 1  
                }  
            }  
        }  
    }  
}  
  
[PIDupinRootline = 509]  
backend_layout.rows.1.columns.2 >  
[GLOBAL]
```

MISC

- Hat man mehrere Login-Formulare über EXT:felogin auf einer Seite, funktioniert die „Passwort vergessen“ Funktion nicht sauber. Daher kann dies nun gezielt ein- und ausgeschaltet werden über die TypoScript Option `showForgotPassword`
- Die Frontend-Ausgabe wird nun immer mit Content-length Header ausgeliefert. Dafür wird die TypoScript-Eigenschaft `config.enableContentLengthHeader` per Default auf TRUE (1) gesetzt.
- Die Resultlist der Extension indexed_search hat nun stdWrap Eigenschaften über die Option `plugin.tx_indexedsearch.resultlist_stdWrap`

Package **Management**

Integration der PackageManager API - 1

- Der Package Manager von TYPO3 Flow wurde für TYPO3 CMS portiert
- Das ermöglicht Vendor-Namespaces und vereinheitlicht die Paket-Formate
- Extensions sind damit nur eine spezielle Art von „Packages“
- **Folgende Schritte wurden dafür durchgeführt - Teil 1:**
 - Entfernung des Schlüssels `$TYPO3_CONF['EXT']['extListArray']` aus der Datei `typo3conf/LocalConfiguration.php` (es wurde aber ein Fallback-Array-Zugriff eingerichtet)
 - Der Inhalt der alten `LocalConfiguration.php` Datei (sofern bereits eine vorhanden war), wird umkopiert in `typo3conf/LocalConfiguration.beforePackageStatesMigration.php`
 - In der Datei `typo3conf/PackageStates.php` werden nun die Stati (active/inactive) und die Positionen der Extensions im Filesystem eingetragen
 - Ermittelt werden die Extensions in den Verzeichnissen:
 - `typo3/sysex`
 - `typo3/ext`
 - `typo3/contrib`
 - `typo3conf/ext`
 - Packages (rekursiv)

Integration der PackageManager API - 2

- **Folgende Schritte wurden dafür durchgeführt - Teil 2:**
 - Jede Extension erhält nun eine Datei `composer.json` im Hauptverzeichnis der Extension und eine Datei `Package.php` im Verzeichnis `Classes/`
 - In der Datei `Classes/Package.php` gibt eine Extension über eine Property `protected $protected = TRUE;` an, dass die Extension „required“ ist (Flag)
 - Fehlt die Datei `PackageStates.php` (oder das `extListArray`), so wird diese Datei neu erstellt, indem lediglich Extensions dort eingetragen werden, die das Flag auf `TRUE` gestellt haben
 - Der Autoloader bekommt ein eigenes Caching-Backend
- Weitere Details finden sich hier:

<http://wiki.typo3.org/Blueprints/PackageManager>

Integration der PackageManager API - 3

- Aufbau des Arrays in `typo3conf/PackageManager.php`

```
return array (
    'packages' =>
        array (
            'core' =>
                array (
                    'manifestPath' => '',
                    'composerName' => 'typo3/cms/core',
                    'state' => 'active',
                    'packagePath' => 'typo3/sysex/core/',
                    'classesPath' => 'Classes/',
                ),
            'workspaces' =>
                array (
                    'manifestPath' => '',
                    'composerName' => 'typo3/cms/workspaces',
                    'state' => 'inactive',
                    'packagePath' => 'typo3/sysex/workspaces/',
                    'classesPath' => 'Classes/',
                ),
            ...
        ),
    'version' => 4,
)
```

Integration der PackageManager API - 4

- Aufbau einer `composer.json` Datei

```
{  
    "name": "typo3/cms/cms",  
    "type": "typo3-cms-framework",  
    "description": "TYPO3 Core",  
    "homepage": "http://typo3.org",  
    "license": ["GPL-2.0+"],  
    "version": "6.2.0",  
  
    "require": {  
        "typo3/cms/core": "*"  
    },  
    "replace": {  
        "cms": "*"   
    }  
}
```

- Infos zum Schema: <http://getcomposer.org/doc/04-schema.md>

Integration der PackageManager API - 5

- Aufbau einer Classes/Package.php Datei (einer Extension, die „required“ ist)

```
<?php
namespace TYPO3\CMS\Cms;
use TYPO3\CMS\Core\Package\Package as BasePackage;

/**
 * This is the cms package
 */
class Package extends BasePackage {

    /**
     * @var boolean
     */
    protected $protected = TRUE;

}
```


Änderungen im **System**

Workspaces Änderungen

- Workspaces User können nun selbst entscheiden, wem sie Notifications schicken wollen
- Der Tab „All“ im Workspacemodul ist nun für alle User (nicht nur Admins) sichtbar



Inkludierung von Normalize.css

- **Normalize.css** sorgt dafür, dass Browser Elemente konsistenter unter Zuhilfenahme von modernen Webstandards dargestellt werden
- So wird bei **Normalize.css** kein kompletter Reset durchgeführt, sondern
 - sinnvolle Voreinstellungen behalten
 - HTML-Elemente in ihrem Aussehen angeglichen
 - Bugs behoben und Browser-Inkompatibilitäten ausgeglichen und
 - die Anwendbarkeit von CSS auf einige störrische Elemente verbessert
- Zunächst wurde nur die Datei „normalize.css“ hier zugefügt (wird aber noch nicht verwendet):
`typo3/sysexternal/backend/Resources/Public/Css/structure/normalize.css`
- Informationen zu „normalize.css“
<http://necolas.github.io/normalize.css/>

Neue TCA displayCond-Optionen BIT und !BIT

- Damit ist es möglich, in einer Display-Condition im TCA (displayCond) auch gegen Multi-Value-Feldern (bitweise) zu testen:

```
'content' => array(
    'label' => '...',
    'config' => array(
        'type' => 'check',
        'items' => array(
            array('Content A', ''),
            array('Content B', ''),
            array('Content C', ''),
        ),
    ),
),
```

Dafür kann man BIT und !BIT (negieren) verwenden

```
'content_a' => array(
    'label' => '...',
    'displayCond' => 'FIELD:content:BIT:1',
    'config' => array(
        'type' => 'text',
    ),
),
'content_b' => array(
    'label' => '...',
    'displayCond' => 'FIELD:content:!BIT:2',
    'config' => array(
        'type' => 'text',
    ),
),
```

Automatisches Update für Sprachen möglich

- Es gibt nun einen Extbase Command Controller zum automatischen Update von Sprachen für Extensions
- Die Registrierung ist dabei wie folgt:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['extbase']['commandControllers'][]  
= 'TYPO3\CMS\Lang\Command\UpdateCommandController';
```

