

TYPO3 CMS 7.1 – What's New

Übersicht der neuen Funktionen, Änderungen und Verbesserungen

Patrick Lobacher und Michael Schams

TYPO3 CMS 7.1 - 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 7.1 - Die Fakten

- Veröffentlichungsdatum: 17. Februar 2015
- Releasetyp: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Hauptfokus: Core Cleanup and Streamlining



Einführung

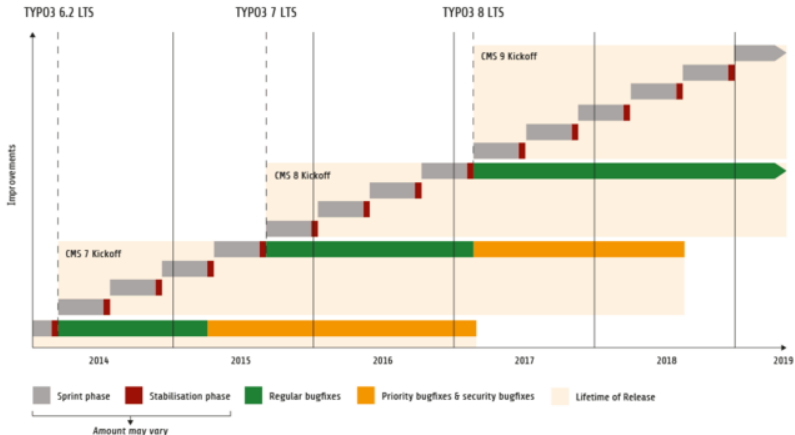
Systemvoraussetzungen

- 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 → 17/Feb/2015 Core Cleanup & Streamlining
- v7.2 → 10/Mär/2015 Frontend
- v7.3 → 21/Apr/2015 Composer Ecosystem
- v7.4 → 09/Jun/2015 Backend Overhaul Vol 2
- v7.5 → 28/Jul/2015 (*noch unbestimmt*)
- v7.6 → 13/Okt/2015 pre-LTS inferno
- v7.7 → 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/7.1
$ tar xzf typo3_src-7.1.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.1.0 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_7.1
- 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

Look & Feel: Date/Time Picker

Date/Time Picker im Backend wurde auf Twitter Bootstrap umgestellt

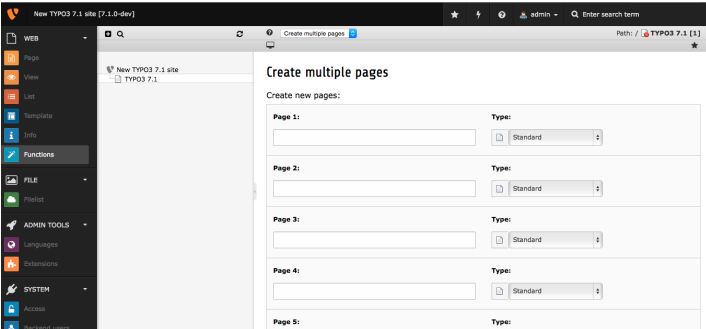
The screenshot shows a date/time picker interface. At the top, it displays the month and year: "February 2015". Below this is a calendar grid with days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and dates. The date "14" is highlighted in blue. To the right of the calendar is a time selection area showing "06 : 57". Below the calendar and time selection, there is a search bar containing "6:57 14-2-2015" and a calendar icon. To the right of the search bar is a "Enabled" checkbox. The interface also includes a "Categories" dropdown menu and a "to Subpages" link.

