

TYPO3 CMS 7.2 – What's New

Übersicht der neuen Funktionen, Änderungen und Verbesserungen

Patrick Lobacher und Michael Schams

TYPO3 CMS 7.2 - What's New

Kapitelübersicht

Einführung

Backend User Interface

TSconfig & TypoScript

Änderungen im System

Extbase & Fluid

Veraltete/Entfernte Funktionen

Quellen und Autoren

Einführung (Die Fakten)

Einführung

TYPO3 CMS 7.2 – Die Fakten

- Veröffentlichungsdatum: 28. April 2015
- Releasetyp: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Hauptfokus: Frontend



Einführung

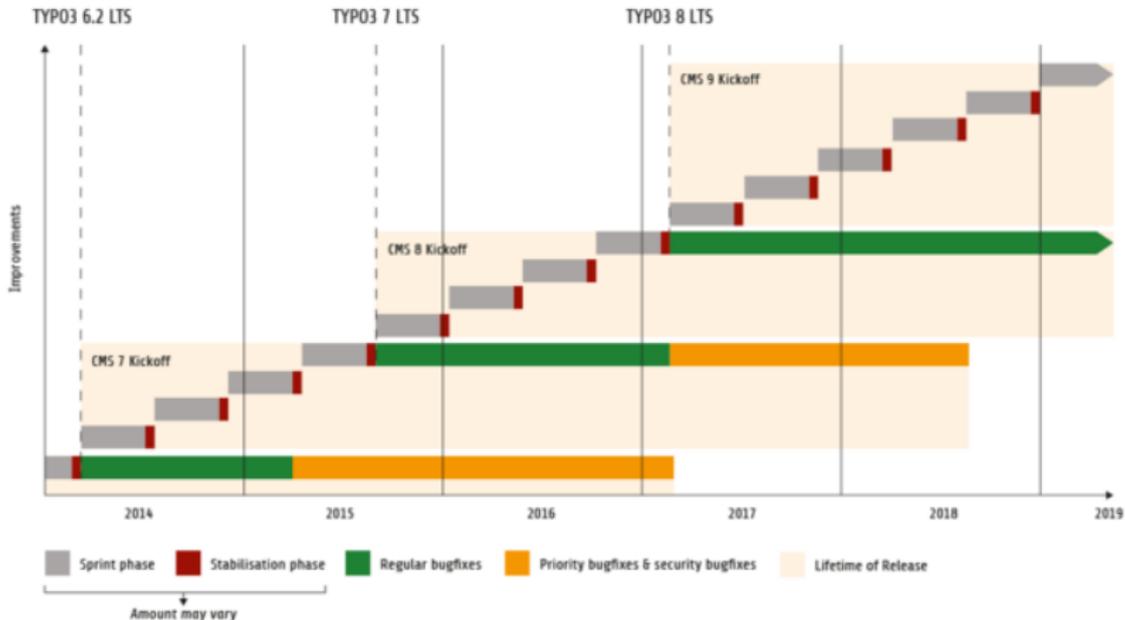
Systemvoraussetzungen

- PHP*: v5.5.0 – v5.6.x
- MySQL: v5.5.x – v5.6.x (no strict mode)
- Festplattenplatz: mindestens 200 MB
- PHP Einstellungen:
 - `memory_limit` \geq 128M
 - `max_execution_time` \geq 240s
 - PHP Kompilierungsoption `-disable-ipv6` darf nicht aktiviert sein
- Backend benötigt IE \geq 9 oder jeden anderen modernen Browser

*) weitere Details: [PHP Minimum Requirements for TYPO3 CMS 7](#)

Einführung

Release-Zyklus



Einführung

TYPO3 CMS Roadmap

Voraussichtliche Veröffentlichungen und deren Hauptfokus:

- v7.0 02/Dez/2014 Backend Overhaul Vol 1
- v7.1 24/Feb/2015 Core Cleanup & Streamlining
- **v7.2 28/Apr/2015 Frontend**
- v7.3 09/Jun/2015 Package Ecosystem, Composer und Extension Handling
- v7.4 04/Aug/2015 Backend Overhaul Vol 2
- v7.5 29/Sep/2015 (*noch unbestimmt*)
- v7.6 xx/xxx/2015 **TYPO3 CMS 7 LTS** (Long Term Release)

<https://typo3.org/typo3-cms/roadmap/>

<http://typo3.org/news/article/embrace-and-innovate-typo3-cms-7/>

Einführung

Installation

- Empfohlene Installationsschritte unter Linux/Mac OS X
(DocumentRoot ist beispielsweise `/var/www/site/htdocs`):