- Ein möglicher Aufruf wäre:

```
./cli_dispatch.phpsh extbase update:update de,en,fr
```

- Durch die Angabe der locales (kommasepariert) kann die Aktualisierung auf lediglich diese Sprachen eingeschränkt werden - lässt man die Angabe weg, werden alle Sprachen aktualisiert, die im Modul „Sprache“ ausgewählt sind

ReST: reStructuredText Format für alle Manuals - 1

- Die Manuals aller Systemextensions sind nun nicht mehr im Open Office SXW Format, sondern im Format reStructured Text (ReST).
- ReST ist eine vereinfachte Auszeichnungssprache mit dem Ziel, in der reinen Textform besonders lesbar zu sein. Weiterhin soll reStructuredText leicht in andere Formate umwandelbar sein.
- Eine kurze Einführung zu ReST findet sich bei der Wikipedia:
`http://de.wikipedia.org/wiki/ReStructuredText`
- ReST Syntax:
`http://wiki.typo3.org/ReST_Syntax`
- Einrichten einer ReST-Umgebung zum Rendern
`http://wiki.typo3.org/Rendering_reST`

ReST: reStructuredText Format für alle Manuals - 2

- Folgende Manuals wurden als ReST hinterlegt:
 - typ03/sysexst/**css_styled_content**/Documentation/*
 - typ03/sysexst/**dbal**/Documentation/*
 - typ03/sysexst/**felogin**/Documentation/*
 - typ03/sysexst/**form**/Documentation/*
 - typ03/sysexst/**indexed_search**/Documentation/*
 - typ03/sysexst/**linkvalidator**/Documentation/*
 - typ03/sysexst/**openid**/Documentation/*
 - typ03/sysexst/**recycler**/Documentation/*
 - typ03/sysexst/**rsaauth**/Documentation/*
 - typ03/sysexst/**rtehtmlarea**/Documentation/*
 - typ03/sysexst/**saltedpasswords**/Documentation/*
 - typ03/systext/**scheduler**/Documentation/*
 - typ03/sysexst/**sys_action**/Documentation/*
 - typ03/sysexst/**taskcenter**/Documentation/*
 - typ03/sysexst/**workspaces**/Documentation/*

Eigene Übersetzungsserver für Extensions

- Mit XLIFF und dem kostenfreien Pootle Übersetzungsserver ist es ein leichtes einen eigenen Übersetzungsserver für Extensions aufzusetzen. Dafür wurde ein Signal/Slot implementiert

- **Datei EXT:myext/localconf.php:**

```
/** @var \TYPO3\CMS\Extbase\SignalSlot\Dispatcher $signalSlotDispatcher */
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance('TYPO3\\CMS\\Extbase\\SignalSlot\\Dispatcher');
$signalSlotDispatcher->connect(
    'TYPO3\\CMS\\Lang\\Service\\UpdateTranslationService', 'postProcessMirrorUrl',
    'Company\\Extension\\Slots\\CustomMirror', 'postProcessMirrorUrl'
);
```

- **Datei EXT:myext/Classes/Slots/CustomMirror.php:**

```
<?php
namespace Company\Extensions\Slots;
class CustomMirror {
    /** @var string */
    protected static $extKey = 'myext';
    public function postProcessMirrorUrl($extensionKey, &$mirrorUrl) {
        if ($extensionKey === self::$extKey) {
            $mirrorUrl = 'http://mycompany.tld/typo3-packages/';
        }
    }
}
```

Automatisierte t3d Importe

- Extensions können nun t3d (TYPO3 Data - also Daten für die Extension) Dateien bereitstellen, die automatisch inkludiert werden
- In einer t3d-Datei kann vieles enthalten sein: Struktur, Daten, Relationen, Dateien
- Dafür muss das t3d-File an der folgenden Stelle platziert werden

`<Extension-Verzeichnis>/Initialisation/data.t3d`

- Sobald die Extension das erste Mal installiert wird, wird der Inhalt der t3d-Datei eingelesen
- Bei einer erneuten Installation, wird aber die Datei **nicht** erneut eingelesen

Automatisierte Datei Importe

- Extensions können nun Dateien bereitstellen, die automatisch inkludiert werden
- Dafür müssen die Dateien an der folgenden Stelle platziert werden

<Extension-Verzeichnis>/Initialisation/Files/*
- Sobald die Extension das erste Mal installiert wird, werden die Dateien nach **fileadmin/**
<extensionname> kopiert
- Bei einer erneuten Installation, werden die Dateien aber **nicht** erneut eingelesen

Extensions als Repositories nutzen

- Manchmal hängen Extensions von Extensions ab, die verändert und angepasst wurden oder von Extensions, die nicht im TER veröffentlicht wurden
- Ab sofort kann man einer Extension erlauben, eine Abhängigkeiten mitzubringen, in dem man die abhängigen Extensions in das folgende Verzeichnis (ungepackt) legt

<Extension-Verzeichnis>/Initialisation/Extensions/

- Die dort enthaltenen Extensions werden dann in das Verzeichnis `typo3conf/ext` umkopiert, um die Abhängigkeiten aufzulösen

Install/Uninstall von Extensions über CLI

- Extensions können nun via Command Line Interface (CLI) installiert werden
- Dafür muss ein Backend Benutzer `_cli_lowlevel` angelegt werden
- Nun kann man Extensions installieren und deinstallieren
- Der Extension-Key wird über `[key]` angegeben

```
./typo3/cli_dispatch.phpsh extbase extension:install [key]
```

```
./typo3/cli_dispatch.phpsh extbase extension:uninstall [key]
```


TCA: Kaskadierendes Löschen von Datensätzen

- Im TCA gibt es nun eine Einstellung, die es erlaubt, das kaskadierende Löschen von Kind-Datensätzen ein- bzw. auszuschalten
- Dafür muss die Relation vom Typ „inline“ sein
- Der Default-Wert ist 1 - d.h. beim Löschen eines Datensatzes werden die Kind-Datensätze automatisch mitgelöscht.

```
...  
    'type' => 'inline',  
    'foreign_table' => ...,  
    'behaviour' => array(  
        'enableCascadingDelete' => 0,  
    )  
...
```

Mehrere Kategorien pro Tabelle

- Bislang konnte über die Funktion

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

nur ein Kategorie-Feld pro Tabelle definiert werden. Weitere Aufrufe haben die vorhergehenden überschrieben

