TYPO3 CMS 7.3 – What's New

Übersicht der neuen Funktionen, Änderungen und Verbesserungen

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

[+]pluswerk
Digitale Leidenschaft

17/June/2015

TYPO3

TYPO3 CMS 7.3 - What's New

Kapitelübersicht

TYPO3

Einführung

(Die Fakten)

TYPO3

Einführung

TYPO3 CMS 7.3 – Die Fakten

- Veröffentlichungsdatum: 16. Juni 2015
- Releasetyp: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Hauptfokus: Package Ecosystem, Composer und Extension Handling

Einführung

Systemvoraussetzungen

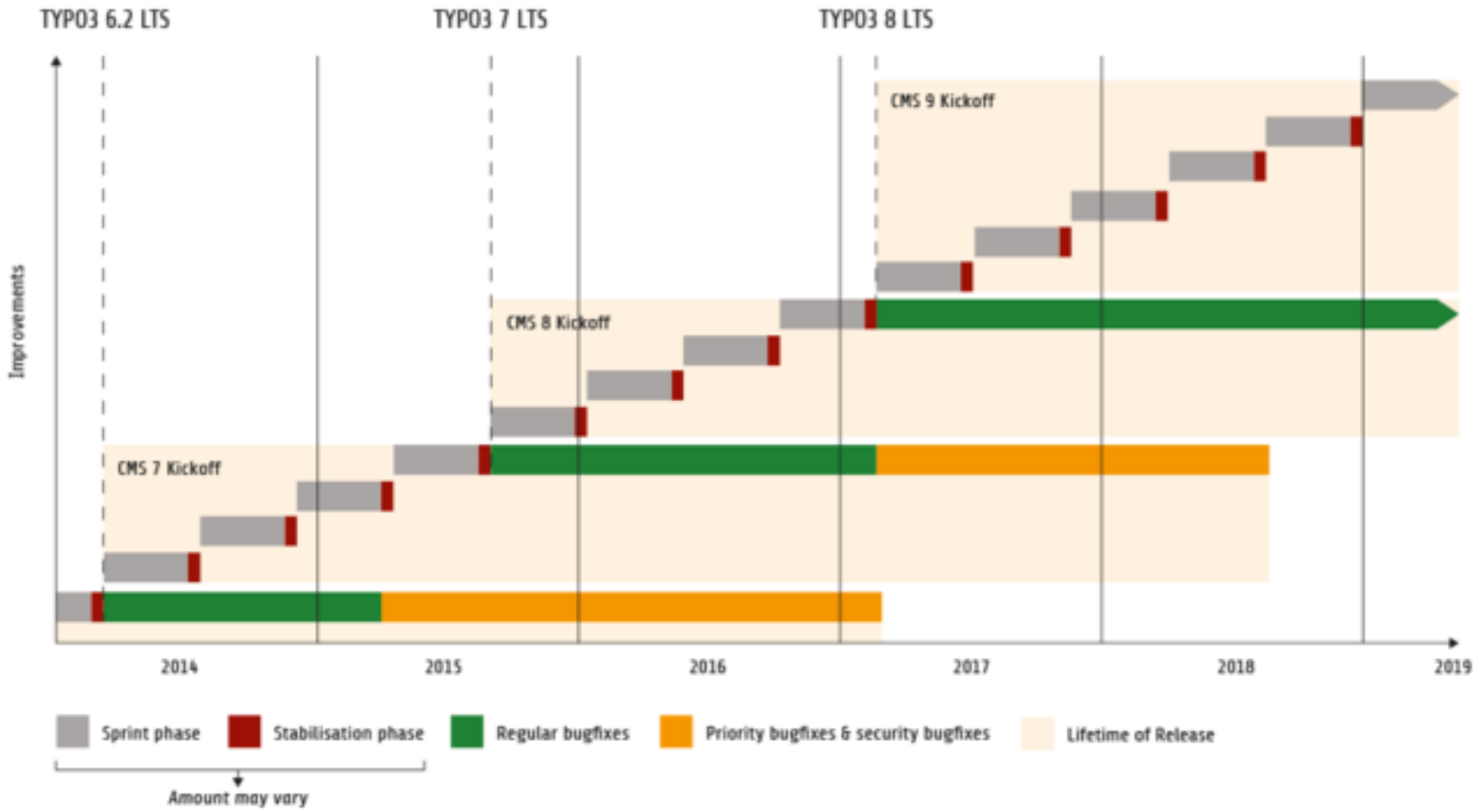
- PHP*: v5.5.0 - v5.6.x
- MySQL: v5.5.x - v5.6.x (no strict mode)
- Festplattenplatz: mindestens 200 MB
- PHP Einstellungen:
  - memory_limit >= 128M
  - max_execution_time >= 240s
  - PHP Kompilierungsoption `--disable-ipv6` darf nicht aktiviert sein
- Backend benötigt IE >= 9 oder jeden anderen modernen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