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/7.2
$ tar xzf typo3_src-7.2.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.2.0 typo3_src
$ ln -s typo3_src/index.php
$ ln -s typo3_src/typo3
$ touch FIRST_INSTALL
```

- Symbolische Links unter Microsoft Windows:
 - unter Windows XP/2000 kann `junction` benutzt werden
 - unter Windows Vista und Windows 7 kann `mklink` benutzt werden

Upgrade zu TYPO3 CMS 7

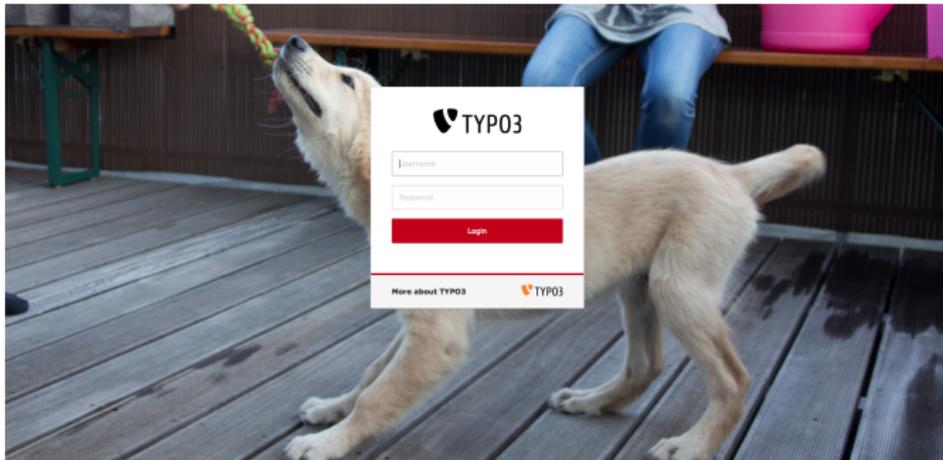
- Upgrades nur von TYPO3 CMS 6.2 LTS möglich
- TYPO3 CMS < 6.2 sollte man erst auf TYPO3 CMS 6.2 LTS aktualisieren
- Upgrade-Anleitung:
http://wiki.typo3.org/Upgrade#Upgrading_to_7.2
- Offizielles TYPO3 Guide "TYPO3 Installation and Upgrading":
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Generelles Vorgehen:
 - Prüfen, ob Mindestvoraussetzungen erfüllt sind (PHP, MySQL, etc.)
 - Das **deprecation_*.log** der TYPO3 Instanz durchsehen
 - Sämtliche Extensions auf den aktuellsten Stand bringen
 - Neuen TYPO3 Quellcode entpacken und im Install Tool den Upgrade Wizard ausführen
 - Startup Modul von Backend Benutzern überprüfen (optional)

Kapitel 1: Backend User Interface

Backend User Interface

Anpassbares Anmeldeformular

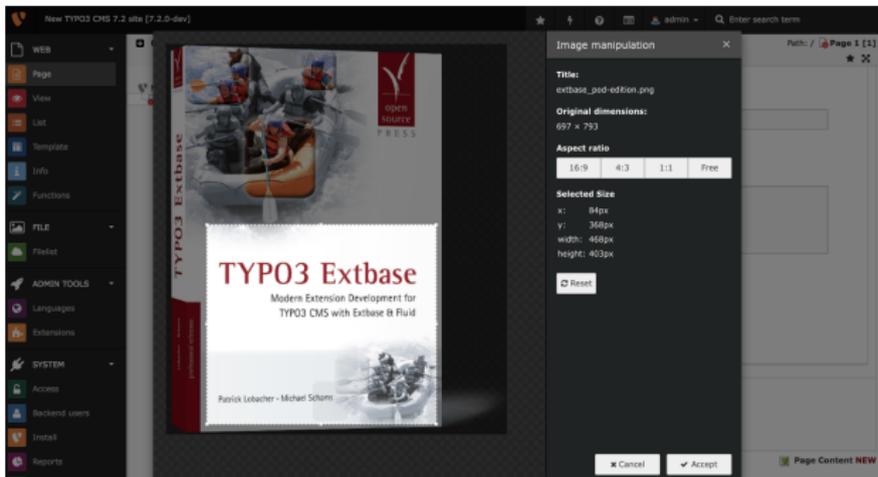
In der Systemextension backend kann sowohl ein Hintergrundbild, ein Logo und die Signalfarbe für die Anmeldeformular eingestellt werden:



Backend User Interface

Bild-Manipulation (Cropping)

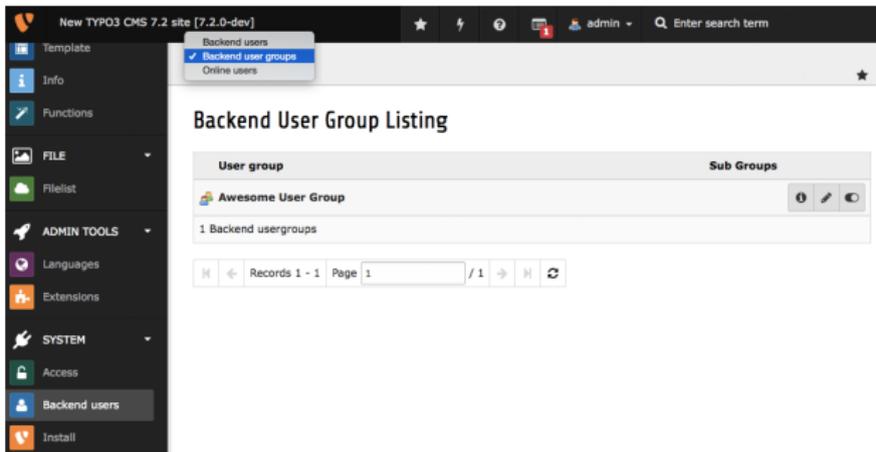
Für Bilder kann im Backend bei der Verwendung einer Referenz (z.B. in Inhalts-Elementen) ein Ausschnitt ausgewählt werden. Diese Funktion muss allerdings für den Redakteur explizit erlaubt werden ("Exclude Fields"):



Backend User Interface

Benutzergruppen

Die Backend Benutzergruppen können im Modul "Backend Users" verwaltet werden:



The screenshot shows the TYPO3 CMS Backend User Group Listing interface. The top navigation bar includes the site name "New TYPO3 CMS 7.2 site [7.2.0-dev]", a search bar, and the user "admin". A dropdown menu is open, showing "Backend users", "Backend user groups" (selected), and "Online users". The main content area is titled "Backend User Group Listing" and displays a table with the following structure:

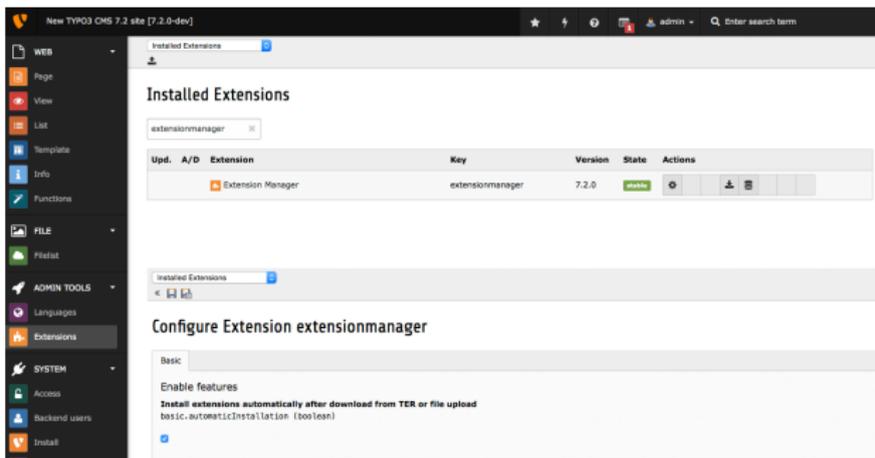
User group	Sub Groups
 Awesome User Group	  

Below the table, it indicates "1 Backend usergroups". At the bottom, there is a pagination control showing "Records 1 - 1 Page 1 / 1".

Backend User Interface

Automatische Installationen unterbinden

In den Einstellungen des Extension-Managers kann die automatische Installation von Extensions nach dem Download deaktiviert werden:



The screenshot displays the TYPO3 Backend User Interface. On the left is a dark sidebar with navigation menus for WEB, FILE, ADMIN TOOLS, and SYSTEM. The 'ADMIN TOOLS' menu is expanded, and 'Extensions' is selected. The main content area shows the 'Installed Extensions' page for the 'extensionmanager' extension. Below this, the 'Configure Extension extensionmanager' page is visible, with the 'Basic' tab selected. Under the heading 'Enable features', there is a checkbox for 'Install extensions automatically after download from TER or file upload' which is currently checked. The text below the checkbox reads 'basic.automaticInstallation (boolean)'.