- Ab sofort können beliebig viele solcher Felder existieren
- Über den vierten Parameter (`options`) kann mit dem Key `label` ein eigenes Label für das Feld eingebracht werden

Data Provider für Backend Layouts

- Backend Layouts konnten bislang ausschließlich in der Datenbank gespeichert werden
- Damit war es nicht möglich, dass Extensions eigene Backend Layouts mitbringen konnten
- Das ist nun geändert worden - es können sogenannte „Data Provider“ definiert werden
- Dies sind Klassen, die das Interface

TYPO3\CMS\Backend\View\BackendLayout\DataProviderInterface
implementieren müssen

- Registriert wird die Data Provider Klasse wie folgt:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['BackendLayoutDataProvider']  
[$_EXTKEY] = 'Classname';
```

- Zudem gibt es neue API-Befehle für das Backend Layout Handling, z.B:

- `'itemsProcFunc' => 'TYPO3\CMS\Backend\View\BackendLayoutView->addBackendLayoutItems`
- `getBackendLayoutView()->getSelectedCombinedIdentifier($id);`
- `getBackendLayoutView()->getSelectedBackendLayout();`

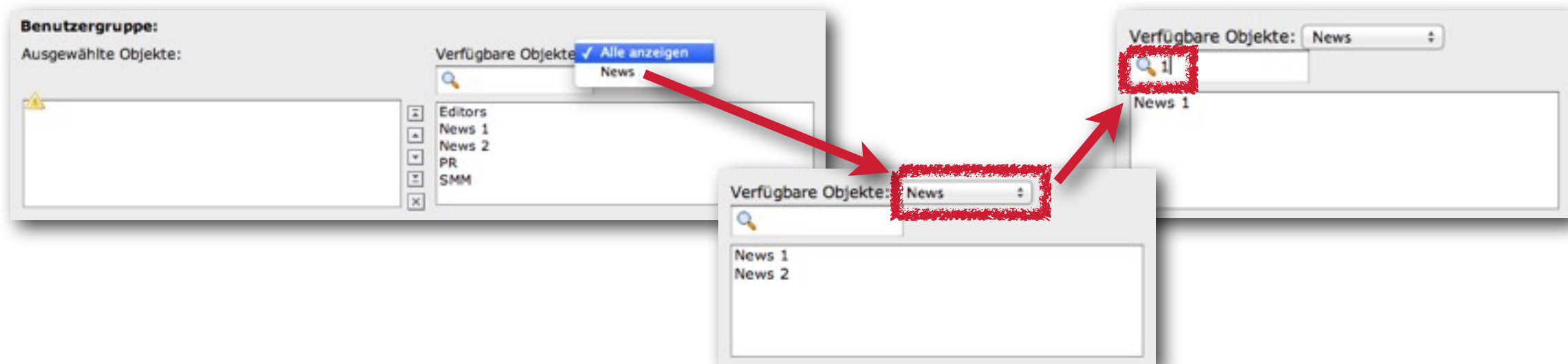
- Neue PageTSconfig Option, um Backend Layouts auszuschließen (von der Zuweisung):

```
options.backendLayout.exclude = default_1, my_extension__headerLayout
```

Filter für Multiselect-Felder

- Multiselect-Felder können schnell unübersichtlich werden
- Daher wurde hier ein Filter realisiert, der es ermöglicht, die Einträge per Vorauswahl und/oder per Suchbegriff einzuschränken
- Dafür muss das TCA entsprechend angepasst werden (z.B. in `typo3conf/extTables.php`):

```
$GLOBALS['TCA']['fe_users']['columns']['usergroup']['config']['enableMultiSelectFilterTextField'] = TRUE;  
$GLOBALS['TCA']['fe_users']['columns']['usergroup']['config']['multiSelectFilterItems'] = array(  
    array('', 'Alle anzeigen'),          // Erster Wert: Filter, Zweiter Wert: Label  
    array('News', 'News'),              // LLL: kann ebenfalls verwendet werden  
);
```



Add Link in IRRE kann nun geändert werden

- Über die neue Option `newRecordLinkAddTitle` kann der Add-Link-Titel in IRRE-Felder angepasst werden

```
$TCA["tx_irretutorial_inff_hotel"] = Array (  
    "columns" => Array (  
        "offers" => Array (  
            "config" => Array (  
                "type" => "inline",  
                ...  
                "appearance" => Array (  
                    "newRecordLinkAddTitle" => 1,  
                    "newRecordLinkTitle" => "Neuen Post anlegen!"  
                    "newRecordLinkPosition" => "both",  
                ),  
            ),  
        ),  
    ),  
)
```



ohne Option

mit Option `newRecordLinkTitle`

MISC 1

- **Eigene Record-List:** Es ist nun möglich eine eigene Record-List im Element-Browser zu verwenden, wenn diese eine Instanz der Klasse `\TYPO3\CMS\Backend\RecordList\ElementBrowserRecordList` ist (über die Eigenschaft `$recordList` in der Klasse `\TYPO3\CMS\Recordlist\Browser\ElementBrowser`).
- **Mehr Untergruppen möglich:** Bislang war das Feld zum Speichern von Untergruppen (`subgroup`) eine BE-Usergruppe nur 255 Zeichen groß - `varchar(255)`. Ab sofort ist der Typ auf `text` umgestellt.
- **Modul Web > Template nur noch eine Extension:** Die Extension „tstemplate“ wurde bislang technisch durch die folgenden zusätzlichen Extensions realisiert: „tstemplate_ceditor“, „tstemplate_info“, „tstemplate_objbrowser“, „tstemplate_analyzer“. Diese wurden nun in die Extension „testemplate“ integriert.
- **Support für die TCA-Option `label_userFunc_options` zur `BackendUtility` Klasse** zugefügt
- **Download von Extensions:** Sobald man im Erweiterungsmanager eine Extension herunterlädt, wird der Timestamp an den Dateinamen hinzugefügt: `<extensionKey>_<version>_<timestamp>.zip`

MISC 2

- **Logger ist PSR-3 kompatibel:** Der Logger der Logging API ist nun kompatibel mit dem PSR-3 Standard der PHP Framework Interop Group: <http://www.php-fig.org/psr/3/>
- Extension **saltedpassword** ist per Default required und aktiviert für das Backend
- Man kann nun die Argumente, die an einen SignalSlot-Dispatcher übergeben werden, verändern und diese werden nun auch vom Dispatcher (verändert) wieder zurückgegeben
- Der Workspace-Preview enthält nun alle GET-Parameter (und damit insbesondere die von Extensions)
- Der **Sprite-Generator** (`TYPO3\CMS\Backend\Sprite\SpriteGenerator`) unterstützt nun Bilder mit **doppelter Auflösung** und erzeugt somit zwei Dateien. Die Datei mit der höheren Auflösung erhält den Suffix `@x2.png`. Über CSS3 wird nun dafür gesorgt, dass Displays mit hoher Auflösung direkt das höher aufgelöste Icon laden.
- Unterstützung für **Proxy NTLM Authentifizierung** (Aktivierung über Install Tool):
`$GLOBALS['TYPO3_CONF_VARS']['SYS']['curlProxyNTLM']`
- Die Install-Tool Option `cookieHttpOnly` wurde per Default aktiviert - diese sorgt dafür, dass das SessionCookie nicht per JavaScript zugänglich ist

Application Programming Interface (API)

Hook für `tsfe::checkEnableFields`

- Bislang war es nicht möglich, das Feature „Auf Unterseiten ausdehnen“ (extend to subpages) in eigenen Extensions zu verwenden, um zusätzliche Regeln für die Sichtbarkeit der Seiten bereitzustellen. Dies lag daran, dass die Liste der zu überprüfenden Felder hartkodiert war (auf `hidden`, `endtime`, `starttime`)
- Mit diesem Hook kann man die Liste beliebig erweitern und damit andere Felder prüfen
- Klasse: `\TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController`

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['tslib/class.tslib_fe.php']  
['hook_checkEnableFields']
```