*) weitere Details: PHP Minimum Requirements for TYPO3 CMS 7

TYPO3

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 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/

TYPO3

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/7.3
$ tar xzf typo3_src-7.3.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.3.0 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 `mlink` benutzt werden

Einführung

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_7.3
- 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)

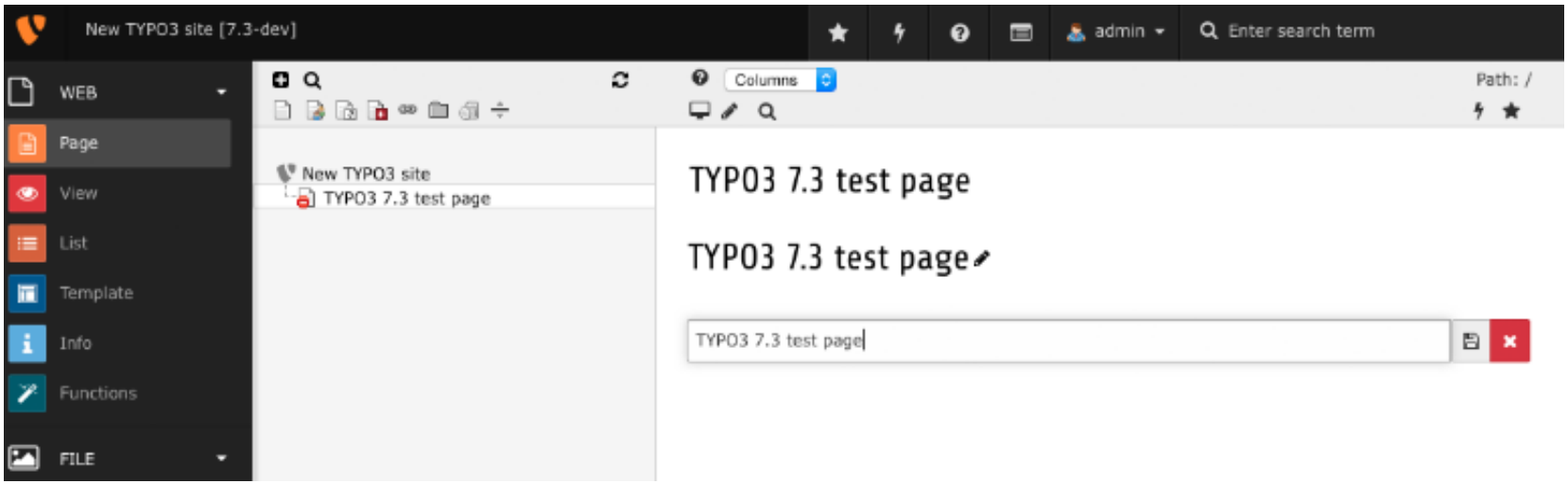
Kapitel 1:

Backend User Interface

Backend User Interface

Seitentitel im Page- und List-Modul

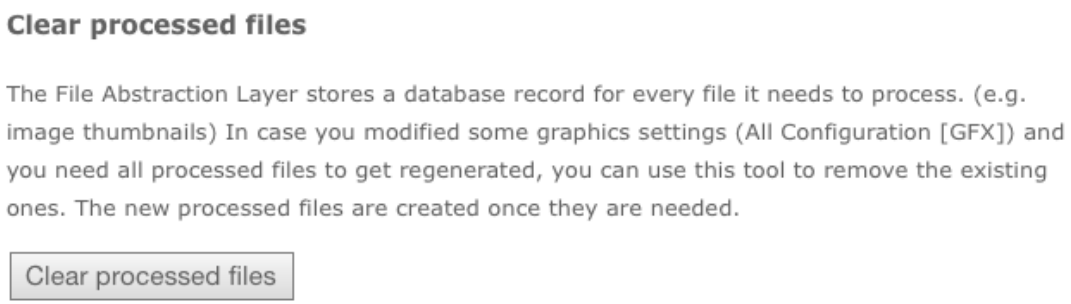
Im Page- und List-Modul kann man den Seitentitel entweder per Doppelklick oder mit Klick auf das Bearbeitungssymbol ändern.



Backend User Interface

Prozessierte FAL Dateien im Install Tool löschen

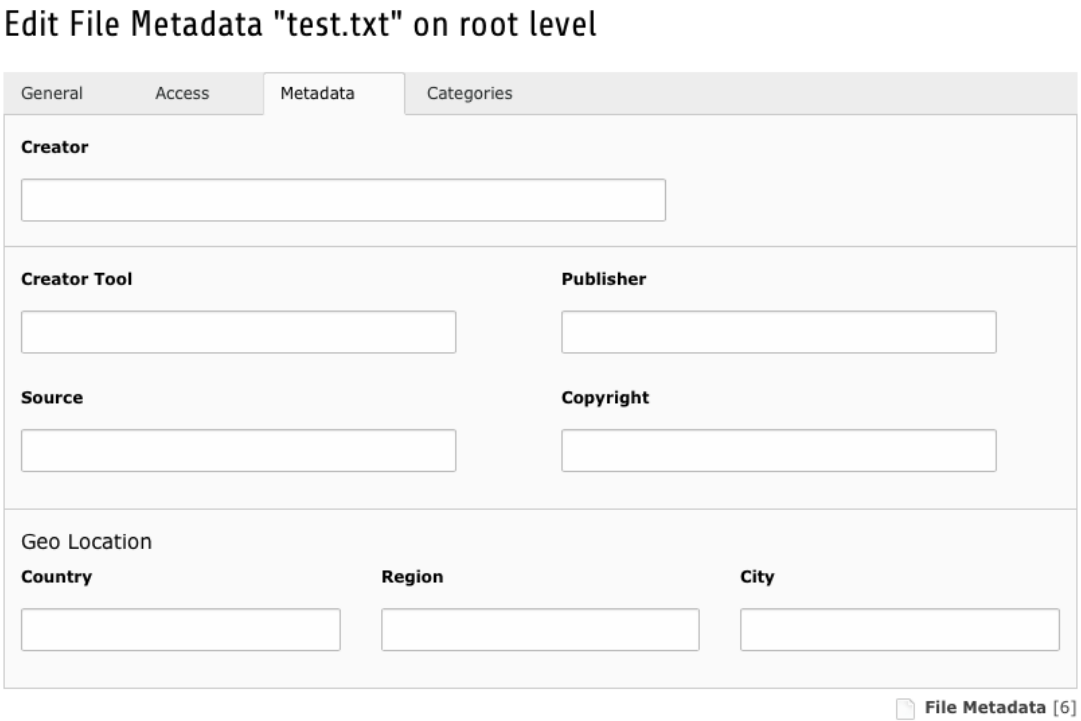
Das Install Tool enthält nun ein neues Tool (unterhalb von "Clean up"), um prozessierte FAL Dateien (wie z.B. Thumbnails) zu löschen. Das ist insbesondere hilfreich, wenn man grafik-relevante Settings ändern oder wenn man GraphicsMagick/ImageMagick aktualisiert hat und alle Dateien neu generieren will.



Backend User Interface

Copyright in FAL Meta-Daten

In den zusätzlichen FAL Meta-Daten (Extension: `filemetadata`) gibt es nun ein Feld "**Copyright**".



