

TYPO3 CMS 7.5 – What's New

Übersicht der neuen Funktionen, Änderungen und Verbesserungen

Patrick Lobacher (Vorstand pluswerk AG)
<http://www.pluswerk.ag>



TYPO3 CMS 7.5 - 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 7.5 – Die Fakten

- Veröffentlichungsdatum: 29. September 2015
- Releasetyp: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Hauptfokus: Finalization



Einführung

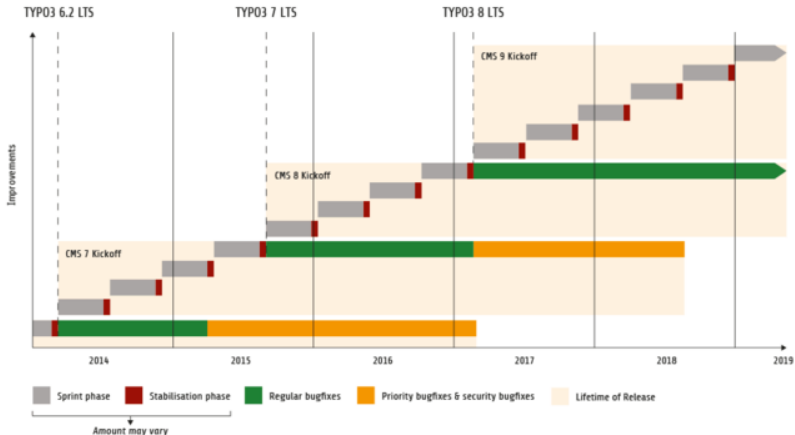
Systemvoraussetzungen

- 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 16/Jun/2015 Package Ecosystem, Composer
- v7.4 04/Aug/2015 Backend Overhaul Vol 2
- v7.5 29/Sep/2015 Finalization
- v7 LTS Okt/Nov/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/7.5
$ tar xzf typo3_src-7.5.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.5.0 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

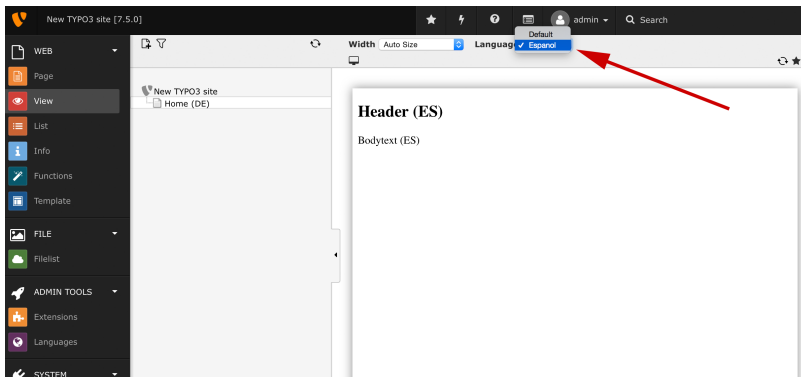
- Upgrades nur von TYPO3 CMS 6.2 LTS möglich
- TYPO3 CMS < 6.2 sollte man erst auf TYPO3 CMS 6.2 LTS aktualisieren
- Upgrade-Anleitung:
http://wiki.typo3.org/Upgrade#Upgrading_to_7.5
- Offizielles TYPO3 Guide "TYPO3 Installation and Upgrading":
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Generelles 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

Sprachauswahl im Modul "View"

Das Modul WEB->View bietet nun eine komfortable Sprachauswahl.
(kann durch `mod.SHARED.view.disableLanguageSelector = 1` deaktiviert werden)









Backend User Interface

Inhaltselement `textmedia`

Ein neues Inhaltselement **"Text & Media"**, fasst die bisher bekannten Elemente `text`, `image` und `textpic` zusammen.