Backend User Interface

Look & Feel: Funktionen-Modul

Die Funktionen "Seiten sortieren" und "mehrere Seiten erzeugen" sind jetzt unter WEB => Funktionen zu finden

(in TYPO3 CMS < 7.1 waren diese unter "WEB => Funktionen => Wizards" zu finden)



The screenshot displays the TYPO3 CMS Backend User Interface. The top navigation bar shows the current site as "New TYPO3 7.1 site [7.1.0-dev]" and the user as "admin". The left sidebar contains a menu with categories: WEB, FILE, ADMIN TOOLS, and SYSTEM. The "WEB" category is expanded, showing options like Page, View, List, Template, Info, and **Functions**. The "Functions" option is selected, leading to the "Create multiple pages" wizard. The wizard is titled "Create multiple pages" and contains a section "Create new pages:" with five rows. Each row is labeled "Page 1:" through "Page 5:" and includes a text input field and a "Type:" dropdown menu set to "Standard".

Backend User Interface

Look & Feel: Access-Modul

Im Modul WEB => Access kann man Benutzer und/oder Gruppen unverändert lassen, wenn man nur die Berechtigungen ändern möchte

Permissions: EDIT

Owner

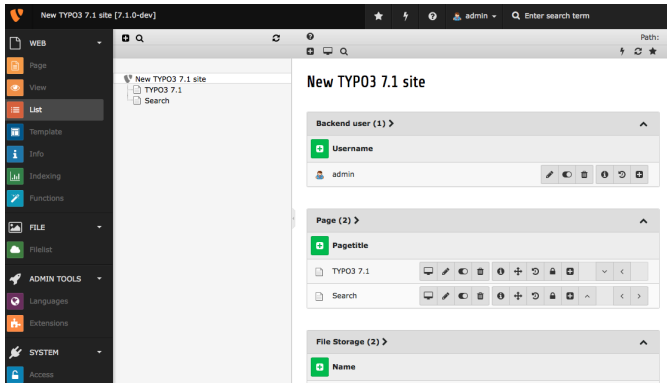
Group

| | Show page | Edit content | Edit page | Delete page | New pages |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Owner | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Group | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Everybody | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Backend User Interface

Look & Feel: Icons im List-Modul

Icons ("Action Buttons") im List-Modul sind nun in zwei Gruppen angeordnet (primäre Aktionen (lesen, speichern, löschen) zuerst, anschließend die sekundären Icons)



Kapitel 2: TSconfig & TypoScript

Tsconfig & TypeScript

StdWrap für `page.headTag`

- TypeScript Option `page.headTag` hat nun `stdWrap`-Funktionalität

```
page = PAGE
page.headTag = <head>
page.headTag.override = <head class="special">
page.headTag.override.if {
    isInList.field = uid
    value = 24
}
```


TScnfig & TypoScript

JavaScript-Dateien asynchron laden

- JavaScript-Dateien können nun asynchron geladen werden

```
page {
  includeJS {
    jsFile = /path/to/file.js
    jsFile.async = 1
  }
}
```

- Das gilt für:
 - includeJSlibs / includeJSLibs
 - includeJSFooterlibs
 - includeJS
 - includeJSFooter

TScnfig & TypoScript

HMENU Eigenschaft mit additionalWhere

- TypoScript cObject HMENU erhält eine neue Eigenschaft `additionalWhere`
- Jenes erlaubt eine spezifischere DB Abfrage (z.B. Filterung)
- Beispiel:

```
lib.authormenu = HMENU
lib.authormenu.1 = TMENU
lib.authormenu.1.additionalWhere = AND author!=""
```

Zusätzliche Eigenschaften für HMENU Browse-Menü

- Zwei neue Eigenschaften für das cObject HMENU (Option "special=browse"), um detaillierter definieren zu können, welche Seiten im Menü erscheinen sollen:
 - `excludeNoSearchPages`
 - `includeNotInMenu`
- Beispiel:

```
lib.browsemenu = HMENU
lib.browsemenu.special = browse
lib.browsemenu.special.excludeNoSearchPages = 1
lib.browsemenu.includeNotInMenu = 1
```