Upd.	A/D	Extension	Key	Version	State	Actions
		Extension Manager	extensionmanager	7.2.0	active	  

Backend User Interface

Verbleibende Anzahl von Zeichen

Unterhalb von Textfeldern wird die verbleibende Anzahl der maximal zulässigen Zeichen angezeigt:

Edit Page "Page 1"

General Access Metadata Appearance Behaviour Resources Categories

Page
Type

Standard

Title

Page Title

Page 1

Alternative Navigation Title

This is an alternative navigation title

Remaining characters: 316

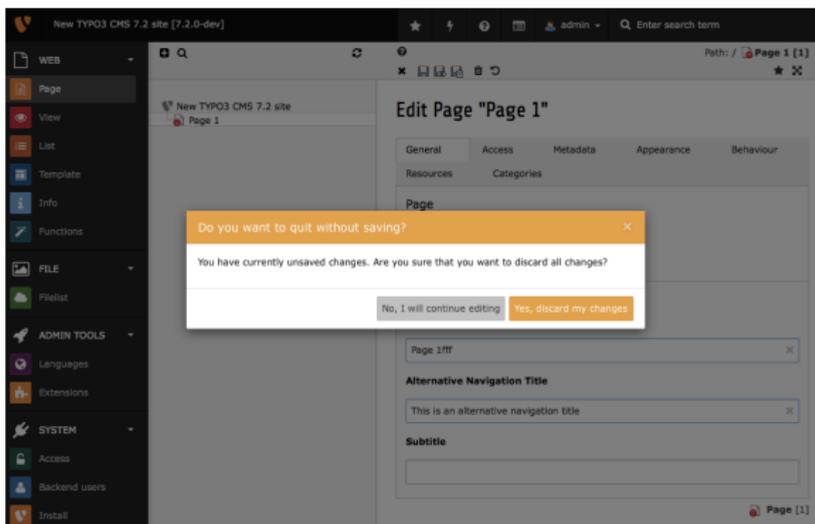
Subtitle

Page [1]

Backend User Interface

Ungespeicherte Änderungen

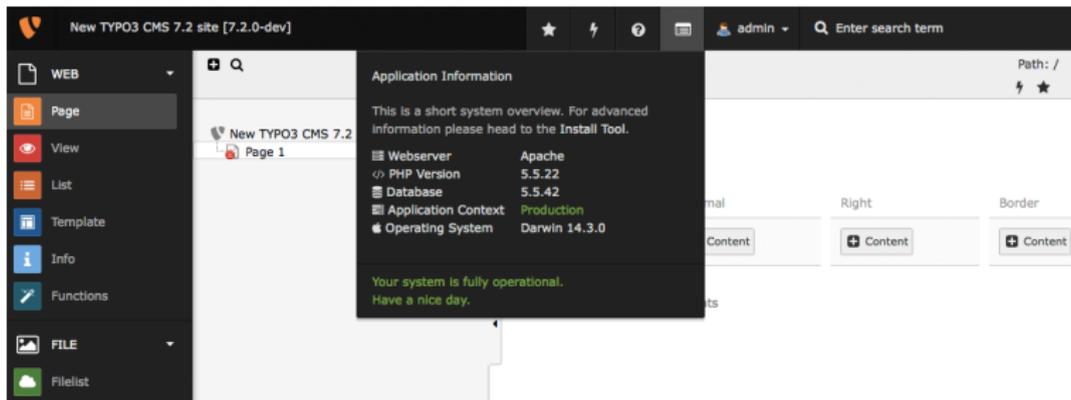
Redakteure erhalten eine Warnung, wenn in einem Editier-Formular der "Schließen"-Button geklickt wird, ohne vorher gespeichert zu haben:



Backend User Interface

Systeminformationen

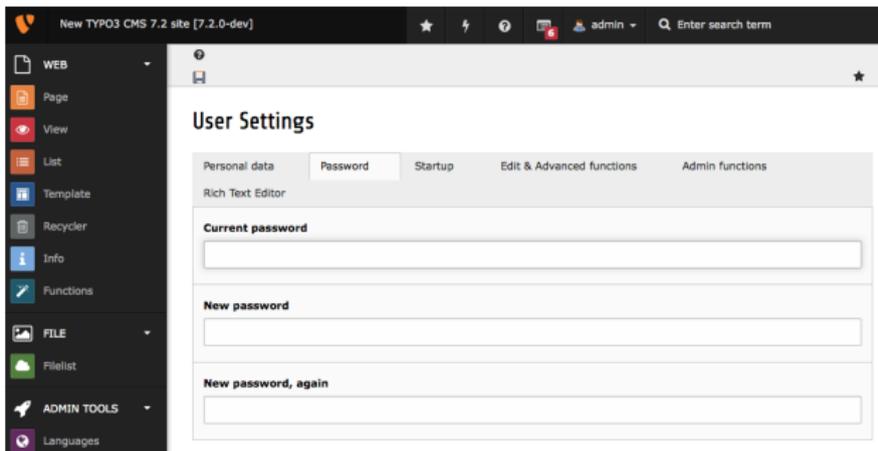
Eine Kurzübersicht der Systeminformationen kann per Klick neben dem Benutzer-Icon abgerufen werden. Die Informationen in diesem Dialog sind erweitern (siehe Kapitel "Änderungen im System"):



Backend User Interface

Passwortänderung

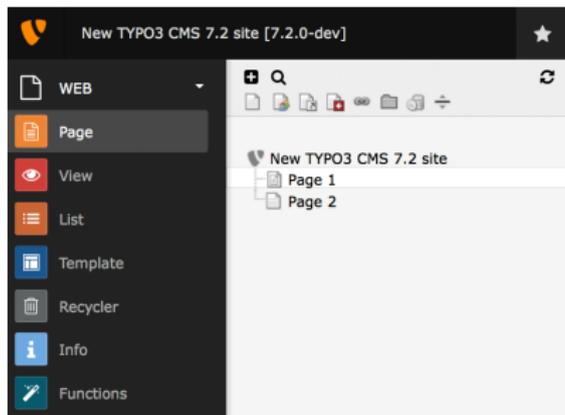
Zur Änderung des Passwortes müssen Backendbenutzer zuerst ihr aktuelles, altes Passwort eingeben:



Backend User Interface

Icon für "Show Content from Page"

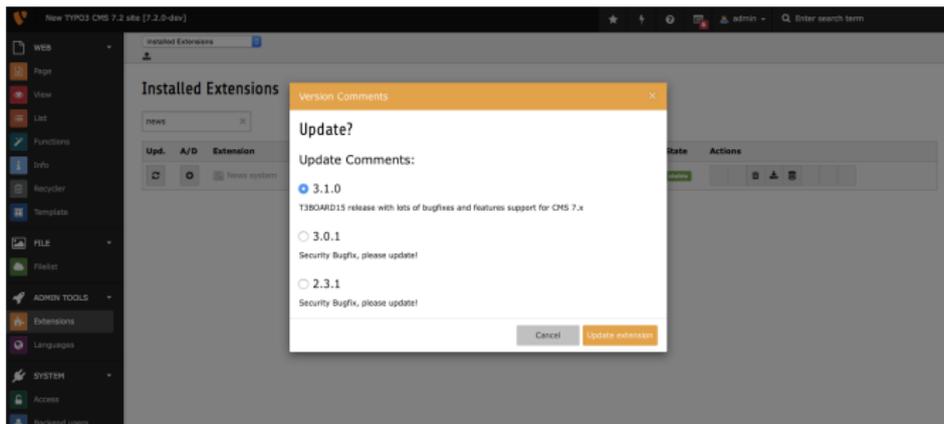
Im Seitenbaum zeigt ein neues Icon an, ob eine Seite Inhalte von einer anderen Seite enthält ("Show Content from Page"):



Backend User Interface

Extension Update

Beim Update einer Extension wird gefragt, auf welche Version aktualisieren werden soll:



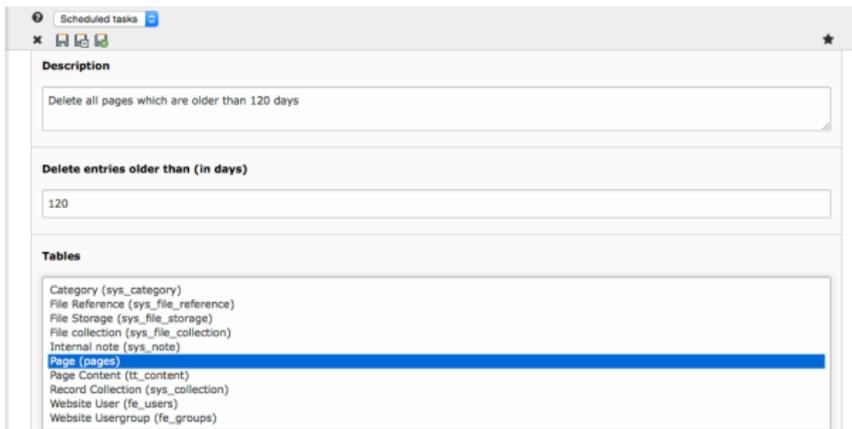
(es wird nicht mehr generell auf die letzte, verfügbare Version aktualisiert)

Backend User Interface

Recycler Task

Die Systemextension `recycler` bringt nun einen Scheduler Task mit, mit dem gelöschte Datensätze aus Content-Tabellen vollständig entfernt werden können (inkl. referenzierte Dateien, sofern vorhanden).

(max. Alter, ab wann Content gelöscht werden kann, ist konfigurierbar)



The screenshot shows the configuration for a scheduled task in the TYPO3 Backend. The task is titled "Scheduled tasks" and is currently active. The configuration includes:

- Description:** A text area containing "Delete all pages which are older than 120 days".
- Delete entries older than (in days):** A text input field containing the value "120".
- Tables:** A list of database tables with checkboxes for selection. The "Page (pages)" table is selected, indicated by a blue highlight. The other tables listed are: Category (sys_category), File Reference (sys_file_reference), File Storage (sys_file_storage), File collection (sys_file_collection), Internal note (sys_note), Page Content (tt_content), Record Collection (sys_collection), Website User (fe_users), and Website Usergroup (fe_groups).

Kapitel 2: TSconfig & TypoScript

TScnfig & TypoScript

Konfigurierbarer Vorschau-Link (1)

- Es ist nun möglich, die URL zur Vorschau einer Seite zu definieren, die durch den Button "Speichern & Vorschau" aufgerufen wird.
- Damit kann man unterschiedliche Links für Blog- oder News-Datensätze, aber auch für Inhaltselemente generieren lassen.