New content element

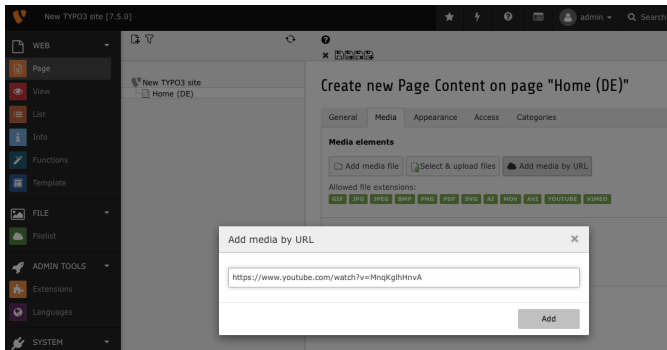
Please select the type of page content you wish to create:

Typical page content	Special elements	Form elements	Plugins
 Header Only Adds a header only.			
 Images Only Any number of images aligned in columns and rows with a caption.			
 Bullet List A single bullet list.			
 Table A simple table.			
 Text & Media Any number of media wrapped right around a regular text element.			
 File Links Makes a list of files for download.			

Backend User Interface

YouTube- und Vimeo-Dateien

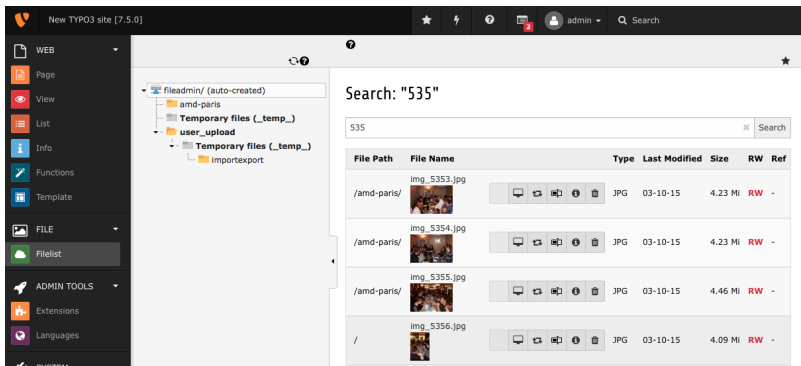
Über das erwähnte Inhaltselement **"Text & Media"** können auch externe YouTube- und Vimeo-Dateien, sowie lokale Dateien eingefügt werden.



Backend User Interface

Suche im Filelist Modul

Im Modul "Filelist" gibt es eine Suche, die rekursiv durch alle Ordner sucht



The screenshot shows the TYPO3 Backend Filelist module interface. On the left is a sidebar with navigation options: WEB (Page, View, List, Info, Functions, Template), FILE (Filelist), and ADMIN TOOLS (Extensions, Languages). The main area displays a file tree for 'fileadmin/ (auto-created)' with folders like 'amd-paris', 'Temporary files (_temp_)', and 'user_upload'. A search bar at the top right contains the text '535'. Below the search bar, a table lists search results for files named 'img_535*.jpg'.

File Path	File Name	Type	Last Modified	Size	RW	Ref
/amd-paris/	img_5353.jpg	JPG	03-10-15	4.23 Mi	RW	-
/amd-paris/	img_5354.jpg	JPG	03-10-15	4.23 Mi	RW	-
/amd-paris/	img_5355.jpg	JPG	03-10-15	4.46 Mi	RW	-
/	img_5356.jpg	JPG	03-10-15	4.09 Mi	RW	-

Kapitel 2: TSconfig & TypoScript

TSconfig & TypeScript

Conditions für TypeScript-Include

- Der INCLUDE_TYPOSCRIPT Tag besitzt nun das optionale Attribut "condition", welches es ermöglicht, die Datei (oder das Verzeichnis) nur dann zu inkludieren, wenn die Condition erfüllt ist:

```
// TypeScript nur laden, wenn Benutzer eingeloggt ist:  
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypeScript/feuser.ts"  
  condition="[loginUser = *]">
```

```
// TypeScript nur laden, wenn ApplicationContext gesetzt ist:  
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypeScript/staging.ts"  
  condition="applicationContext = /^Production\\/Staging\\/Server\\d+$/">
```


TCA-Option, um Datum Feldweise auszublenden

- Es gibt nun eine TCA-Option `disableAgeDisplay`, um die Anzeige des Datums auszublenden
Voraussetzung hierfür ist, dass der Typ des Feldes `input` ist, und `eval` auf `date` gesetzt ist

```
$GLOBALS['TCA']['tt_content']['columns']['date']['config']['disableAgeDisplay'] = true;
```