TScnfig & TypoScript

Mehrere HTTP-Header

- HTTP Header können nun mittels `config.additionalHeaders` als Array gesetzt werden
- Das ermöglicht es, mehreren Header-Zeilen auf einmal zu konfigurieren

```
config.additionalHeaders {
    10 {
        # header string
        header = WWW-Authenticate: Negotiate
        # (optional) replace previous headers with the same name (default: 1)
        replace = 0
        # (optional) force HTTP response code
        httpResponseCode = 401
    }
    # set second additional HTTP header
    20.header = Cache-control: Private
}
```

Tsconfig & TypoScript

Option "auto" für `config.absRefPrefix`

- TypoScript Konfiguration `config.absRefPrefix` kann verwendet werden, um der URL einen Prefix bei relativen Pfaden zu geben. Als Alternative zu `config.baseURL` (um eine bestimmte Domain zu spezifizieren), erkennt `absRefPrefix` die Site-Root automatisch:

```
config.absRefPrefix = auto
```

anstelle von:

```
[ApplicationContext = Production]  
config.absRefPrefix = /
```

```
[ApplicationContext = Testing]  
config.absRefPrefix = /my_site_root/
```

Hinweis: diese Option ist "Multi-Domain"-sicher und mehrfaches Caching der selben Daten wird verhindern.

Tsconfig & TypoScript

Zwei-Zeichen ISO Code für `sys_language` (1)

- Die Behandlung von Sprachen wird durch Einträge in DB Tabelle `sys_language` vorgenommen, die durch `sys_language_uid` referenziert werden
- In TYPO3 CMS 7.1 wurden ISO 639-1 Zwei-Zeichen Codes implementiert:
 - Neues DB Feld: `sys_language.language_isocode`
 - Neue TypoScript-Option: `sys_language_isocode`

Hinweis: bei **ISO 639** handelt es sich um eine Sammlung von Standards der "International Organization for Standardization". Eine List der ISO 639-1 Codes ist hier abrufbar:

http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

TScnfig & TypoScript

Zwei-Zeichen ISO Code für sys_language (2)

■ Beispiel:

```
# Danish by default
config.sys_language_uid = 0
config.sys_language_isocode_default = da

[globalVar = GP:L = 1]
  # ISO code stored in table sys_language (uid 1)
  config.sys_language_uid = 1
  # overwrite ISO code as required
  config.sys_language_isocode = fr
[GLOBAL]

page.10 = TEXT
page.10.data = TSFE:sys_language_isocode
page.10.wrap = <div class="main" data-language="|">
```

TSconfig & TypoScript

Eigene Conditions im Backend

- Eigene Conditions für das **Frontend** wurden bereits mit TYPO3 CMS 7.0 eingeführt
- Seit TYPO3 CMS 7.1 ist es nun auch möglich, eigene Conditions für das **Backend** zu implementieren
- Die Condition muss von `AbstractCondition` ableiten und die Methode `matchCondition` bereitstellen
- Anwendungsbeispiel in TypoScript:

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition]
```

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = 7]
```

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = 7, != 6]
```

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = {$mysite.myconstant}]
```


TScnfig & TypoScript

Zufügen von Icons in TCEFORM via PageTScnfig

- Eigene Werte und Labels von Select-Feldern können bereits mit der PageTScnfig Option `addItem`s vergeben werden
- Nun können auch Icons für diese Felder definiert werden
 - Option 1: mittels `addItem`s und der Eigenschaft `.icon`
 - Option 2: mittels `altIcons` (generell für alle Felder)
- Beispiel:

```
TCEFORM.pages.doktype.addItem {
    10 = My Label
    10.icon = EXT:t3skin/icons/gfx/i/pages.gif
}
TCEFORM.pages.doktype.altIcons {
    10 = EXT:myext/icon.gif
}
```

Element Browser: Mountpoints hinzufügen

- Neue UserTScnfig Option `.append` erlaubt es Administratoren Mountpoints **hinzuzufügen**, anstatt die Liste der konfigurierten DB Mountpoints eines Benutzers neu zu schreiben
- Beispiel:

```
options.pageTree.altElementBrowserMountPoints = 20,31  
options.pageTree.altElementBrowserMountPoints.append = 1
```