```
TCEMAIN.preview {
  <table name> {
    previewPageId = 123
    useDefaultLanguageRecord = 0
    fieldToParameterMap {
      uid = tx_myext_pi1[showUid]
    }
    additionalGetParameters {
      tx_myext_pi1[special] = HELLO
    }
  }
}
```

Konfigurierbarer Vorschau-Link (2)

- `previewPageId`:
UID der Seite, die für den Preview verwendet werden soll
(ohne Angabe wird die aktuelle Seite verwendet)
- `useDefaultLanguageRecord`:
definiert, ob übersetzte Datensätze die UID des Default-Datensatzes verwenden
(standardmäßig ist jenes aktiviert, default: 1)
- `fieldToParameterMap`:
Mapping (Key = Value) von Feldern des Datensatzes, die als GET-Parameter an den Link angehängt werden
- `additionalGetParameters`:
wie `fieldToParameterMap`, aber für beliebige Parameter

TSconfig & TypoScript

RTE Konfiguration: Default-Target

- Das Default-Target in der RTE Konfiguration ist nun im PageTSconfig abhängig vom Typ einstellbar

```
buttons.link.[ type ].properties.target.default = ...
```

- Als "type" sind folgende Werte zulässig:
(weitere können via Extensions eingebracht werden)
 - page
 - file
 - url
 - mail
 - spec

Leere HTML-Tags im HTMLparser löschen

- Es ist nun möglich, leere HTML-Tags im HTMLparser zu löschen