TSconfig & TypeScript

Inline Sprachlabels mit TypeScript (1)

- Man kann nun Sprachdateien mittels TypeScript auslesen und als Inline-Array in den Quelltext schreiben, um z.B. per JavaScript darauf zuzugreifen
- Folgende Optionen sind möglich:
 - `selectionPrefix`:
nur Schlüssel, die mit diesem Prefix anfangen, werden ermittelt
 - `stripFromSelectionName`:
String, der von jedem Schlüssel entfernt wird
 - `errorMode`:
Mode, wenn die Sprachdatei nicht gefunden wird
(0: Eintrag im Syslog vornehmen, 1: ignorieren, 3: Exception generieren)

TSconfig & TypeScript

Inline Sprachlabels mit TypeScript (2)

■ Beispiel:

```
page = PAGE
page.inlineLanguageLabelFiles {
    someLabels = EXT:myExt/Resources/Private/Language/locallang.xlf
    someLabels.selectionPrefix = idPrefix
    someLabels.stripFromSelectionName = strip_me
    someLabels.errorMode = 2
}
```

■ Ausgabe:

```
<script type="text/javascript">
/**]
    var TYPO3 = TYPO3 || {};
    TYPO3.lang = {"firstLabel":["source":"first Label","target":"erstes Label"],
        "secondLabel":["source":"second Label","target":"zweites Label"]};
/*]]&gt;*/
&lt;/script&gt;</pre></div><div data-bbox="22 936 210 958" data-label="Page-Footer"><p>TYPO3 CMS 7.5 - What's New</p></div><div data-bbox="806 886 970 951" data-label="Page-Footer"><img alt="TYPO3 logo" data-bbox="806 886 970 951"/>The logo for TYPO3, consisting of an orange stylized 'T' shape followed by the text 'TYPO3' in a bold, sans-serif font.</div>
```

Workspace Preview per TScnfig

- Standardmäßig erzeugt TYPO3 lediglich Vorschau-Links für die Tabellen `tt_content`, `pages` und `pages_language_overlay`
- Dies kann nun per PageTScnfig angepasst werden:

```
# Verwendung der Seite 123 fuer Workspace Preview (fuer alle Tabellen)
options.workspaces.previewPageId = 123
```

```
# Verwendung des Feldes pid (fuer alle Tabellen)
options.workspaces.previewPageId = field:pid
```

```
# Verwendung der Seite 123 fuer Workspace Preview (fuer die Tabelle tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = 123
```

```
# Verwendung des Feldes pid fuer Workspace Preview (fuer die Tabelle tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = field:pid
```

Bildqualität kann per SourceCollection gesetzt werden

- Die Bildqualität jeder sourceCollection kann nun konfiguriert werden
- Dies überschreibt die Einstellungen, die im Install Tool gemacht wurden und in der Datei LocalConfiguration.php gespeichert sind

```
# fuer kleine Retina Bilder  
tt_content.image.20.1.sourceCollection.smallRetina.quality = 80  
  
# fuer groessere Retina Bilder  
tt_content.image.20.1.sourceCollection.largeRetina.quality = 65
```

TSconfig & TypeScript

Count für Split hinzugefügt

- Es wurde eine neue Eigenschaft `returnCount` zur `stdWrap`-Funktion `split` hinzugefügt, die die Anzahl der Elemente nach dem Split enthält

```
1 = TEXT
1 {
  value = x,y,z,1,2,3,a,b,c
  split.token = ,
  split.returnCount = 1
}

# result: 9
```

Handling von Backend-Layouts vereinfacht (1)

- Das Handling, um Backend-Layouts mit Templates für die Frontend-Ausgabe zu versehen, wurde vereinfacht, indem die Option `pagelayout` eingeführt wurde
- **Beispiel:**

```
page.10 = FLUIDTEMPLATE
page.10 {
    file.stdWrap.cObject = CASE
    file.stdWrap.cObject {
        key.data = pagelayout
        default = TEXT
        default.value = EXT:sitepackage/Resources/Private/Templates/Home.html
        3 = TEXT
        3.value = EXT:sitepackage/Resources/Private/Templates/1-col.html
        4 = TEXT
        4.value = EXT:sitepackage/Resources/Private/Templates/2-col.html
    }
}
```

