

TYP03 CMS 7 LTS – What's New

Extbase & Fluid

Patrick Lobacher und Michael Schams

Introduction

TYPO3 CMS 7 LTS - What's New

Die folgenden Slides sind auf ein spezielles Thema fokussiert. Abhängig von der Rolle, können die folgenden Themen ebenfalls interessant sein:

	<i>BE User Interface</i>	<i>TypoScript</i>	<i>In-Depth Changes</i>	<i>Extbase/Fluid</i>	<i>Deprecated/Removed</i>	<i>Sys.Administration</i>
Redakteure	X					
Integratoren		X	X		X	
Entwickler			X	X	X	
SysAdmins						X

Download aller **What's New Slides** unter typo3.org

Extbase & Fluid

Extbase und Fluid sind der de facto Standard wenn es um moderne Technologien im TYPO3 Universum geht. Auch wenn Extbase einige wenige neue Features spendiert bekommen hat, wurden in TYPO3 CMS 7 LTS die meisten Änderungen im Bereich Fluid vorgenommen.

Einige ViewHelpers wurden aktualisiert, verbessert oder gar neue eingeführt. So gibt es zum Beispiel nun einen **MediaViewHelper**, der dafür verwendet werden kann, um Video, Audio oder andere Media-Dateien im Frontend zu rendern.

Zusätzlich wurde die System Extension `form` auf Extbase & Fluid portiert.

Template Path Fallback

- Sowohl Fluid Standalone View wie auch das TypoScript-Objekt FLUIDTEMPLATE unterstützen nun Fallback Pfade

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:myextension/Resources/Private/Templates/Main.html
page.10.partialRootPaths {
    10 = EXT:myextension/Resources/Private/Partials
    20 = EXT:fallback/Resources/Private/Partials
}
```

- Verwendet man die alten Optionen (z.B. partialRootPath) zusammen mit den neuen, so wird die alte auf die erste Position in der Fallback-Liste gestellt (Index = 0)

Extbase & Fluid

Typolink ViewHelper

- Es wurde ein Typolink-ViewHelper zugefügt, der beispielsweise Felder auswerten und darstellen kann, die über einen Link-Wizard im Backend erzeugt wurden

```
<f:link.typolink parameter="{link}" target="_blank" class="ico-class" title="some title"
    additionalAttributes="{type: 'button'}">
```

`link` enthält beispielsweise:

```
42 _blank - "This is the link title" &foo=bar
```

Ausgabe:

```
<a href="index.php?id=42&foo=bar" title="This is the title" target="_blank" class="ico-class"
    type="button">
```

Lediglich `parameter` wird benötigt, der Rest ist optional

Extbase & Fluid

Allgemeine data-* Attribut

- Alle ViewHelper, die HTML Tags ausgeben, unterstützen nun das HTML5 data-* Attribut
- Array, welches als data übergeben wird, wird ausgewertet und dessen Schlüssel-/Werte-Paar ergeben das Attribut: data-key="value"

Beispiel:

```
<f:form.textfield data="{foo: 'bar', baz: 'foos'}" />
```

Ausgabe:

```
<input data-foo="bar" data-baz="foos" ... />
```

Class Tag Values via Reflection

- Extbase Reflection-Service kann nun Tags bzw. Annotations zurückliefern, die zu einer Klasse zugefügt wurden

Beispiel:

```
/**
 * @SomeClassAnnotation A value
 */
class MyClass {
}
```

Die Annotation kann dann wie folgt ermittelt werden:

```
$service = new \TYPO3\CMS\Extbase\Reflection\ReflectionService();

// Returns all tags and their values the specified class is tagged with
$classValues = $service->getClassTagsValues('MyClass');

// Returns the values of the specified class tag
$classValue = $service->getClassTagValue('MyClass', 'SomeClassAnnotation');
```

PaginateViewHelper

- Der Paginate-ViewHelper unterstützt nun folgende Input-Werte:

- QueryResultInterface
- ObjectStorage
- ArrayAccess
- array

- Beispiel:

```
<f:widget.paginate objects="{blogs}" as="paginatedBlogs">
  <f:for each="{paginatedBlogs}" as="blog">
    <h4>{blog.title}</h4>
  </f:for>
</f:widget.paginate>
```


ContainerViewHelper lädt RequireJS Module

- Der ContainerViewHelper kann RequireJS-Module via `includeRequireJsModules` Attribut laden
- Beispiel:

```
<f:be.container pageTitle="Extension Module" loadJQuery="true"
  includeRequireJsModules="{
    0: 'TYPO3\CMS\Extension\Module1',
    1: 'TYPO3\CMS\Extension\Module2',
    2: 'TYPO3\CMS\Extension\Module3',
    3: 'TYPO3\CMS\Extension\Module4'
  }" >
```

Methode `has()` im `ObjectAccess`

- Für die Benutzung in Fluid, `object.property` und `object.isProperty` unterstützten bereits die folgenden Methoden:
 - `isProperty()`
 - `getProperty()`
- Neu in TYPO3 CMS 7.1: `hasProperty()`
- Hier wird die Method `$object->hasProperty()` aufgerufen, wenn `object.hasProperty` in Fluid benutzt wird

Hochladen mehrerer Dateien im FormUpload-ViewHelper

- Der FormUpload-Viewhelper unterstützt das neue Attribut `multiple`, welches es ermöglicht, mehrere Dateien auf einmal zu übertragen

```
<f:form.upload property="files" multiple="multiple" />
```

Hinweis: es ist darauf zu achten, dass für das Property-Mapping ein eigener `TypeConverter` erstellt werden muss!

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.

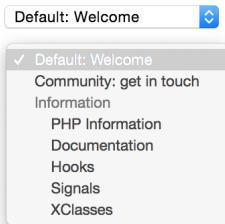
ActionMenuItemGroupViewHelper (1)

- Mit diesem neuen ViewHelper können im Backend Optionen-Gruppen für Select-Felder verwendet werden
- Beispiel:

```
<f:be.menus.actionMenu>
  <f:be.menus.actionMenuItem label="Default: Welcome" controller="Default" action="index" />
  <f:be.menus.actionMenuItem label="Community: get in touch" controller="Community"
    action="index" />
  <f:be.menus.actionMenuItemGroup label="Information">
    <f:be.menus.actionMenuItem label="PHP Information" controller="Information"
      action="listPhpInfo" />
    <f:be.menus.actionMenuItem label="Documentation" controller="Information"
      action="documentation" />
    <f:be.menus.actionMenuItem label="Hooks" controller="Information" action="hooks" />
    <f:be.menus.actionMenuItem label="Signals" controller="Information" action="signals" />
    <f:be.menus.actionMenuItem label="XClasses" controller="Information" action="xclass" />
  </f:be.menus.actionMenuItemGroup>
</f:be.menus.actionMenu>
```

ActionMenuItemGroupViewHelper (2)

- Beispiel auf der vorherigen Slide resultiert in folgender Ausgabe:



Template-Support für FlashMessagesViewHelper

- Der FlashMessagesViewHelper hat nun Template Unterstützung
- Mit dem neuen Attribut `as` kann man eine Variable festlegen, über der der Zugriff auf die Messages möglich ist
- Beispiel:

```
<f:flashMessages as="flashMessages">
  <ul class="myFlashMessages">
    <f:for each="{flashMessages}" as="flashMessage">
      <li class="alert {flashMessage.class}">
        <h4>{flashMessage.title}</h4>
        <span class="fancy-icon">{flashMessage.message}</span>
      </li>
    </f:for>
  </ul>
</f:flashMessages>
```

- Die Option `renderMode` ist ab sofort "deprecated"!

Neue Eigenschaften im cObject FLUIDTEMPLATE (1)

- Das cObject FLUIDTEMPLATE wurde um die Eigenschaften `templateRootPaths` und `templateName` ergänzt
- Man kann nun einen Template-Namen setzen, der zusammen mit dem Format im angegebenen Template-Pfad gesucht wird
- `templateRootPaths` hat die selbe Fallback-Logik wie `layoutRootPath` und `partialRootPath`
 - `templateName` (string/stdWrap)
 - `templateRootPaths` (Array mit Datei-Pfaden und "EXT:" Unterstützung)

Neue Eigenschaften im cObject FLUIDTEMPLATE (2)

■ TypeScript Beispiel:

```
lib.stdContent = FLUIDTEMPLATE
lib.stdContent {
    templateName = TEXT
    templateName.stdWrap {
        cObject = TEXT
        cObject {
            data = levelfield:-2,backend_layout_next_level,slide
            override.field = backend_layout
            split {
                token = frontend__
                1.current = 1
                1.wrap = |
            }
        }
        ifEmpty = Default
    }
    templateRootPaths {
        10 = EXT:frontend/Resources/Private/Templates
        20 = EXT:sitemodification/Resources/Private/Templates
    }
}
```

Entfernung von `xmlns`-Attributes und des HTML-Tags (1)

- Durch die Einführung von `xmlns:*`-Attributen in Tags ist es einer IDE möglich Fluid direkt zu unterstützen (z.B. Syntax-Highlighting, Autovervollständigung, usw.).
Diese Attribute werden allerdings auch ausgegeben.
- Der Workaround, den eigentlichen Inhalt in Sections auszulagern, ist in Layouts nicht möglich und zudem nicht intuitiv
- Daher wird ein Namespace, sofern dieser den Aufbau `http://typo3.org/ns/<phpNamespace>` hat, automatisch entfernt (alle anderen Namespaces bleiben erhalten)

Entfernung von xmlns-Attributes und des HTML-Tags (2)

- Verwendet man im HTML-Tag das Attribut `data-namespace-typo3-fluid="true"`, wird das gesamte HTML-Tag (öffnendes und schließendes) nicht gerendert

```
<html data-namespace-typo3-fluid="true"
  xmlns:f="http://typo3.org/ns/TYPO3/CMS/Fluid/ViewHelpers"
  xmlns:n="http://typo3.org/ns/GeorgRinger/News/ViewHelpers">

  <f:if condition="{newsItem.title}">
    <f:then>
      <n:titleTag>{newsItem.title}</n:titleTag>
    </f:then>
    <f:else>
      <n:titleTag>News-Detail</n:titleTag>
    </f:else>
  </f:if>

</html>
```

Neue Methoden im Fluid-StandaloneView

- Der StandaloneView wird mit den Methoden `setTemplateRootPaths($templatePaths)` und `setTemplate($templateName, $throwException = TRUE)` erweitert
- Selbe Funktionalität wie im cObject FLUIDTEMPLATE
- Beispiel (Rendern eines Email-Templates):

```
$view = GeneralUtility::makeInstance(StandaloneView::class);  
$view->setLayoutRootPaths(array(GenericUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Layouts')));  
$view->setPartialRootPaths(array(GenericUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Partials')));  
$view->setTemplateRootPaths(array(GenericUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Templates')));  
$view->setTemplate('Email/Notification');  
$emailBody = $view->render();
```

Data Processing für FLUIDTEMPLATE cObject (1)

- Das cObject FLUIDTEMPLATE wird mit der Option `dataProcessing` (enthält ein Array aus FQCN) ausgerüstet, mit dessen Hilfe man das `$data` Array manipulieren kann, welches das aktuell zu rendernde Objekt (wie `page` oder `tt_content`) enthält
- Der Prozessor muss das Interface `FluidTemplateDataProcessorInterface` implementieren und folgende Methode enthalten:

```
function process(array &$data, array $processorConfiguration,  
    array $configuration, StandaloneView $view) {  
    [...]  
}
```

Data Processing für FLUIDTEMPLATE cObject (2)

■ Beispiel:

```
my_custom_ctype = FLUIDTEMPLATE
my_custom_ctype {
    templateRootPaths {
        10 = EXT:your_extension_key/Resources/Private/Templates
    }
    templateName = CustomName
    settings {
        extraParam = 1
    }
    dataProcessing {
        1 = Vendor\YourExtensionKey\DataProcessing\MyFirstCustomProcessor
        2 = AnotherVendor\AnotherExtensionKey\DataProcessing\MySecondCustomProcessor
        2 {
            options {
                myOption = SomeValue
            }
        }
    }
}
```


Section-Anker für Pagination Widget

- Es ist nun möglich einen Section-Anker im Pagination Widget zu verwenden
- Dazu gibt es den Schlüssel `section` im Attribut `configuration`
- Im folgenden Beispiel wird der Anker `#archive` an jeden Widget-Link angehängt:

```
<f:widget.paginate objects="{plantpestWarnings}" as="paginatedWarnings"  
  configuration="{section: 'archive', itemsPerPage: 10, insertAbove: 0, insertBelow: 1,  
  maximumNumberOfLinks: 10}">
```

[...]