Kapitel 2:

TSconfig & TypoScript

TSconfig & TypoScript

stdWrap Funktion `strtotime`

- Es gibt nun eine stdWrap Funktion `strtotime`, welche es ermöglicht, formatierte Datum-Angaben in einen Timestamp umzuwandeln

```
date_as_timestamp = TEXT
date_as_timestamp {
  value = 2015-04-15
  strtotime = 1
}

next_weekday = TEXT
next_weekday {
  data = GP:selected_date
  strtotime = + 2 weekdays
  strftime = %Y-%m-%d
}
```

TSconfig & TypoScript

`GPmerged` in Conditions

- Prüft man in Conditions nur mittels GP so wird beim gleichzeitigen Vorhandensein von POST- und GET-Variablen (z.B. `tx_demo_demo[...]=...`), lediglich die POST-Variable zurückgegeben
- Mit der neuen Option `GPmerged` werden beide Variablen zusammengeführt und dann zurückgegeben

```
[globalVar = GPmerged:tx_demo|foo = 1]
  page.90 = TEXT
  page.90.value = DEMO
[global]
```

TSconfig & TypoScript

Weitere Werte für die Funktion `stdWrap.case`

- Die stdWrap Funktion `case` ist um die beiden Werte `uppercamelcase` und `lowercamelcase` ergänzt worden
- Beispiel:

```
tt_content = CASE
tt_content {
  key.field = CType
  my_custom_ctype =< lib.userContent
  my_custom_ctype {
    file = EXT:site_base/Resources/Private/Templates/SomeOtherTemplate.html
    settings.extraParam = 1
  }
  default =< lib.userContent
  default {
    file = TEXT
    file.field = CType
    file.stdWrap.case = uppercamelcase
    file.wrap = EXT:site_base/Resources/Private/Templates/|.html
  }
}
```

TSconfig & TypoScript

Eigenschaft `integrity` für JavaScript-Dateien (1)

- Es wurde die Eigenschaft `integrity` zugefügt, um einen SRI Hash zum JavaScript-Markup hinzuzufügen, mit dem die Quelle verifiziert werden kann (SRI: Sub-Resource Integrity)
- Dies betrifft die Eigenschaften `page.includeJSLibs`, `page.includeJSFooterlibs`, `includeJS` und `includeJSFooter`
- Beispiel:

```
page {
  includeJS {
    jQuery = https://code.jquery.com/jquery-1.11.3.min.js
    jquery.external = 1
    jQuery.disableCompression = 1
    jQuery.excludeFromConcatenation = 1
    jQuery.integrity = sha256-7LkWEzqTdpEfELxcZZlS6wAx5Ff13zZ83lYO2/ujj7g=
  }
}
```


TYPO3

TSconfig & TypoScript

Eigenschaft `integrity` für JavaScript-Dateien (2)

- SRI ist eine Spezifikation des W3C, die es ermöglicht zu verifizieren, ob extern gehostete Dateien manipuliert worden sind
- Erstellen von Integrity Hashes:
  - Option 1: https://srihash.org
  - Option 2: durch folgende Kommandos

```
cat FILENAME.js | openssl dgst -sha256 -binary | openssl enc -base64 -A
```

- Weitere Informationen:
  - http://www.w3.org/TR/SRI/

Kapitel 3:

Änderungen im System

Änderungen im System

Symfony/Console im CommandController (1)

Der CommandController verwendet nun intern Symfony/Console und stellt damit verschiedene Methoden zur Verfügung:

- TableHelper
  - `outputTable($rows, $headers = NULL)`
- DialogHelper
  - `select($question, $choices, $default = NULL, $multiSelect = false, $attempts = FALSE)`
  - `ask($question, $default = NULL, array $autocomplete = array())`
  - `askConfirmation($question, $default = TRUE)`
  - `askHiddenResponse($question, $fallback = TRUE)`
  - `askAndValidate($question, $validator, $attempts = FALSE, $default = NULL, array $autocomplete = NULL)`
  - `askHiddenResponseAndValidate($question, $validator, $attempts = FALSE, $fallback = TRUE)`