(Fortsetzung auf nächster Seite)

Handling von Backend-Layouts vereinfacht (2)

- pagelayout ersetzt dabei den folgenden Code:

```
field = backend_layout
ifEmpty.data = levelfield:-2,backend_layout_next_level,slide
ifEmpty.ifEmpty = default
```


Tsconfig & TypoScript

Diverse

- Für die mit TYPO3 CMS 7.4 eingeführte `stdWrap`-Funktion `bytes` kann nun die Basis (z.B. 1000 oder 1024) gesetzt werden:
`bytes.base = 1000`

Kapitel 3: Änderungen im System

Änderungen im System

Fluid-basierte Inhaltselemente (1)

- Es wurde eine Alternative zur Extension *CSS Styled Content* geschaffen: **"Fluid-based Content Elements"**
- Hier werden anstelle von TypoScript Fluid-Templates für das Rendering von Inhalten verwendet
- Dazu müssen die folgenden beiden static-Templates eingebunden werden:
 - Content Elements (`fluid_styled_content`)
 - Content Elements CSS (optional) (`fluid_styled_content`)

Änderungen im System

Fluid-basierte Inhaltselemente (2)

- Zusätzlich muss das PageTSconfig Template Fluid-based Content Elements `fluid_styled_content` in den Seiteneigenschaften eingebunden werden, damit der New-Content-Element Wizard entsprechend angepasst wird
- Eigene Fluid-Templates können wie folgt festgelegt werden:

```
lib.fluidContent.templateRootPaths.50 = EXT:site_example/Resources/Private/Templates/  
lib.fluidContent.partialRootPaths.50 = EXT:site_example/Resources/Private/Partials/  
lib.fluidContent.layoutRootPaths.50 = EXT:site_example/Resources/Private/Layouts/
```

Änderungen im System

Fluid-basierte Inhaltselemente (3)

- Um eine Installation auf die neue Struktur zu migrieren, kann man wie folgt vorgehen:
 - Deinstallieren der Extension `css_styled_content`
 - Installieren der Extension `fluid_styled_content`
 - Nun ist ein "Upgrade Wizard" im Install Tool verfügbar, der die Migration der Inhaltselemente `text`, `image` und `textpic` in `textmedia`, durchführt

Änderungen im System

SELECTmmQuery Methode für Datenbank-Zugang

- Bislang enthielt die Datenbank-Klasse die Methode `exec_SELECT_mm_query`, die die Datenbank-Abfrage direkt ausführte
- Nun wurde die Generierung des Queries (*Query-Building*) und Ausführung getrennt, indem die Methode `SELECT_mm_query` hinzugefügt wurde

```
$query = SELECT_mm_query('*','table1','table1_table2_mm','table2','AND table1.uid = 1',  
'', 'table1.title DESC');
```

Änderungen im System

Scheduler Task zur Datenbank-Optimierung

- Es wurde ein Scheduler Task implementiert, der die Datenbank via MySQL-Kommando `OPTIMIZE TABLE` optimiert
- Optimiert werden können lediglich Tabellen vom Typ MyISAM, InnoDB und ARCHIVE
- DBAL wird nicht unterstützt

Änderungen im System

Online Medien Unterstützung (1)

- Der Core wurde um eine externe Medien-Unterstützung erweitert (exemplarisch für YouTube- und Vimeo-Videos)
- Diese kann (z.B. im Inhaltselement "**Text & Media**") als URL eingegeben werden. Anschließend wird die Resource wie eine interne Datei integriert.