```
</f:widget.paginate>
```

Attribut `base` für `Date-ViewHelper`

- Der `Date-ViewHelper` wurde um das optionale Attribut `base` ergänzt
- Damit kann man relative Berechnungen durchführen
- Wird das Datum als `DateTime` angegeben, wird `base` ignoriert
- Erlaubte Werte: siehe [PHP Dokumentation](#)
- Das folgende Beispiel gibt "2016" zurück, wenn das Objekt "dateObject" ein beliebiges Datum in 2017 enthält:

```
<f:format.date format="Y" base="{dateObject}">-1 year</f:format.date>
```

dataProcessing bei FLUIDTEMPLATE

- Mit TYPO3 CMS 7.3 wurde die Option dataProcessing beim Content-Objekt FLUIDTEMPLATE eingeführt
- Hierfür ändert sich das zu implementierende Interface von FluidTemplateDataProcessorInterface in DataProcessorInterface und damit auch die Methode process()

```
public function process(  
    ContentObjectRenderer $cObj,  
    array $contentObjectConfiguration,  
    array $processorConfiguration,  
    array $processedData  
);
```

BREAKING CHANGE!

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

Relationen zu gleichen Tabellen

- Es nun möglich ein Domain Model zu verwenden, in der ein Objekt eine Relation zu einem anderen Objekt der selben Klasse hat

```
namespace \Vendor\Extension\Domain\Model;
class A {
    /**
     * @var \Vendor\Extension\Domain\Model\A
     */
    protected $parent;
}
```

```
namespace \Vendor\Extension\Domain\Model;
class A {
    /**
     * @var \Vendor\Extension\Domain\Model\B
     */
    protected $x;

    /**
     * @var \Vendor\Extension\Domain\Model\B
     */
    protected $y;
}
```

absolute Option für Image-ViewHelper

- Der ImageViewHelper und Uri/ImageViewHelper haben nun eine Option `absolute`, mit der eine absolute URL ausgegeben werden kann.
- Beispiel 1 (ImageViewHelper):

```
<f:image image="{file}" width="400" height="375" absolute="1" ></f:image>

// Ausgabe

```

- Beispiel 2 (Uri/ImageViewHelper):

```
<f:uri.image image="{file}" width="400" height="375" absolute="1" ></f:uri>

// Ausgabe
http://www.mydomain.com/fileadmin/_processed_/323223424.png
```

ViewHelper um Whitespaces in HTML zu entfernen

- Es wurde ein ViewHelper `spaceless` eingeführt, um überflüssige Leerzeichen zwischen HTML-Tages zu entfernen. Beispiel:

```
<f:spaceless>
<div>
  <div>
    <div>text

text</div>
</div>
</div>
```

- Ausgabe:

```
<div><div><div>text

text</div></div></div>
```

RootLevel Konfiguration

- Der RootLevel einer Tabelle kann nun im TCA konfiguriert werden (damit wird festgelegt, wo die zugehörigen Datensätze gesucht werden)
 - 0: Nur im Seitenbaum
 - 1: Nur auf der Root-Seite (PID 0)
 - -1: In beiden
- Zudem muss das TCA konfiguriert werden:

```
$GLOBALS['TCA']['tx_myext_domain_model_record']['ctrl']['rootLevel'] = -1;
```

Quellen und Autoren

Sources and Authors

Quellennachweis

TYPO3 News:

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

Release Infos:

- https://wiki.typo3.org/Category:ReleaseNotes/TYPO3_7.x
- [INSTALL.md](#) and [Changelog](#)
- [typo3/sysex/core/Documentation/Changelog/](http://typo3.org/sysex/core/Documentation/Changelog/)*

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>

Sources and Authors

TYPO3 CMS What's New Slides:

Patrick Lobacher

(Recherche, Informationsdokumentation und deutsche Version)

Michael Schams

(Project Leader und englische Version)

Übersetzungen und Mitwirkung von:

Andrey Aksenov, Paul Blondiaux, Pierrick Caillon, Sergio Catalá,
Ben van't Ende, Jigal van Hemert, Sinisa Mitrovic, Michel Mix, Angeliki Plati,
Nena Jelena Radovic und Roberto Torresani

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

Lizensiert unter Creative Commons BY-NC-SA 3.0