```
stdWrap {
    // Hier werden alle leeren HTML-Tags entfernt
    HTMLparser.stripEmptyTags = 1
    // Hier werden nur leere h2 und h3 Tags entfernt
    HTMLparser.stripEmptyTags.tags = h2, h3
}

RTE.default.proc.entryHTMLparser_db {
    stripEmptyTags = 1
    stripEmptyTags.tags = p
    stripEmptyTags.treatNonBreakingSpaceAsEmpty = 1
}
```

- Da der HtmlParser unbekannte Tags grundsätzlich entfernt, ist es ratsam, diese zunächst zu behalten:

```
HTMLparser.keepNonMatchedTags = 1
```

Tsconfig & TypeScript

Diverses

- Der Button für "Abkürzung" (engl. *abbreviation*) im RTE kann nun in der PageTsconfig ausgeblendet werden (da nicht mehr HTML5 konform):

mögliche Werte sind:

acronym, definedAcronym, abbreviation, definedAbbreviation

buttons.abbreviation.removeFieldsets = acronym,definedAcronym

- Die Eigenschaft `inlineLanguageLabel` des Objekts PAGE kann nun auch mit LLL:-Referenzen umgehen

Kapitel 3: Änderungen im System

Änderungen im System

SVG Support im Core

- Der Core unterstützt nun SVG-Bilder ("Scalable Vector Graphics")
- Wenn ein SVG-Bild skaliert wird, wird kein prozessiertes Bild abgelegt, dafür aber die neuen Größenangaben in einem Datensatz `sys_file_processedfile` gespeichert (außer, das Bild wird - z.B. durch Cropping - weiterverarbeitet).
- Zudem wurde ein zusätzlicher Fallback eingebaut, falls ImageMagick/GraphicsMagick nicht in der Lage sein sollte, die Dimensionen zu berechnen: in diesem Fall wird das XML ausgelesen.
- SVG wurde außerdem zur Liste der zulässigen Bildtypen hinzugefügt:
`$GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext']`

Änderungen im System

Erweiterung der FAL-Treiber

- Um die Performance der Dateiliste bei (Remote-)Storages innerhalb von FAL zu erhöhen, ist es notwendig, die Sortierung und das Ermitteln der Anzahl direkt im Treiber zu erledigen. Dafür wurden zwei neue Parameter `sort` und `sortRev` eingebracht:

```
public function getFilesInFolder($folderIdentifier, $start = 0, $numberOfItems = 0,
    $recursive = FALSE, array $filenameFilterCallbacks = array(), $sort = '', $sortRev = FALSE);
```

```
public function getFoldersInFolder($folderIdentifier, $start = 0, $numberOfItems = 0,
    $recursive = FALSE, array $folderNameFilterCallbacks = array(), $sort = '', $sortRev = FALSE);
```

- Außerdem wurden zwei neue Methoden eingeführt:

```
public function getFilesInFolderCount($folderIdentifier, $recursive = FALSE,
    array $filenameFilterCallbacks = array());
```

```
public function getFoldersInFolderCount($folderIdentifier, $recursive = FALSE,
    array $folderNameFilterCallbacks = array());
```

Änderungen im System

Backend Routing API (1)

- Der Core enthält nun eine Backend Routing API, welche die Entry Points ins Backend verwaltet
- Die Routing API wurde vom Symfony Routing Framework inspiriert und ist weitgehend kompatibel mit dieser (auch wenn für TYPO3 zur Zeit nur ca. 20% genutzt werden)
- Grundsätzlich existieren hierfür drei Klassen:
 - class Route: enthält Angaben zum Pfad und Optionen
 - class Router: API, um die Route zu matchen
 - class UrlGenerator: generiert die URL

Änderungen im System

Backend Routing API (2)

- Routen werden dabei in folgender Datei in der entsprechenden Extension definiert: `Configuration/Backend/Routes.php` (siehe Systemextension `backend` als Beispiel)
- Weitere Informationen zur Backend Routing API:
<http://wiki.typo3.org/Blueprints/BackendRouting>

Änderungen im System

Neue Systemextension für Media Inhaltselemente

- Neue Systemextension "mediace" enthält folgende cObjects:
 - MULTIMEDIA
 - MEDIA
 - SWFOBJECT
 - FLOWPLAYER
 - QTOBJECT
- Die Inhaltselemente `media` und `multimedia` wurden ebenfalls in die Systemextension verschoben, ebenso der "Media Wizard Provider"
- Die Extension ist standardmäßig **nicht** installiert!

Änderungen im System

Fremd-Bibliotheken an neuer Stelle

- Sämtliche Fremd-Bibliotheken werden nicht mehr in Packages/Libraries, sondern in typo3/contrib/vendor abgelegt
- Grundsätzlich ist dafür die Installation der Bibliotheken mittels composer install notwendig
- Probleme gibt es beim Upgrade einer Installation, wenn dort phpunit verwendet wurde! Dies kann wie folgt behoben werden:

```
# cd htdocs/  
# rm -rf typo3/contrib/vendor/ bin/ Packages/Libraries/ composer.lock  
# composer install
```

Änderungen im System

API für JavaScript Notifikationen

- Neue API, um JavaScript Notifikationen zu erzeugen:

```
// Bisheriger (veralteter) Weg:  
top.TYP03.Flashmessages.display(TYP03.Severity.notice)
```

```
// Neuer Weg:  
top.TYP03.Notification.notice(title, message)
```

- Es existieren folgende API-Funktionen:
(Parameter `duration` ist optional und standardmäßig auf 5s eingestellt)
 - `top.TYP03.Notification.notice(title, message, duration)`
 - `top.TYP03.Notification.info(title, message, duration)`
 - `top.TYP03.Notification.success(title, message, duration)`
 - `top.TYP03.Notification.warning(title, message, duration)`
 - `top.TYP03.Notification.error(title, message, duration)`

Änderungen im System

Systeminformationen (1)

- Das Dropdown mit Systeminformationen kann über folgenden Slot erweitert werden:

```
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class);  
  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Backend\Backend\ToolBarItems\SystemInformationToolBarItem::class,  
    'getSystemInformation',  
    \Vendor\Extension\SystemInformation\Item::class,  
    'getItem'  
);
```

Änderungen im System

Systeminformationen (2)

- Zur Ansprache benötigt man die Klasse `Item` und dazugehörig die Methode `getItem()` innerhalb einer Extension

`EXT:extension\Classes\SystemInformation\Item.php`:

```
class Item {
    public function getItem() {
        return array(array(
            'title' => 'The title shown on hover',
            'value' => 'Description shown in the list',
            'status' => SystemInformationHookInterface::STATUS_OK,
            'count' => 4,
            'icon' => \TYPO3\CMS\Backend\Utility\IconUtility::getSpriteIcon(
                'extensions-example-information-icon')
        ));
    }
}
```

Änderungen im System

Systeminformationen (3)

- Das Icon `extensions-example-information-icon` wird in der Datei `ext_localconf.php` registriert:

```
\TYPO3\CMS\Backend\Sprite\SpriteManager::addSingleIcons(  
    array(  
        'information-icon' => \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extRelPath(  
            $_EXTKEY) . 'Resources/Public/Images/Icons/information-icon.png'  
        ),  
        $_EXTKEY  
    );
```

Änderungen im System

Systeminformationen (4)

- Nachrichten werden am unteren Ende des Dropdowns angezeigt
- Über den folgenden Slot können eigene Nachrichten eingebracht werden:

```
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class);  
  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Backend\Backend\ToolBarItems\SystemInformationToolBarItem::class,  
    'loadMessages',  
    \Vendor\Extension\SystemInformation\Message::class,  
    'getMessage'  
);
```

Änderungen im System

Systeminformationen (5)

- Zur Ansprache benötigt man die Klasse Message und dazugehörig die Methode getMessage() innerhalb einer Extension

EXT:extension\Classes\SystemInformation\Message.php:

```
class Message {
    public function getMessage() {
        return array(array(
            'status' => SystemInformationHookInterface::STATUS_OK,
            'text' => 'Something went wrong. Take a look at the reports module.'
        ));
    }
}
```

Änderungen im System

Einstellungen für Bild-Manipulation (1)

- Folgende Einstellungen können über TypoScript getätigt werden:

```
# Cropping fuer alle Bilder deaktivieren
```

```
tt_content.image.20.1.file.crop =
```

```
# Ueberschreiben/Setzen der Cropping Eigenschaften
```

```
# offsetX,offsetY,width,height
```

```
tt_content.image.20.1.file.crop = 50,50,100,100
```

Änderungen im System

Einstellungen für Bild-Manipulation (2)

- Das Cropping kann auch in Fluid verwendet werden:

```
# Cropping fuer alle Bilder deaktivieren
```

```
<f:image image="{imageObject}" crop="" ></f:image>
```

```
# Ueberschreiben/Setzen der Cropping Eigenschaften
```

```
# offsetX,offsetY,width,height
```

```
<f:image image="{imageObject}" crop="50,50,100,100" ></f:image>
```

Änderungen im System

Einstellungen für Bild-Manipulation (3)

- Im TCA wird das Image-Cropping wie folgt zur Verfügung gestellt:

- Column Type: image_manipulation
- Config file_field: string (default: uid_local)
- Config enableZoom: boolean (default: FALSE)
- Config allowedExtensions: string
(default: \$GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext'])
- Config ratios: array, default:

```
array(  
    '1.7777777777777777' => '16:9',  
    '1.3333333333333333' => '4:3',  
    '1' => '1:1',  
    'NaN' => 'Free'  
)
```

Änderungen im System

Zusätzliche Parameter für HTMLparser userFunc

- Die userFunc im HTMLparser kann nun zusätzliche Parameter aufnehmen:

```
myobj = TEXT
myobj.value = <a href="/" class="myclass">MyText</a>
myobj.HTMLparser.tags.a.fixAttrib.class {
    userFunc = Tx\MyExt\Myclass->htmlUserFunc
    userFunc.myparam = test
}
```

- Diese können in einer Extension wie folgt abgerufen werden:

```
function htmlUserFunc(array $params, HtmlParser $htmlParser) {
    // $params['attributeValue'] enthaelt den Wert der
    // verarbeiteten Eigenschaft - hier also "myclass"
    // $params['myparam'] enthaelt den Wert "test"
    ...
}
```

Änderungen im System

Locking API (1)

- Es wurde eine neue Locking-API eingeführt, welche verschiedene Locking-Methoden (SimpleFile, Semaphore, ...) zur Verfügung stellt
- Eine Locking-Methode muss dabei das LockingStrategyInterface implementieren

```
$lockFactory = GeneralUtility::makeInstance(LockFactory::class);  
$locker = $lockFactory->createLocker('someId');  
$locker->acquire() || die('Could not acquire lock.');
```

...