Änderungen im System

Online Medien Unterstützung (2)

Folgende YouTube/Vimeo URLs sind möglich:

`youtu.be/<code>`

`www.youtube.com/watch?v=<code>`

`www.youtube.com/v/<code>`

`www.youtube-nocookie.com/v/<code>`

`www.youtube.com/embed/<code>`

`vimeo.com/<code>`

`player.vimeo.com/video/<code>`

Änderungen im System

Online Medien Unterstützung (3)

- Der Zugriff per Fluid kann z.B. wie folgt durchgeführt werden:

```
<!-- enable js api and set no-cookie support for YouTube videos -->  
<f:media file="{file}" additionalConfig="{enablejsapi:1, 'no-cookie': true}" ></f:media>  
  
<!-- show title and uploader for YouTube and Vimeo before video starts playing -->  
<f:media file="{file}" additionalConfig="{showinfo:1}" ></f:media>
```

- Für YouTube existieren folgende Optionen:
autoplay, controls, loop, enablejsapi, showinfo, no-cookie
- Für Vimeo existieren folgende Optionen:
autoplay, loop, showinfo

Änderungen im System

Online Medien Unterstützung (4)

- Für einen eigenen Media-Service benötigt man eine `OnlineMediaHelper` Klasse, welche das `OnlineMediaHelperInterface` implementiert, sowie eine `FileRenderer` Klasse, die das `FileRendererInterface` implementiert

```
// Registrierung eines eigenen Online-Video-Services
$GLOBALS['TYPO3_CONF_VARS']['SYS']['OnlineMediaHelpers']['myvideo'] =
    \MyCompany\Myextension\Helpers\MyVideoHelper::class;

$rendererRegistry = \TYPO3\CMS\Core\Resource\Rendering\RendererRegistry::getInstance();
$rendererRegistry->registerRendererClass(
    \MyCompany\Myextension\Rendering\MyVideoRenderer::class
);

// Registrierung eines eigenen Mime-Types
$GLOBALS['TYPO3_CONF_VARS']['SYS']['FileInfo']['fileExtensionToMimeType']['myvideo'] =
    'video/myvideo';

// Registrierung einer eigenen Datei-Extension
$GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] .= ',myvideo';
```

Änderungen im System

Backend Routing

- Es wurde eine neue Routing Komponente zum TYPO3-Backend hinzugefügt, welche verschiedene Aufrufe handhaben kann (z.B. `http://www.example.com/typo3/document/edit`)
- Die Routen werden in folgender Datei definiert:

`Configuration/Backend/Routes.php`

```
return [  
    'myRouteIdentifier' => [  
        'path' => '/document/edit',  
        'controller' => Acme\MyExtension\Controller\MyExampleController::class . '::methodToCall'  
    ]  
];
```

- Die Methode erhält das Response- und Request-Objekt:

```
public function methodToCall(ServerRequestInterface $request, ResponseInterface $response) {  
    ...  
}
```

Änderungen im System

Autoload Definition in `ext_emconf.php`

- Zusätzlich zur Datei `composer.json` können nun Autoload-Definitionen in der Datei `ext_emconf.php` hinterlegt werden
- Das hat den Vorteil, dass nicht die gesamte Extension nach Klassen gescannt wird

```
$EM_CONF[$_EXTKEY] = array (  
    'title' => 'Extension Skeleton for TYPO3 CMS 7',  
    ...  
    'autoload' =>  
        array(  
            'psr-4' => array(  
                'Helhum\\ExtScaffold\\' => 'Classes'  
            )  
        )  
    );
```

Änderungen im System

Neue Icon-Factory (1)

- Die Logik, um mit Icons, Größen und Overlays zu arbeiten, wurde in die neue IconFactory ausgelagert
- Es gibt drei "IconProvider": BitmapIconProvider, FontawesomeIconProvider und SvgIconProvider
- Die Registrierung eines Icons erfolgt folgendermaßen:

```
IconRegistry::registerIcon($identifier, $iconProviderClassName, array $options = array());
```

Änderungen im System

Neue Icon-Factory (2)

■ Anwendung:

```
$iconFactory = GeneralUtility::makeInstance(IconFactory::class);  
$iconFactory->getIcon(  
    $identifier,  
    Icon::SIZE_SMALL,  
    $overlay,  
    IconState::cast(IconState::STATE_DEFAULT)  
)->render();
```

- Zulässige Werte für `Icon::SIZE_...` sind:
SIZE_SMALL, SIZE_DEFAULT und SIZE_LARGE

- Zulässige Werte für `Icon::STATE_...` sind:
STATE_DEFAULT und STATE_DISABLED

Änderungen im System

Neue Icon-Factory (3)