Überschreiben der Labels von Radio-Buttons und Checkboxes

- Labels von Radio-Buttons und Checkboxes können nun überschrieben werden
- Beispiel:

```
// field with a single checkbox (use ".default")
TCEFORM.pages.hidden.altLabels.default = new label
TCEFORM.pages.hidden.altLabels.default = LLL:path/to/languagefile.xlf:individuallabel

// field with multiple checkboxes (0, 1, 2, 3...)
TCEFORM.pages.l18n_cfg.altLabels.0 = new label of first checkbox
TCEFORM.pages.l18n_cfg.altLabels.1 = new label of second checkbox
TCEFORM.pages.l18n_cfg.altLabels.2 = new label of third checkbox
...
```

TSconfig & TypoScript

Diverses (1)

- Breite und Höhe des Element-Browsers können nun per UserTSconfig festgelegt werden

```
options.popupWindowSize = 400x900  
options.RTE.popupWindowSize = 200x200
```

- PageTSconfig: mit einer neue RTE-Konfiguration kann das Standard-Ziel von Links beeinflusst werden

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

Wobei [type] zum Beispiel page, file, url, mail or spec sein kann
(Extensions können weitere Typen zur Verfügung stellen)

Tsconfig & TypoScript

Diverses (2)

- Standardmäßig sind Section-Headlines der Indexed-Search Resultate verlinkt. Das kann nun mittels TypoScript deaktiviert werden

```
plugin.tx_indexedsearch.linkSectionTitles = 0
```

- `getData` kann jetzt auch `field`-Daten abfragen (nicht nur Arrays, wie beispielsweise `GPVar` und `TSFE`)

```
10 = TEXT  
10.data = field:fieldname|level1|level2
```

- TypoScript Konfiguration `config.pageTitle` hat jetzt `stdWrap`-Funktionalität

```
# make value of <title> upper case  
page = PAGE  
page.config.pageTitle.case = upper
```

Kapitel 3: Änderungen im System

Änderungen im System

TCA: Maximum chars in text element

- TCA-Typ `text` unterstützt nun das HTML5-Attribut `maxlength`, um die maximale Anzahl der einzugebenden Zeichen zu beschränken (Hinweis: Zeilenumbrüche zählen hierbei als zwei Zeichen)

```
'teaser' => array(  
    'label' => 'Teaser',  
    'config' => array(  
        'type' => 'text',  
        'cols' => 60,  
        'rows' => 2,  
        'max' => '30' // <-- maxlength  
    )  
)
```

Es ist zu beachten, dass nicht alle Browser dieses Attribut unterstützen.

Siehe: [Browserübersicht](#)

Änderungen im System

New SplFileInfo implementation

- Neue Klasse: TYPO3\CMS\Core\Type\File\FileInfo
- Diese erweitert SplFileInfo, die wiederum Meta-Informationen von Dateien ermittelt

```
$fileIdentifier = '/tmp/foo.html';  
$fileInfo = GeneralUtility::makeInstance(  
    \TYPO3\CMS\Core\Type\File\FileInfo::class,  
    $fileIdentifier  
);  
echo $fileInfo->getMimeType(); // output: text/html
```

- Entwickler können über folgenden Hook auf die Funktionalität zugreifen:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']  
    [\TYPO3\CMS\Core\Type\File\FileInfo::class]['mimeTypeGuessers']
```


Änderungen im System

UserFunc in TCA Display Condition

- `userFunc displayCondition` ermöglicht es auf jeden erdenklichen Status und jede Condition zu prüfen
- Sollte irgendeine Situation nicht mit den existierenden Checks abgefangen werden können, ist es auch möglich, eigene Funktionen zu schreiben (diese müssen lediglich TRUE/FALSE zurückgeben, um das entsprechende TCA Field sichtbar zu machen oder zu verbergen)