Hook für checkFlexFormValue im Data-Handler

- Mit diesem Hook kann man vor dem Speichern von FlexForm-Daten die Daten verändern.
- Beim Update der Daten einer Flexform wird normalerweise nicht geprüft, ob der Wert selbst bereits aus der Datenbank gelöscht wurde.
- Dies ist insbesondere in Extbase der Fall, wenn man SwitchableControllerActions verwenden möchte. Alte Actions, die es nicht mehr gibt, müssen so manuell aus der Datenbank entfernt werden

- Klasse: `\TYPO3\CMS\Core\DataHandling\DataHandler`

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_tcemain.php']  
[ 'checkFlexFormValue' ]
```

- Methode: `checkFlexFormValue_beforeMerge()`

Hook für den Header im PageLayoutController

- Mit diesem Hook kann man den Header im Modul „Web > Page“ erweitern
- Mögliche Verwendung: Anzeige aller Medien als Template vor dem Seiteninhalt, die in den Seiteneigenschaften hinterlegt wurden

- Klasse: `\TYPO3\CMS\Backend\Controller\PageLayoutController`

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['cms/layout/db_layout.php']  
[ 'drawHeaderHook' ];
```

- Methode: `callUserFunction`

IRRE: Defaults für erzeugte Werte

- Für IRRE_Datensätze gibt es eine neue TCA-Option für „inline“ Felder
- Über „foreign_record_defaults“ kann man in erzeugten Datensätzen Default-Werte setzen

```
config => array(  
    'type' => 'inline',  
    'foreign_table' => 'tt_content',  
    'foreign_record_defaults' => array(  
        'CType' => 'image'  
    ),  
)
```

Beim Anlegen eines Datensatzes ist dieser per Default vom Typ „image“ (kann aber natürlich vom Redakteur vor dem Speichern geändert werden).

- Folgende Felder können **nicht** mit Default-Werten versehen werden: `uid`, `pid` und `t3ver_*`, sowie die TCA-ctrl-Felder `crdate`, `cruser_id`, `delete`, `origUid`, `transOrigDiffSourceField`, `transOrigPointerField` und `tstamp`

API um Workspaces zu erweitern (1)

- Das Workspace Modul kann nun über eine API erweitert werden (um PHP und JavaScript-Komponenten überschreiben zu können):

ext_localconf.php:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_tcemain.php']  
['processCmdmapClass']['workspaces_logger'] = '[Vendor]\\WorkspacesLogger\\  
\\Hook\\DataHandlerHook';
```

ext_tables.php:

```
\TYPO3\CMS\Workspaces\Service\AdditionalColumnService::getInstance() -  
>register(  
    'WorkspacesLogger_StageChange',  
    '[Vendor]\\WorkspacesLogger\\DataProvider'  
);
```

```
\TYPO3\CMS\Workspaces\Service\AdditionalResourceService::getInstance() -  
>addJavaScriptResource(  
    'WorkspacesLogger',  
    'EXT:[extention_key]/Resources/Public/JavaScript/StageChange.js'  
);
```

API um Workspaces zu erweitern (2)

- Datei: [Vendor]\WorkspacesLogger\Hook\DataHandlerHook
(DEMO: https://github.com/ohader/workspaces_logger)

```
<?php
namespace [Vendor]\WorkspacesLogger\Hook;
use TYPO3\CMS\Core\SingletonInterface;

class DataHandlerHook implements SingletonInterface {
    const TABLE_Name = 'tx_workspaceslogger_event';
    const EVENT_SetStage = 91;
    /**
     * hook that is called when no prepared command was found
     * ...
     * public function processCmdmap($command, $table, $id, $value, &
    $commandIsProcessed, \TYPO3\CMS\Core\DataHandling\DataHandler $tcemainObj) {
        ...
        $action = (string) $value['action'];
        if ($command === 'version' && $action === 'setStage' &&
        $commandIsProcessed) {
            ...
        }
    }
}
```

MISC 1

- Analog zu `\TYPO3\CMS\Core\Utility\MathUtility::canBeInterpretedAsInteger` gibt es nun `\TYPO3\CMS\Core\Utility\MathUtility::canBeInterpretedAsFloat`
- Es gibt nun einen eigenen Enumeration-Type: `\TYPO3\CMS\Core\Type\Enumeration` (verwendet beispielsweise in `\TYPO3\CMS\Core\Versioning\VersionState` und dann im Code z.B. mittels `new VersionState(VersionState::DEFAULT_STATE)`)

Extbase & **Fluid**

Extbase: ObjectManager->getScope() zugefügt

- Wie in TYPO3 Flow kann nun über die Methode getScope() des Objekt Managers überprüft werden, ob eine Klasse vom Typ Singleton oder Prototype ist:

```
/**
 * @var \TYPO3\CMS\Extbase\Object\ObjectManagerInterface
 * @inject
 */
protected $objectManager;

$this->objectManager->getScope($propertyTargetClassName) === \TYPO3\CMS
\Extbase\Object\Container\Container::SCOPE_PROTOTYPE