- Der Core stellt einen eigenen ViewHelper zur Verfügung, um Icons anzuzeigen:

```
{namespace core = TYPO3\CMS\Core\ViewHelpers}

<core:icon identifier="my-icon-identifier"></core:icon>

<!-- use the "small" size if none given -->
<core:icon identifier="my-icon-identifier"></core:icon>
<core:icon identifier="my-icon-identifier" size="large"></core:icon>
<core:icon identifier="my-icon-identifier" overlay="overlay-identifier"></core:icon>

<core:icon identifier="my-icon-identifier" size="default" overlay="overlay-identifier">
</core:icon>

<core:icon identifier="my-icon-identifier" size="large" overlay="overlay-identifier">
</core:icon>
```


Änderungen im System

Hooks und Signals (1)

- Es wurde ein Signal im LinkValidator zugefügt, welches die zusätzliche Verarbeitung eines Eintrages möglich macht (z.B. um Daten aus der Plugin-Konfiguration zu ermitteln o.ä.).
- Der Hook kann wie folgt in der Datei `ext_localconf.php` registriert werden:

```
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class  
);  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Linkvalidator\LinkAnalyzer::class,  
    'beforeAnalyzeRecord',  
    \Vendor\Package\Slots\RecordAnalyzerSlot::class,  
    'beforeAnalyzeRecord'  
);
```

Änderungen im System

JumpUrl als System-Extension (1)

- Die Erzeugung und das Handling von JumpURLs wurde aus der Frontend-Extension entfernt und zur neuen System-Extension `jumpurl` verschoben
- Hook zur Manipulation von **URLs** in der Datei `ext_localconf.php`:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlHandlers']  
    ['myext_myidentifier']['handler'] = \Company\MyExt\MyUrlHandler::class;  
  
// Die Klasse muss das UrlHandlerInterface implementieren  
class MyUrlHandler implements \TYPO3\CMS\Frontend\Http\UrlHandlerInterface {  
    ...  
}
```

BREAKING CHANGE!

Änderungen im System

JumpUrl als System-Extension (2)

- Handling von **Links** in der Datei `ext_localconf.php`:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlProcessors']  
    ['myext_myidentifizier']['processor'] = \Company\MyExt\MyUrlProcessor::class;  
  
// Die Klasse muss das UrlProcessorInterface implementieren  
class MyUrlProcessor implements \TYPO3\CMS\Frontend\Http\UrlProcessorInterface {  
    ...  
}
```

BREAKING CHANGE!

Änderungen im System

Kommandozeilenaufruf (CLI)

- Sollte es beim Aufruf von `typo3/cli_dispatch.phpsh` zu Fehlern kommen, so werden diese farbig dargestellt
- CommandController können nun auch in Unterordnern liegen
- Beispiel:

Controller in Datei:

```
my_ext/Classes/Command/Hello/WorldCommandController.php
```

...kann im CLI wie folgt aufgerufen werden:

```
typo3/cli_dispatch.sh extbase my_ext:hello:world <arguments>
```

Änderungen im System

Diverse Änderungen (1)

- Die Verschieben-Buttons beim TCA-Type group können nun mit der TCA-Option `hideMoveIcons = TRUE` deaktiviert werden
- Die Funktion `makeCategorizable()` kann nun überschrieben werden, sofern diese vorher bereits aufgerufen wurde (z.B. für `tt_content`).
- Beispiel:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable(
    'css_styled_content', 'tt_content', 'categories', array(), TRUE
);
```

Der letzte Parameter steuert das Überschreiben (hier: `TRUE`).
Standardmäßig ist der Wert `FALSE`.

Änderungen im System

Diverse Änderungen (2)

- Es gibt nun eine Funktion, um eine Unique-ID zu erzeugen

```
$uniqueId = \TYPO3\CMS\Core\Utility\StringUtility::getUniqueId('Prefix');
```

- Als Plaintext Dateieindung wurde `typoscript` hinzugefügt

- Es gibt nun eine neue Konfigurations-Option, die regelt, welche Dateieindungen als Media-Dateien interpretiert werden:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] =  
    'gif,jpg,jpeg,bmp,png,pdf,svg,ai,mov,avi';
```

BREAKING CHANGE!