Änderungen im System

Symfony/Console im CommandController (2)

- ProgressHelper
  - `progressStart($max = NULL)`
  - `progressSet($current)`
  - `progressAdvance($step = 1)`
  - `progressFinish()`

(siehe Code-Beispiel auf den folgenden Slides)

Änderungen im System

Symfony/Console im CommandController (3)

```
<?php
namespace Acme\Demo\Command;
use TYPO3\CMS\Extbase\Mvc\Controller\CommandController;

class MyCommandController extends CommandController {
  public function myCommand() {

    // Table rendern
    $this->output->outputTable(array(
      array('Bob', 34, 'm'),
      array('Sally', 21, 'f'),
      array('Blake', 56, 'm')
    ),
    array('Name', 'Age', 'Gender'));

    // Selektieren
    $colors = array('red', 'blue', 'yellow');
    $selectedColorIndex = $this->output->select('Please select one color', $colors, 'red');
    $this->outputLine('You choose the color %s.', array($colors[$selectedColorIndex]));

    [...]
```

Änderungen im System

Symfony/Console im CommandController (4)

```
    [...]
    // Abfrage
    $name = $this->output->ask('What is your name?' . PHP_EOL, 'Bob', array('Bob', 'Sally', 'Blake'));
    $this->outputLine('Hello %s.', array($name));

    // Prompt
    $likesDogs = $this->output->askConfirmation('Do you like dogs?');
    if ($likesDogs) {
      $this->outputLine('You do like dogs!');
    }

    // Progress
    $this->output->progressStart(600);
    for ($i = 0; $i < 300; $i ++) {
      $this->output->progressAdvance();
      usleep(5000);
    }
    $this->output->progressFinish();
  }
}
?>
```


TYPO3

Änderungen im System

Backend Login API (1)

- Das Backend-Login wurde nun komplett als API realisiert und lässt sich damit selbst per Programmierung ansprechen
- Grundsätzlich muss man dazu einen sogenannten Login-Provider in der Datei `ext_localconf.php` registrieren:

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['backend']['loginProviders'][1433416020] = [
  'provider' => \TYPO3\CMS\Backend\LoginProvider\UsernamePasswordLoginProvider::class,
  'sorting' => 50,
  'icon-class' => 'fa-key',
  'label' => 'LLL:EXT:backend/Resources/Private/Language/locallang.xlf:login.link'
];
```

Änderungen im System

Backend Login API (2)

- Die Optionen sind wie folgt definiert:
  - `provider`:
    Name der Login-Provider Klasse, die das Interface `TYPO3\CMS\Backend\LoginProvider\LoginProviderInterface` implementiert
  - `sorting`:
    Sortierung, der die Reihenfolge der Links zum Login-Provider festlegt
  - `icon-class`:
    Font-Awesome Icon-Name für den Link auf dem Login-Screen
  - `label`:
    Label für den Link auf dem Login-Screen

Änderungen im System

Backend Login API (3)

- Das `LoginProviderInterface` enthält lediglich die Methode
  `public function render(StandaloneView $view, PageRenderer $pageRenderer, LoginController $loginController);`
- Die Parameter sind wie folgt definiert:
  - `$view`:
    Der Fluid-StandaloneView, welcher den Login-Screen rendert. Hier muss das Template-File gesetzt und die Variablen übergeben werden, die man benötigt.
  - `$pageRenderer`:
    PageRenderer-Instanz, welche z.B. JavaScript-Ressourcen einbringen kann.
  - `$loginController`:
    Instanz des Login-Controllers.

Änderungen im System

Backend Login API (4)

- Das Template muss `<f:layout name="Login">` und `<f:section name="loginFormFields">` (für die Formular-Felder) enthalten:

```
<f:layout name="Login" />
<f:section name="loginFormFields">
  <div class="form-group t3js-login-openid-section" id="t3-login-openid_url-section">
    <div class="input-group">
      <input type="text" id="openid_url"
        name="openid_url"
        value="{presetOpenId}"
        autofocus="autofocus"
        placeholder="{f:translate(key: 'openId', extensionName: 'openid')}"
        class="form-control input-login t3js-clearable t3js-login-openid-field" />
      <div class="input-group-addon">
        <span class="fa fa-openid"></span>
      </div>
    </div>
  </div>
</f:section>
```

Änderungen im System

`CategoryRegistry` mit neue Optionen

- Die Methode `CategoryRegistry->addTcaColumn` hat Optionen erhalten, um sowohl `l10n_mode` als auch `l10n_display` zu setzen:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable(
  $extensionKey,
  $tableName,
  'categories',
  array(
    'l10n_mode' => 'string (keyword)',
    'l10n_display' => 'list of keywords'
  )
);
```

Änderungen im System

Sprites in Backend Modulen

- Backend Module (z.B. Hauptmodule wie "Web" sowie Untermodule wie "Filelist") können nun Sprites als Icons verwenden
  (nur Sprites, die TYPO3 bekannt sind!)
- Beispiel:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::addModule(
  'web',
  'layout',
  'top',
  \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extPath($_EXTKEY) . 'Modules/Layout/',
  array(
    'script' => '_DISPATCH',
    'access' => 'user,group',
    'name' => 'web_layout',
    'configuration' => array('icon' => 'module-web'),
    'labels' => array(
      'll_ref' => 'LLL:EXT:cms/layout/locallang_mod.xlf',
    ),
  )
);
```

Änderungen im System

FormEngine NodeFactory API (1)

- Es ist nun möglich, neue Nodes zu registrieren und bestehende zu überschreiben

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['formEngine']['nodeRegistry'][1433196792] = array(
  'nodeName' => 'input',
  'priority' => 40,
  'class' => \MyVendor\MyExtension\Form\Element\T3editorElement::class
);
```

- Dies registriert eine neue Klasse, die den TCA-Type `input` rendert und das `NodeInterface` implementieren muss
- Als Array-Key wird der Unix-Timestamp des Zeitpunkts verwendet, wenn man das Element zufügt

Änderungen im System

FormEngine NodeFactory API (2)

- Wenn mehrere Registry-Elemente für den selben Typ registriert werden, gewinnt das, mit der höchsten Priorität (0-100)
- Ein neuer TCA-Type wird wie folgt registriert:

TCA

```
'columns' => array(
  'bodytext' => array(
    'config' => array(
      'type' => 'text',
      'renderType' => '3dCloud',
    ),
  ),
)
```

ext_localconf.php

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['formEngine']['nodeRegistry'][1433197759] = array(
  'nodeName' => '3dCloud',
  'priority' => 40,
  'class' => \MyVendor\MyExtension\Form\Element\ShowTextAs3dCloudElement::class
);
```

Änderungen im System

Signal `postProcessMirrorUrl` wurde verschoben

- Die Klasse für das Signal `postProcessMirrorUrl` wurde geändert

BREAKING CHANGE!

- Mit folgendem Code kann man die Klasse je nach TYPO3-Version ansprechen:

```
$signalSlotDispatcher->connect(
  version_compare(TYPO3_version, '7.0', '<')
    ? 'TYPO3\\CMS\\Lang\\Service\\UpdateTranslationService'
    : 'TYPO3\\CMS\\Lang\\Service\\TranslationService',
  'postProcessMirrorUrl',
  'Vendor\\Extension\\Slots\\CustomMirror',
  'postProcessMirrorUrl'
);
```

Kapitel 4:

Extbase & Fluid

Extbase & Fluid

ActionMenuItemGroupViewHelper (1)

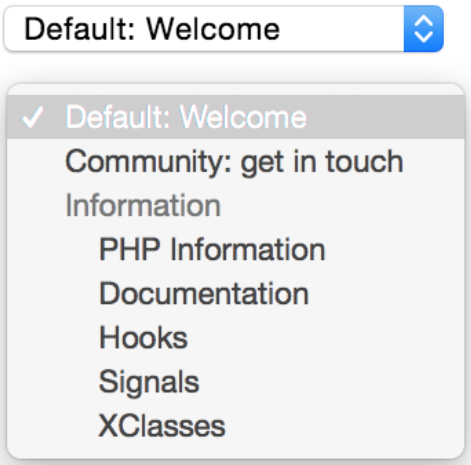
- Mit diesem neuen ViewHelper können im Backend Optionen-Gruppen für Select-Felder verwendet werden
- Beispiel:

```
<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>
```

Extbase & Fluid

ActionMenuItemGroupViewHelper (2)

- Beispiel auf der vorherigen Slide resultiert in folgender Ausgabe:



Extbase & Fluid

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"!

Extbase & Fluid

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)

Extbase & Fluid

Neue Eigenschaften im cObject FLUIDTEMPLATE (2)

- TypoScript 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
  }
}
```

Extbase & Fluid

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)

Extbase & Fluid

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>
```

Extbase & Fluid

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

Extbase & Fluid

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) {
    [...]
}
```

Extbase & Fluid

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
      }
    }
  }
}
```

Kapitel 5:

Veraltete und entfernte Funktionen

Veraltete/Entfernte Funktionen

FormEngine Refactoring

TCA:

- Optionen `_PADDING`, `_VALIGN` und `DISTANCE` wurden aus `TCA['aTable']['columns']['aField']['config']['wizards']` entfernt
- Schlüssel `TCA['aTable']['ctrl']['mainPalette']` wurde entfernt

TSconfig:

- Schlüssel `mod.web_layout.tt_content.fieldOrder` und `TCEFORM.aTable.aField.linkTitleToSelf` wurden entfernt

Hooks:

- Hooks verwenden nun den Schlüssel `type` anstatt `form_type`
- Hook `getSingleFieldClass` wurde entfernt

Veraltete/Entfernte Funktionen

`IdentityMap` wurde aus der Extbase Persistenz entfernt

- Die Klasse `IdentityMap` wurde aus der Extbase Persistenz entfernt (bei Verwendung wird eine `ReflectionException` erzeugt)
- Damit ist der Zugriff der `IdentityMap` innerhalb von `DataMapper` und `Repository` nicht mehr möglich
- Anstelle der `IdentityMap` kann nun die Persistenz "Sessions" verwendet werden:

```
$session = GeneralUtility::makeInstance(ObjectManager::class)->get(
  \TYPO3\CMS\Extbase\Persistence\Generic\Session::class
);

$session->registerObject($object, $identifier);

if($session->hasIdentifier($identifier)) {
  $object = $session->getObjectByIdentifier($identifier, $className);
}
```

Veraltete/Entfernte Funktionen

Diverses (1)

- Datei `typo3conf/extTables.php` ist "deprecated". Stattdessen sollte die folgende Datei verwendet werden:

  `typo3conf/ext/<your_extension>/Configuration/TCA/Overrides/pages.php`

- Konfiguration `$TYPO3_CONF_VARS[GFX][png_to_gif]` wurde entfernt
- Installationen, die die Extension `rsaauth` nicht installiert haben, übertragen das Login-Passwort nun im Klartext
  (Abhilfe: Extension `rsaauth` installieren oder BE mittels https sichern)
- Methode `exec_SELECTgetRows()` wertet nun den Parameter `$uidIndexField` aus. Dadurch kann es zu Fehlern kommen, wenn das spezifizierte Feld in der Datenbank nicht vorhanden ist.

Veraltete/Entfernte Funktionen

Diverses (2)

- DBAL-Option `config.classFile` wurde entfernt
- Optionen `iconOnly` und `styleAttributes` des `CshViewHelper` sind "deprecated"
- TypoScript Option `page.bgImg` ist ab sofort "deprecated"
- Methode `isEnabled()` der Klasse `T3editor` ist ab sofort "deprecated"
- Der alte TYPO3 ClassLoader wurde zugunsten eines Composer ClassLoaders entfernt

Kapitel 6:

Quellen und Autoren

Quellen und Autoren

Quellennachweis

TYPO3 News:

- http://typo3.org/news

Release Infos:

- http://wiki.typo3.org/TYPO3_CMS_7.3.0
- INSTALL.md and ChangeLog
- typo3/sysext/core/Documentation/Changelog/7.3/*

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

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 69 260 99 70 50 | facebook.com/pluswerk