$this->objectManager->getScope($propertyTargetClassName) === \TYPO3\CMS
\Extbase\Object\Container\Container::SCOPE_SINGLETON
```

Extbase: Ermittlung des PageTypes in URLs

- Wenn man einen Link setzt, hinter dem das Format geändert werden soll, muss gleichzeitig auch der Parameter `pageType` gesetzt werden:

```
<f:link.action arguments="{blog: blog}"  
pageType="{settings.plaintextPageType}" format="txt">[plaintext]</  
f:link.action></li>
```

- Über eine neue TypeScript Option `formatToPageTypeMapping`, kann nun ein Mapping hinterlegt werden, welches automatisch zur Erkennung des PageTypes beim Wechsel des Formats sorgt:

```
plugin.tx_myext {  
    view.formatToPageTypeMapping {  
        txt = 99  
        pdf = 123  
    }  
}
```

Nun reicht:

```
<f:link.action arguments="{blog: blog}" format="txt">[plaintext]</  
f:link.action></li>
```

Extbase: Object Type Converter (aus Flow) - 1 von 2

- Der in TYPO3 Flow eingeführte „Object Type Converter“ wurde nach Extbase portiert
- Damit ist es möglich, Arrays in nicht-persistente Objekte zu konvertieren
- Beispiel: Erstellen eines nicht-persistenten Objekts `$demand` aus einem GET-Request heraus:

- **Der GET-Request:**

```
http://master.dev/index.php?
id=299&tx_mapperexample_piexample[action]=list&tx_mapperexample_piexample[controller]
=Entity&tx_mapperexample_piexample[demand]
[title]=foo&tx_mapperexample_piexample[demand][relation]=1
```

- **Die initializeListAction() des Entity-Controllers:**

```
use [Vendor]\[ExtKey]\Domain\Dto\Demand;
public function initializeListAction() {
    /** @var PropertyMappingConfiguration $demandConfiguration */
    $demandConfiguration = $this->arguments['demand']->
    >getPropertyMappingConfiguration();
    $demandConfiguration->allowAllProperties()->forProperty('relation')
    ->allowAllProperties()->setTypeConverterOption(
    'TYPO3\\CMS\\Extbase\\Property\\TypeConverter\\PersistentObjectConverter',
    PersistentObjectConverter::CONFIGURATION_CREATION_ALLOWED,
    TRUE);
}
```


Extbase: Object Type Converter (aus Flow) - 2 von 2

- **Die listAction() des Entity-Controllers:**

```
use [Vendor]\[ExtKey]\Domain\Dto\Demand;
/**
 * @param Demand $demand
 */
public function listAction(Demand $demand = NULL) {
    $entities = $this->entityRepository->findAll();
    $this->view->assign('entities', $entities);
}
```

- **Die Model-Datei [Vendor]\[ExtKey]\Domain\Dto\Demand.php**

```
<?php
namespace [Vendor]\[ExtKey]\Domain\Dto;
use [Vendor]\[ExtKey]\Domain\Model\Relation;
class Demand {
    protected $relation;
    /**
     * @param \TYPO3Friends\MapperExample\Domain\Model\Relation $relation
     */
    public function setRelation($relation) {
        $this->relation = $relation;
    }
}
```

Extbase: Chaining von QuerySettings

- Das Chaining (Verketten) von QuerySettings ist nun auch mit den „neuen“ Optionen `setIncludeDeleted` und `setIgnoreEnableFields` (die mit TYPO3 CMS 6.0 eingeführt wurden) möglich:

```
$query->getQuerySettings()  
->setRespectStoragePage(FALSE)  
->setRespectSysLanguage(FALSE)  
->setIgnoreEnableFields(TRUE)  
->setIncludeDeleted(TRUE);
```

Extbase: RawQueryResults per Query

- Bisher war es möglich, mittels

```
$query->getQuerySettings()->setReturnRawQueryResult(TRUE)
```

dafür zu sorgen, dass keine Objekte vom QueryManager rekonstruiert werden, sondern das Query-Ergebnis „roh“ zurückgegeben wird.

- Ab sofort gibt es diese Möglichkeit nicht mehr zentral, sondern man gibt dies per Query im execute-Statement mittels TRUE an:

```
$query->execute(TRUE);
```

Extbase: Rekursive Validierung und leere Werte

- Extbase verwendet nun die von Flow portierte sogenannte „Rekursive Validierung“
- Dies bedeutet, dass bei Erzeugung von verschachtelten Objekten (Objekt-Baum) durch den Property-Mapper auch die inneren Objekte validiert werden und nicht nur das äußere Objekt wie bisher
- Zudem ist es nun möglich, leere (optionale) Werte zu zulassen. Will man daher explizit erreichen, dass eine Eigenschaft angegeben werden muss, so muss der `NotEmptyValidator` verwendet werden.

Extbase: Application Context

- Innerhalb von Extbase kann man nun den von TYPO3 Flow bekannten ApplicationContext verwenden
- Der Kontext wird über die Umgebungs-Variable `TYP03_CONTEXT` gesetzt. Als Root-Kontext gibt es: „Development“, „Testing“ und „Production“. Anschließend kann es einen Sub-Kontext geben - z.B.: „Production/Staging“
- Ist kein Kontext gesetzt, wird dieser per Default auf „Production“ gesetzt
- Gesetzt werden kann der Kontext beispielsweise wie folgt (z.B. in der Datei `.htaccess`):

```
# Rules to set ApplicationContext based on hostname
RewriteCond %{HTTP_HOST} ^dev\.example\.com$
RewriteRule (.*?) $1 [E=TYP03_CONTEXT:Development]
RewriteCond %{HTTP_HOST} ^staging\.example\.com$
RewriteRule (.*?) $1 [E=TYP03_CONTEXT:Production/Staging]
RewriteCond %{HTTP_HOST} ^www\.example\.com$
RewriteRule (.*?) $1 [E=TYP03_CONTEXT:Production]
```

- Abgefragt werden kann der Kontext wie folgt:

```
\TYPO3\CMS\Core\Core\Bootstrap::getInstance()->getContext();
\TYPO3\CMS\Core\Utility\GeneralUtility::getContext();
```

Fluid: Image-ViewHelper nun mit optionalem Title-Attribut

- Im Image-ViewHelper kann nun das Attribut `title` weggelassen werden und dieses wird dann auch nicht mit ausgegeben. Das Verhalten vor TYPO3 CMS 6.2 war so, dass in diesem Fall ein Attribut `title` gerendert und dafür der Inhalt des Attributs `alt` verwendet wurde:

```
<f:image src="background.jpg" alt="Text" />
```

führt zu

```

```

(und nicht wie vor TYPO3 CMS 6.2 zu)

```

```

Fluid: Placeholder-Attribut für Textfield und Textarea

- Die beiden ViewHelper `form.textfield` und `form.textarea` können nun mit einem Attribut `placeholder` ausgestattet werden:

```
<f:form.textfield
    id="powermail_field_{field.marker}"
    placeholder="{field.title -> vh:string.RawAndRemoveXss()}"
    name="field[{field.uid}]"
    value="{vh:Misc.PrefillField(field: '{field}')}"
    class="powermail_field powermail_input {vh:Misc.ValidationClass(field:
'{field}') {vh:Misc.ErrorClass(field: '{field}', class:
'powermail_field_error')}}"
    required="{field.mandatory}" />
```


Fluid: switch ViewHelper zugefügt

- Es gibt nun einen `switch`-ViewHelper der es ermöglicht, aufgrund eines Ausdrucks (Attribut `expression`) einen der Fälle (die mit dem `case`-ViewHelper realisiert werden) auszuwählen:

```
<f:switch expression="{person.gender}">
  <f:case value="male">Mr.</f:case>
  <f:case value="female">Mrs.</f:case>
</f:switch>
```

- **Achtung:** Die exzessive Verwendung dieses ViewHelpers könnte auf eine schlechte Architektur hindeuten. So wäre in obigen Beispiel auch folgendes möglich:

```
<f:render partial="title.{person.gender}" />
```

mit den entsprechenden Partialen `title.male.html` und `title.female.html`.

Benötigt man sehr viele `case`-ViewHelper, sollte man diese besser durch Partialen ersetzen

Fluid: format.bytes ViewHelper zugefügt

- Dieser ViewHelper verwandelt eine Byte-Angabe (in Form eines Integers) in eine lesbare Version

Beispiel (fileSize = 1263616)

```
{fileSize -> f:format.bytes() }
```

Ausgabe = 1234 KB

```
{fileSize -> f:format.bytes(decimals: 2, decimalSeparator: '.',  
thousandsSeparator: ',') }
```

Ausgabe: 1,234.00 KB

Fluid: Backend Container ViewHelper Überarbeitung 1

- **Der ViewHelper `be.container` wurde überarbeitet (Neue Optionen in Rot, Deprecated in Bold)**

```
@param string $pageTitle title tag of the module.  
@param boolean $enableJumpToUrl If TRUE, includes "jumpTpUrl" javascript function required  
by ActionMenu. Defaults to TRUE  
@param boolean $enableClickMenu If TRUE, loads clickmenu.js required by BE context menus.  
Defaults to TRUE  
@param boolean $loadPrototype specifies whether to load prototype library. Defaults to TRUE  
@param boolean $loadScriptaculous specifies whether to load scriptaculous libraries.  
Defaults to FALSE  
@param string $scriptaculousModule additional modules for scriptaculous  
@param boolean $loadExtJs specifies whether to load ExtJS library. Defaults to FALSE  
@param boolean $loadExtJsTheme whether to load ExtJS "grey" theme. Defaults to FALSE  
@param string $extJsAdapter load alternative adapter (ext-base is default adapter)  
@param boolean $enableExtJsDebug if TRUE, debug version of ExtJS is loaded. Use this for  
development only  
@param string $addCssFile Custom CSS file to be loaded (deprecated, use $includeCssFiles)  
@param string $addJsFile Custom JavaScript file to be loaded (deprecated, use  
$includeJsFiles)  
@param boolean $loadjQuery whether to load jQuery library. Defaults to FALSE  
@param array $includeCssFiles List of custom CSS file to be loaded  
@param array $includeJsFiles List of custom JavaScript file to be loaded  
@param array $addJsInlineLabels Custom labels to add to JavaScript inline labels
```

Fluid: Backend Container ViewHelper Überarbeitung 2

- Der ViewHelper `be.container` wurde überarbeitet

Beispiel:

```
<f:be.container pageTitle="foo" enableJumpToUrl="false"
enableClickMenu="false" loadPrototype="false" loadScriptaculous="false"
scriptaculousModule="someModule,someOtherModule" loadExtJs="true"
loadExtJsTheme="false" extJsAdapter="jQuery" enableExtJsDebug="true"
loadjQuery="true" includeCssFiles="0: '{f:uri.resource(path:\'Styles/
Styles.css\')}'" includeJsFiles="0: '{f:uri.resource(path:\'JavaScript/
Library1.js\')}', 1: '{f:uri.resource(path:\'JavaScript/Library2.js\')}'"
addJsInlineLabels="{0: 'label1', 1: 'label2'}">your module content</
f:be.container>
```

Fluid: Backend ViewHelper Button.Icon finalisiert

- Der Button.Icon ViewHelper, ist nun nicht mehr „experimental“
- Zweck des ViewHelpers ist es, ein Button-Icon zurückzuliefern (und ggf. zu verlinken)

```
<f:be.buttons.icon uri="{f:uri.action(action: 'new')}" icon="actions-document-new" title="Create new Foo" />
```

```
<f:be.buttons.icon icon="actions-document-new" title="Create new Foo" />
```

- Für das Attribut „icon“ können zahlreiche (über 310) Werte verwendet werden, die in der folgenden Datei unter dem Schlüssel `$GLOBALS['TBE_STYLES']['spriteIconApi']['coreSpriteImageNames']` aufgelistet sind:

```
typo3/sysextext/core/ext_tables.php
```

Fluid: addQueryStringMethod Support

- Die Option `addQueryString` arbeitet bisher nur mit GET-Parametern, indem diese an den erzeugten Link wieder angehängt werden
- Damit ist es aber nicht möglich POST Parameter (wie sie z.B. bei Widgets vorkommen) anzuhängen
- Über die Option **`addQueryStringMethod`** kann nun angegeben werden, welche Parameter verwendet werden sollen: `GET` oder `POST` oder `GET, POST` oder `POST, GET` - default ist `GET`
- Der UriBuilder verfügt nun ebenfalls über diese Option
- Folgende ViewHelper wurden mit dieser Option ausgestattet
 - `link.action`
 - `link.page`
 - `uri.action`
 - `uri.page`
 - `widget.link`
 - `widget.uri`
 - `widget.pagination`

Fluid: Fluid Template Fallback Support

- Wenn man in Extbase Templates erweitern will, musste man bislang alle Templates kopieren
- Ab sofort kann man „Fallback-Pfade“ definieren. Fluid schaut in dem Verzeichnis mit dem höchsten Index und geht (wenn das Template nicht gefunden wurde) runter bis zum kleinsten Index
- Fallbacks gibt es für Templates, Partials und Layouts:

templateRootPaths, partialRootPaths, layoutRootPaths