Kapitel 4: Extbase & Fluid

Severity-Filter für FlashMessages

- Bislang konnten nur alle FlashMessages auf einmal ermittelt und/oder gelöscht werden
- Nun kann man diese entsprechend der Severity (Gewichtung) filtern

```
FlashMessageQueue::getAllMessages($severity);  
FlashMessageQueue::getAllMessagesAndFlush($severity);  
FlashMessageQueue::removeAllFlashMessagesFromSession($severity);  
FlashMessageQueue::clear($severity);
```


Extbase & Fluid

Query-Support für `between` hinzugefügt

- Es wurde `between` zum Extbase Query Objekt hinzugefügt, welches prüft, ob sich ein Wert innerhalb einer oberen und unteren Grenze (einschließlich) befindet
- Dies wird zu `(min <= expr AND expr <= max)` übersetzt. Dadurch hat dies keine Performance-Auswirkungen und funktioniert auf jedem DBMS

```
$query->matching(  
    $query->between('uid', 3, 5)  
);
```

Mehrere Message-Queues

- Es können nun mehrere Message-Queues in Extbase realisiert werden:

```
$queueIdentifier = 'myQueue';  
$this->controllerContext->getFlashMessageQueue($queueIdentifier);
```

- In Fluid kann wie folgt darauf zugegriffen werden:

```
<f:flashMessages queueIdentifier="myQueue" />
```

Media-ViewHelper (1)

- Um Medien komfortabel im Frontend rendern zu können (z.B. Video, Audio, registrierte Renderer), wurde ein MediaViewHelper zugefügt
- Zuerst versucht der ViewHelper den Renderer aufzurufen; schlägt dies fehl, wird ein Image-Tag gerendert
- Beispiel:

```
<code title="Image Object">  
  <f:media file="{file}" width="400" height="375" />  
</code>
```

```
<output>  
    
</output>
```

Media-ViewHelper (2)

■ Beispiel (Fortsetzung):

```
<code title="MP4 Video Object">
  <f:media file="{file}" width="400" height="375" />
</code>
```

```
<output>
  <video width="400" height="375" controls>
    <source src="fileadmin/user_upload/my-video.mp4" type="video/mp4">
  </video>
</output>
```

```
<code title="MP4 Video Object with loop and autoplay option set">
  <f:media file="{file}" width="400" height="375"
    additionalConfig="{loop: '1', autoplay: '1'}" />
</code>
```

```
<output>
  <video width="400" height="375" controls loop>
    <source src="fileadmin/user_upload/my-video.mp4" type="video/mp4">
  </video>
</output>
```

Extbase & Fluid

System-Extension form (1)

- Die System-Extension form (inkl. Daten-Model, Controller-Logig, Property Validation, Views und Templating) wurde so adaptiert, dass der Extbase/Fluid MVC Stack unterstützt wird
- Die Ausgabe basiert nun komplett auf Fluid und kann somit entsprechend angepasst werden. Pro Form-Element gibt es ein eigenes Partial, welches nun auch individuell über die TypoScript-Option `partialPath = ...` angepasst werden kann
- Es wurden drei neue ViewHelper implementiert:
 - `AggregateSelectOptionsViewHelper` (für optgroup Tags)
 - `SelectViewHelper` (für von optgroup Tags)
 - `PlainMailViewHelper` (zum Rendern von Plaintext Mails)

System-Extension `form` (2)

- Außerdem gibt drei Views:
 - `show` (das Formular selbst)
 - `confirmation` (die Bestätigungsseite)
 - `postProcessor/mail` (die Email)
- Die Template-Pfade und Sichtbarkeiten der Felder können für jeden View individuell angepasst werden

Extbase & Fluid

Annotation `@cli`

- Eine neue Annotation `@cli` wurde eingeführt:
wird diese beim `CommandController` verwendet, so kann dieser nur auf der Kommandozeile, aber nicht im Scheduler verwendet werden

Kapitel 5:

Veraltete und entfernte Funktionen

Veraltete/Entfernte Funktionen

Slash-Methoden in GeneralUtility

- Innerhalb der Klasse GeneralUtility wurden folgende Methoden als **deprecated** deklariert:
 - GeneralUtility::addSlashesOnArray()
 - GeneralUtility::stripSlashesOnArray()
 - GeneralUtility::slashArray()