```
$locker->release();
```

Änderungen im System

Locking API (2)

- Man kann außerdem "non-blocking" Locks realisieren:

```
$lockFactory = GeneralUtility::makeInstance(LockFactory::class);
$locker = $lockFactory->createLocker(
    'someId',
    LockingStrategyInterface::LOCK_CAPABILITY_SHARED |
    LockingStrategyInterface::LOCK_CAPABILITY_NOBLOCK
);
try {
    $result = $locker->acquire(LockingStrategyInterface::LOCK_CAPABILITY_SHARED |
        LockingStrategyInterface::LOCK_CAPABILITY_NOBLOCK);
    catch (\RuntimeException $e) {
        if ($e->getCode() === 1428700748) {
            // einige Prozesse haben noch ein Lock
            // daher sollte etwas in der Zwischenzeit gemacht werden
            ...
        }
    }
}
if ($result) {
    $locker->release();
}
```

Änderungen im System

Signal nach Installation von Extensions

■ In der Methode

`\TYPO3\CMS\Extensionmanager\Utility\InstallUtility::install()`
wurde ein Signal eingebaut, welches emittiert wird, sobald eine
Extension fertig installiert ist und alle Imports/Updates durchgelaufen
sind

```
// Aufruf
$this->emitAfterExtensionInstallSignal($extensionKey);

// Methode
protected function emitAfterExtensionInstallSignal($extensionKey) {
    $this->signalSlotDispatcher->dispatch(
        __CLASS__,
        'afterExtensionInstall',
        array($extensionKey, $this)
    );
}
```

Änderungen im System

Registry für Text-Extraktoren

- Der Core enthält nun eine Registry um Text-Extraktoren anzumelden
- Dabei prüft `canExtractText()` ob eine Extrahierung möglich ist und `extractText()` führt diese durch
- Die Registrierung erfolgt in `ext_localconf.php`:

```
$textExtractorRegistry = \TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::
    getInstance();
$textExtractorRegistry->registerTextExtractor(
    \TYPO3\CMS\Core\Resource\TextExtraction\PlainTextExtractor::class
);
```

- Die Verwendung erfolgt folgendermaßen:

```
$textExtractorRegistry = \TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::
    getInstance();
$extractor = $textExtractorRegistry->getTextExtractor($file);
if($extractor !== NULL) {
    $content = $extractor->extractText($file);
}
```

Änderungen im System

Diverses

- Alle Web-Bibliotheken (wie z.B. Twitter Bootstrap, jQuery, Font Awesome usw.) verwenden nun "Bower" (<http://bower.io>) zur Installation, und sind nicht mehr im TYPO3 Core Git enthalten
 - `bower install` führt eine Installation durch
 - `bower update` führt ein Update durch(die zugehörige Datei `bower.json` befindet sich im Verzeichnis `Build`)
- Ein laufender Scheduler Task kann nun in der Kommandozeile mit der Option `-s` wieder gestoppt werden
- Der "Processing" Ordner eines Storage kann nun auch außerhalb von diesem liegen (z.B. bei read-only Storages)
- Man kann nun auf die ID der ursprünglich angefragten Seite über das TSFE zugreifen: `$TSFE->getRequestedId()`

Kapitel 4: Extbase & Fluid

Callouts anstelle von FlashMessages

- An einigen Stellen im Backend werden nun Callouts anstelle von FlashMessages verwendet
- Dafür wurde ein neuer Fluid ViewHelper `be.infobox` eingeführt:

```
<f:be.infobox title="Message title">  
    Inhalt der Nachricht  
</f:be.infobox>
```

```
<f:be.infobox  
    title="Message title"  
    message="your box content"  
    state="-2"  
    iconName="check"  
    disableIcon="TRUE" />
```

format.case ViewHelper

- Es gibt nun einen `format.case` ViewHelper, der die Schreibweise von Strings verändert:
 - `upper`: Führt zu "UPPERCASE" (Großbuchstaben)
 - `lower`: Führt zu "lowercase" (Kleinbuchstaben)
 - `capital`: Führt zu einem großen Anfangsbuchstaben
 - `uncapital`: Führt zu einem kleinen Anfangsbuchstaben

```
// Fuehrt zu "SOME TEXT WITH MIXED CASE"  
<f:format.case>Some TeXt WiTh miXed cAse</f:format.case>
```

```
// Fuehrt zu "SomeString"  
<f:format.case mode="capital">someString</f:format.case>
```

Extbase & Fluid

Diverses

- Parameter `cHash` wird nicht mehr an URLs angehängt, die auf eine Action leiten, welche nicht gecached wird oder wenn der Request nicht gecached wird.

Kapitel 5:

Veraltete und entfernte Funktionen

Veraltete/Entfernte Funktionen

Refactoring der FormEngine (1)

- Die FormEngine Klassen wurden einem Refactoring unterzogen, welches Auswirkungen auf Eigenschaften und Methoden der Klassen hat
- Viele interne Eigenschaften, wie `FormEngine->$defaultInputWidth` oder `FormEngine->$RTEenabled` werden ab sofort ignoriert
- Die Eigenschaften `FormEngine->$allowOverrideMatrix` und `SuggestElement->class` sind ab TYPO3 CMS 7.2 *protected*
- Wenn im TCA `type=none` gewählt wird, wird nun nicht mehr ein FormEngine Objekt, sondern ein NoneElement Objekt zurückgegeben

Veraltete/Entfernte Funktionen

Refactoring der FormEngine (2)

- Under anderem sind die folgenden Methoden als **deprecated** markiert:
 - FormEngine->renderWizards()
 - FormEngine->dbFileIcons()
 - FormEngine->getClipboardElements()
 - FormEngine->getSingleField_typeNone_render()
 - FormEngine->formMaxWidth()
 - FormEngine->addItem()
 - SuggestElement->init()
 - ...

Tipp: Wenn man auf diese Methoden aufgebaut hat, sollte man das DeprecationLog analysieren, um herauszufinden, wohin diese Methoden portiert wurden

Veraltete/Entfernte Funktionen

Refactoring der FormEngine (3)

- Die folgenden Methoden wurden **umbenannt**:

ALT: \TYPO3\CMS\Backend\Form\Element\SuggestElement

NEU: \TYPO3\CMS\Backend\Form\Wizard\SuggestWizard

ALT: \TYPO3\CMS\Backend\Form\Element\SuggestDefaultReceiver

NEU: \TYPO3\CMS\Backend\Form\Wizard\SuggestWizardDefaultReceiver

ALT: \TYPO3\CMS\Backend\Form\Element\VaueSlider

NEU: \TYPO3\CMS\Backend\Form\Wizard\ValueSliderWizard

Veraltete/Entfernte Funktionen

Entry Points im Backend

- Die folgenden Entry Points im Backend wurden verändert:

■ typo3/logout.php	(logout)
■ typo3/alt_file_navframe.php	(file_navframe)
■ typo3/dummy.php	(dummy)
■ typo3/browser.php	(browser)
■ typo3/db_new.php	(db_new)
■ typo3/show_item.php	(show_item)

- Die URLs können wie folgt ermittelt werden:

```
\TYPO3\CMS\Backend\Utility\BackendUtility::getModuleUrl(...)
```

Zum Beispiel:

```
\TYPO3\CMS\Backend\Utility\BackendUtility::getModuleUrl('logout')
```

Veraltete/Entfernte Funktionen

Diverses (1)

- URLs zu Backend-Modulen, welche durch `mod.php` dispatched wurden, werden nun nicht mehr im Array `$GLOBALS['MCONF']['_']` gespeichert, sondern durch `BackendUtility::getModuleUrl()` erzeugt
- Die Möglichkeit, den lokalen Pfad einer FAL-Datei via TypoScript zu ermitteln, wurde **entfernt**
(`a.value.data = file:current:localPath`)
- Die folgenden `tt_content` TCA-Felder wurden in die Extension `EXT:compatibility6` verschoben:
 - `altText`
 - `imagecaption`
 - `imagecaption_position`
 - `image_link`
 - `longdescURL`
 - `titleText`

Veraltete/Entfernte Funktionen

Diverses (2)

- Die Backend-Benutzeranmeldung wurde einem Refactoring unterzogen und das Signal `LoginController::SIGNAL_RenderLoginForm` **entfernt**
- Außerdem wurden die folgenden Methoden des LoginControllers **entfernt**:
 - `LoginController::makeLoginBoxImage`
 - `LoginController::wrapLoginForm`
 - `LoginController::makeLoginNews`
 - `LoginController::makeLoginForm`
 - `LoginController::makeLogoutForm`
- Die Fremd-Bibliothek `websvg` wurden aus dem TYPO3 CMS Core **entfernt**. Dazu gehören auch TypoScript Optionen (`page.javascriptLibs.SVG.*`), und Methoden des PageRenderers, wie z.B. `$pageRenderer->loadSvg()`

Veraltete/Entfernte Funktionen

Diverses (3)

- Folgende Schlüssel unter `mod.web_info.menu.function` wurden **umbenannt** (dies hat vor allem Auswirkungen auf `PageTSconfig`):

```
tx_cms_webinform_page -> TYPO3\CMS\Frontend\Controller\PageInformationController
tx_cms_webinform_lang -> TYPO3\CMS\Frontend\Controller\TranslationStatusController
tx_belog_webinform -> TYPO3\CMS\Belog\Module\BackendLogModuleBootstrap
tx_infopagetsconfig_webinform -> TYPO3\CMS\InfoPagetsconfig\Controller\
    InfoPageTypescriptConfigController
tx_linkvalidator_ModFuncReport -> TYPO3\CMS\Linkvalidator\Report\LinkValidatorReport
```

- Der Parameter `$returnHTML` der Methode `\TYPO3\CMS\Core\Utility\DebugUtility::debugRows()` wird nicht mehr verwendet und ist daher als **deprecated** markiert

Veraltete/Entfernte Funktionen

Diverses (4)

- Die Funktion `config.prefixLocalAnchors` ist **deprecated**, ebenso wie die dazugehörigen Methoden im `TypoScriptFrontendController`: `prefixLocalAnchorsWithScript()` und `doLocalAnchorFix()`
- Die Methode `$TSFE->checkFileInclude()` im globalen `FrontendController` ist **deprecated**. Man sollte stattdessen den Autoloader oder `$TSFE->tmpl->getFileName()` verwenden.
- Die beiden cObjects `COBJ_ARRAY` (alias für `COA`) und `CASEFUNC` (alias für `CASE`) wurden in die Extension `EXT:compatibility6` verschoben (und als **deprecated** markiert) und stehen somit standardmäßig nicht mehr zur Verfügung).
- Die DataHandler Eigenschaft `stripslashes_values` wurde als **deprecated** markiert.

Veraltete/Entfernte Funktionen

Diverses (5)

- Der "ThumbnailView", sowie `thumbs.php` und `BackendUtility::getThumbNail()` sind ab sofort **deprecated** und werden mit TYPO3 CMS Version 8 **entfernt** (siehe `BackendUtility::thumbCode()` zur Inspiration).
- Im LinkValidator Scheduler Task ändert sich der Namespace `mod.tx_linkvalidator` zu `mod.linkvalidator`.

Kapitel 6: Quellen und Autoren

Quellen und Autoren

Quellennachweis

TYPO3 News:

- <http://typo3.org/news>

Release Infos:

- http://wiki.typo3.org/TYPO3\CMS_7.2.0
- [INSTALL.md](#) and [Changelog](#)
- [typo3/sysex/core/Documentation/Changelog/7.2/*](http://typo3.org/sysex/core/Documentation/Changelog/7.2/*)

TYPO3 Bug-/Issuetracker:

- <https://forge.typo3.org/projects/typo3cms-core>

TYPO3 Git Repositories:

- <https://git.typo3.org/Packages/TYPO3.CMS.git>
- <https://git.typo3.org/Packages/TYPO3.Fluid.git>

TYPO3 CMS What's New Slides:

Patrick Lobacher

(Recherche, Informationsdokumentation und deutsche Version)

Michael Schams

(Project Leader und englische Version)

Übersetzungen von:

Andrey Aksenov, Paul Blondiaux, Sergio Catala, Michel Mix,
Sinisa Mitrovic, Angeliki Plati, Nena Jelena Radovic und Roberto Torresani

<http://typo3.org/download/release-notes/whats-new>

Lizenziert unter Creative Commons BY-NC-SA 3.0