```
plugin.tx_simpleblog {
    view {
        templateRootPath = EXT:simpleblog/Resources/Private/Templates/
    }
}

plugin.tx_simpleblog {
    view {
        templateRootPath >
        templateRootPaths {
            10 = fileadmin/simpleblog/templates
            20 = fileadmin/special/simpleblog/templates
        }
    }
}
```

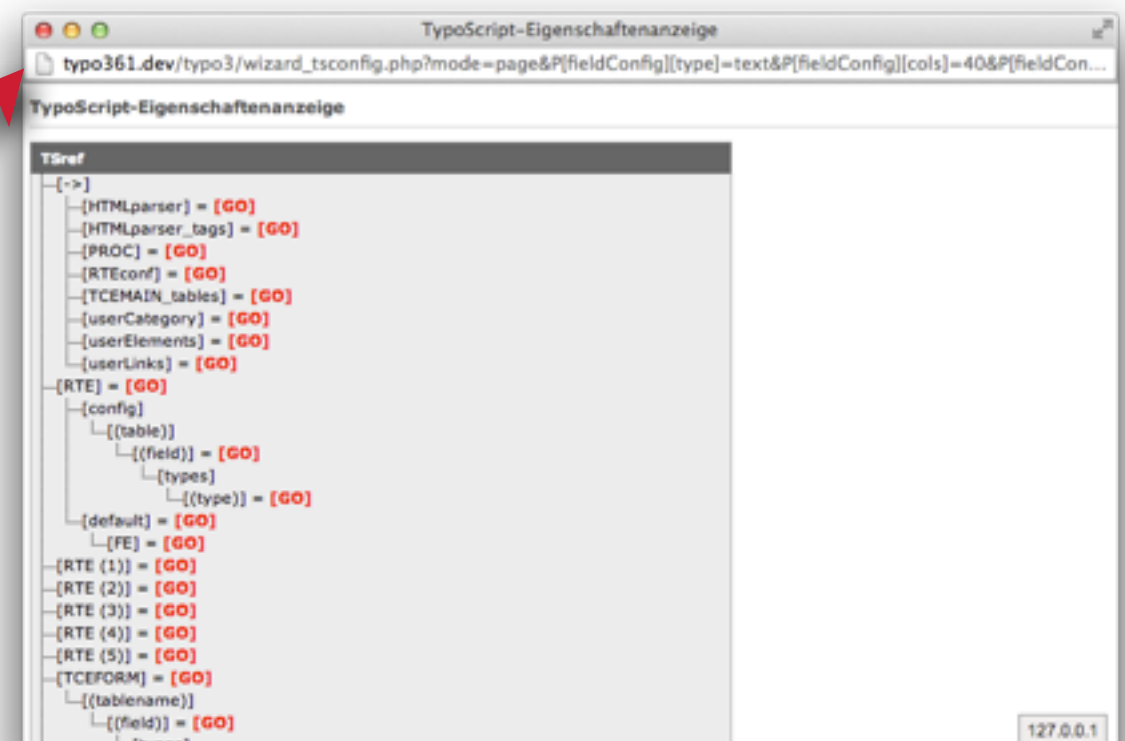
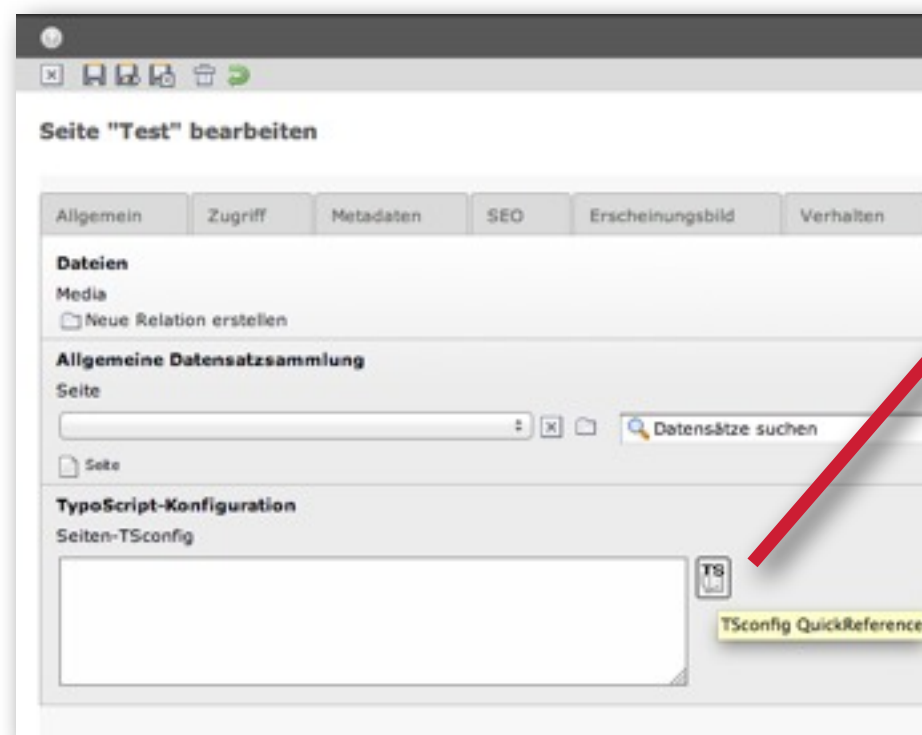

Fluid: MISC

- **autofocus:** Die ViewHelper `form.textfield` und `form.textarea` besitzen nun das optionale HTML5-Attribut `autofocus`

Entfernte & **veraltete Funktionen**

Extension tsconfig_help wurde entfernt

- Die Extension „TSConfig / TypoScript Object Reference“ (tsconfig_help) wurde komplett entfernt
- Einerseits waren die dort vorhandenen Informationen sehr veraltet (Stand TYPO3 4.1)
- Andererseits beruhte der Aktualisierungsprozess auf einem Parsen der Manual-Datei `manual.sxw`, was seit TYPO3 CMS 6.0 fehlschlägt

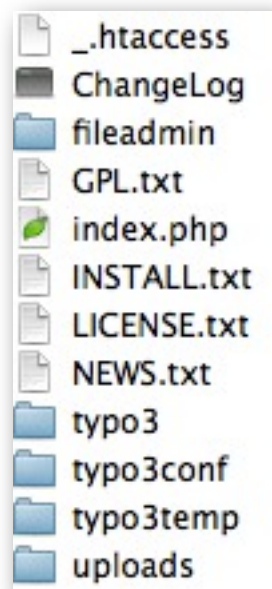


Install Tool: Entfernte Schlüssel

- Im Install-Tool (und damit einhergehend auch in der Datei `LocalConfiguration.php`) wurden folgende Optionen (Schlüssel) entfernt:
 - BE/loginLabels, BE/loginNews, BE/useOnContextMenuHandler
 - EXT/em_mirrorListURL, EXT/em_wsdlURL, EXT/extList, EXT/extList_FE, EXT/noEdit,
 - FE/defaultTypoScript_editorcfg, FE/defaultTypoScript_editorcfg., FE/simulateStaticDocuments,
 - GFX/nolconProc, GFX/TTFLocaleConv,
 - SYS/additionalAllowedClassPrefixes, SYS/caching/cacheBackends, SYS/caching/cacheFrontends, SYS/extCache, SYS/T3instID

Das Verzeichnis t3lib wurde entfernt

- Im Zuge des Umbaus wurde das Verzeichnis t3lib (und tslib) entfernt
- Sämtliche dort enthaltene Dateien wurden in das Verzeichnis **typo3/sysext/core** integriert
- Damit wurden auch alle Referenzen dazu entsprechend angepasst



Upgrade von **früheren TYPO3 Versionen**

Upgrade von TYPO3 CMS 4.5 LTS

- Es gibt eine Extension typo3-upgradereport, die eine 4.5 Installation auf Tauglichkeit überprüft
<https://github.com/nxpthx/typo3-upgradereport>
- Weitere Infos finden sich hier:
<http://forge.typo3.org/projects/typo3cms-smoothmigration/wiki>

Upgrade von früheren TYPO3 Versionen

- **Vorgehensweise**

- Backup erstellen und sicher stellen, dass ein Restore möglich ist
- Sicherstellen, dass man die letzte Patch-Level Version verwendet
- Alle Extensions zur neuesten Version updaten, die noch kompatibel ist
- Entfernen von allen nicht aktivierten Extensions
- Entfernen von allen Extensions die nicht verwendet werden (z.B. über http://typo3.org/extensions/repository/view/additional_reports)
- Upgrade Wizard im Install-Tool aufrufen und alle Schritte durchgehen
- Reference Index aktualisieren

Patches für Extensions

- Für folgende Extensions gibt es Patches, damit diese auf TYPO3 CMS 6.2 LTS laufen (Aktuelle Liste auf: <http://forge.typo3.org/projects/typo3cms-smoothmigration/wiki>)