Veraltete/Entfernte Funktionen

CLI Konstanten und Methoden

- Die Logik hinsichtlich Optionen bei CLI-basierten Skripts wurde an den `CliRequestHandler` übergeben
- Daher gilt folgende Methode als **deprecated**:

```
BackendUserAuthentication->checkCLIuser()
```

- Folgende Konstanten und globale Parameter sind nun ebenfalls **deprecated**:

```
const TYPO3_cliKey  
const TYPO3_cliInclude  
$GLOBALS['MCONF']['name']  
$GLOBALS['temp_cliScriptPath']  
$GLOBALS['temp_cliKey']
```

Veraltete/Entfernte Funktionen

IconUtility

- Die Klasse `IconUtility` gilt ab sofort als **deprecated**. Einige der Methoden wurden zur `IconFactory` verschoben:

```
IconUtility::skinImg()  
IconUtility::getIcon()  
IconUtility::getSpriteIcon()  
IconUtility::getSpriteIconForFile()  
IconUtility::getSpriteIconForRecord()  
IconUtility::getSpriteIconForResource()  
IconUtility::getSpriteIconClasses()
```

- Ebenso wurde folgender `PageTSconfig`-Schlüssel als **deprecated** markiert:

```
mod.wizards.newContentElement.wizardItems.*.elements.*.icon
```

Veraltete/Entfernte Funktionen

Veraltete `HtmlParser`-Methoden

- Die Marker-Ersetzungsfunktionalität wurde verschoben von `core/Classes/Html/HtmlParser.php` zur eigenen Klasse `core/Classes/Service/MarkerBasedTemplateService.php`
- Daher sind folgende Methoden **deprecated** und werden in TYPO3 CMS Version 8 entfernt:

`HtmlParser::getSubpart()`

`HtmlParser::substituteSubpart()`

`HtmlParser::substituteSubpartArray()`

`HtmlParser::substituteMarker()`

`HtmlParser::substituteMarkerArray()`

`HtmlParser::substituteMarkerAndSubpartArrayRecursive()`

Veraltete/Entfernte Funktionen

Änderung in Form Extension

- Die System-Extension `form` basiert nun auf `Extbase/Fluid`
- Daher ist folgender TypoScript-Code veraltet, da man die Eigenschaft `layout` nicht mehr verwenden sollte:

```
10 = FORM
10 {
    layout {
        containerWrap = <div><elements /></div>
        elementWrap = <div><element /></div>
    }
}
```

Veraltete/Entfernte Funktionen

Veraltete ViewHelper und Methoden

- Folgende ViewHelper wurden als **deprecated** markiert und sollten daher nicht mehr verwendet werden:

```
\TYPO3\CMS\Fluid\ViewHelpers\Be\Buttons\IconViewHelper  
\TYPO3\CMS\Backend\ViewHelpers\SpriteManagerIconViewHelper
```

- Die folgenden Methoden wurden als **deprecated** markiert und sollten daher nicht mehr verwendet werden:

```
BackendUtility::getExcludeFields()  
BackendUtility::getExplicitAuthFieldValues()  
BackendUtility::getSystemLanguages()  
BackendUtility::getRegisteredFlexForms()  
BackendUtility::exec_foreign_table_where_query()  
BackendUtility::replaceMarkersInWhereClause()
```

Kapitel 6: Quellen und Autoren

Quellen und Autoren

Quellennachweis

TYPO3 News:

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

Release Infos:

- http://wiki.typo3.org/TYPO3\CMS_7.5.0
- [INSTALL.md](#) and [Changelog](#)
- [typo3/sysext/core/Documentation/Changelog/7.5/*](http://typo3.org/typo3/sysext/core/Documentation/Changelog/7.5/*)

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>

Quellen und Autoren

pluswerk



pluswerk ist eine Full Service Agentur für leidenschaftliche digitale Kommunikation mit 10 Standorten in Deutschland und über 130 Mitarbeitern

www.pluswerk.ag | +49 89 130 145 20 | facebook.com/pluswerk