 - realurl
 - be_secure_pw
 - ratings
 - tt_news
 - module
 - rimp_tmplselector
 - powermail

Quellen & **Ressourcen**

Quellen:

- **NEWS**
<http://typo3.org/news/article/typo3-cms-62-released/>
- **Release Notes**
<http://typo3.org/download/release-notes/typo3-62-release-notes/>
- **Technische Release Notes**
http://wiki.typo3.org/TYPO3_6.2
- **TYPO3 GIT**
<https://git.typo3.org/TYPO3v4/Core.git>
<https://git.typo3.org/TYPO3v4/CoreProjects/MVC/extbase.git>
<https://git.typo3.org/TYPO3v4/CoreProjects/MVC/fluid.git>
- **TYPO3 Bugtracker**
<http://forge.typo3.org/projects/typo3v4-core/issues>
- **NEWS.txt & Changelog**
(im Rootverzeichnis der Sourcen)

Über den **Autor**



Patrick Lobacher

Geschäftsführer typovision GmbH

- 43 Jahre, glücklich verheiratet, wohnhaft in München
- Autor von 9 Fachbüchern und > 40 Fachartikeln zum Thema TYPO3 und Webentwicklung
- Selbständig im Bereich Webentwicklung seit 1994
- Certified TYPO3 Integrator seit 2009
- Mitglied bis 2012 in den TYPO3 Core-Teams: Certification, Extbase und Content Editorial
- Mitglied bis 2012 im Expert Advisory Board der TYPO3 Association (EAB)
- Organisator des TYPO3camp Munich 2008-2013 und der T3DD12
- Speaker auf nationalen und internationalen Kongressen
- Dozent für führende Schulungsinstitute und die MVHS



Veröffentlichungen:




typovision GmbH

Powerful Connections
Technology Fast 50 2013



- Münchner Fullservice-Agentur für digitale Kommunikation
- >40 feste Mitarbeiter vor Ort (+ 15 aus festem Freelancer Pool)
- Geschäftsführer: Sebastian Böttger, Patrick Lobacher
- Hochspezialisiert auf TYPO3 seit 10 Jahren (Extbase/Fluid seit 2009)
- Platinum TYPO3 Association Member seit 3 Jahren
- Spezialisiert auf Enterprise Search (Solr, Elastic Search) seit 3 Jahren
- Fokus: Premium Webtechnologie und CMS im Open Source Bereich
- 2. Platz im Deloitte. Fast 50 „Rising Star“ Award 2013
- Agenturpräsentation unter: www.typovision.de/dieagentur
- Über 600 Projekte jeglicher Größenordnung
- **Vision:**
Wir sind der Partner des Kunden in allen Bereichen seiner digitalen Kommunikation - von der ersten Vision bis hin zur erfolgreichen Umsetzung und darüber hinaus.

Kontakt

Klassisch:

typovision GmbH

Elsenheimerstr. 7
80687 München

Fon: +49 89 45 20 59 3 - 0
Fax: +49 89 45 20 59 3 - 29

Email: info@typovision.de
Web: <http://www.typovision.de>

Twitter:

www.twitter.com/typovision
www.twitter.com/PatrickLobacher
www.twitter.com/crosscontent

Facebook:

www.facebook.com/typovision

Blog:

typoblog.de

Schulungen:

academy.typovision.de

XING:

www.xing.com/companies/typovisiongmbh
www.xing.com/profile/Patrick_Lobacher
www.xing.com/profile/Sebastian_Boettger2

LinkedIn

www.linkedin.com/company/2038844

Slideshare:

www.slideshare.net/plobacher
www.slideshare.net/typovision

Amazon:

www.amazon.de/Patrick-Lobacher/e/



Besten Dank für Ihre
Aufmerksamkeit