```
$GLOBALS['TCA']['tt_content']['columns']['bodytext']['displayCond'] =  
    'USER:Vendor\\Example\\User\\ElementConditionMatcher->  
        checkHeaderGiven:any:more:information';
```

Änderungen im System

API für Twitter Bootstrap Modals (1)

- Zwei neue API Methoden um Modal Popups zu erzeugen/entfernen:
 - `TYPO3.Modal.confirm(title, content, severity, buttons)`
 - `TYPO3.Modal.dismiss()`
- Optionen `title` und `content` sind mindestens erforderlich
- Optionen `buttons.text` und `buttons.trigger` sind erforderlich, wenn `buttons` verwendet wird
- Beispiel 1:

```
TYPO3.Modal.confirm(  
    'The title of the modal',           // title  
    'This the the body of the modal', // content  
    TYPO3.Severity.warning            // severity  
);
```

Änderungen im System

API für Twitter Bootstrap Modals (2)

■ Beispiel 2:

```
TYPO3.Modal.confirm('Warning', 'You may break the internet!',
    TYPO3.Severity.warning,
    [
        {
            text: 'Break it',
            active: true,
            trigger: function() { ... }
        },
        {
            text: 'Abort!',
            trigger: function() {
                TYPO3.Modal.dismiss();
            }
        }
    ]
);
```

Änderungen im System

JavaScript Storage API (1)

- Mittels JavaScript kann auf die BE User Konfiguration zugegriffen werden (`$BE_USER->uc`, einfache Key-Value Paare)
- Zusätzlich kann nun auch der HTML5 Standard [localStorage](#) verwendet werden, um Daten (Client-seitig) im Browser des Benutzers zu speichern und auszulesen
- Zwei neue global TYPO3 Objekte:
 - `top.TYPO3.Storage.Client`
 - `top.TYPO3.Storage.Persistent`
- Jedes Objekt hat folgende API Methoden:
 - `get(key)`: Daten holen
 - `set(key, value)`: Daten schreiben
 - `isset(key)`: Prüfen, ob key genutzt wird
 - `clear()`: Löschen des Speichers

Änderungen im System

JavaScript Storage API (2)

■ Beispiel:

```
// get value of key 'startModule'  
var value = top.TYPO3.Storage.Persistent.get('startModule');  
  
// write value 'web_info' as key 'start_module'  
top.TYPO3.Storage.Persistent.set('startModule', 'web_info');
```

Änderungen im System

Inline Rendering von Checkboxes

- Die Konfiguration `inline` sorgt bei "cols" dafür, dass Checkboxes nebeneinander dargestellt werden, um Platz im Backend User Interface zu sparen

```
'weekdays' => array(
  'label' => 'Weekdays',
  'config' => array(
    'type' => 'check',
    'items' => array(
      array('Mo', ''),
      array('Tu', ''),
      array('We', ''),
      array('Th', ''),
      array('Fr', ''),
      array('Sa', ''),
      array('Su', '')
    ),
    'cols' => 'inline'
  )
),
...
```

Änderungen im System

Content Object Registration

- Es wurde eine neue globale Option eingeführt, um cObjects wie TEXT zu registrieren bzw. zu erweitern

- Eine Liste aller verfügbaren cObjects ist verfügbar als:

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['ContentObjects']
```

- Beispiel: ein neues cObject EXAMPLE registrieren

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['ContentObjects']['EXAMPLE'] =  
    Vendor\MyExtension\ContentObject\ExampleContentObject::class;
```

- Die registrierte Klasse muss von der folgenden Klasse ableiten:

```
TYPO3\CMS\Frontend\ContentObject\AbstractContentObject
```

- Idealerweise speichert man seine Datei im Verzeichnis

```
typo3conf/myextension/Classes/ContentObject/
```

um für zukünftige Autoload-Funktionen vorbereitet zu sein

Änderungen im System

Hooks und Signals (1)

- Neuer Hook wurde am Ende von `PageRepository->init()` hinzugefügt, mit dem die Sichtbarkeit von Seiten beeinflusst werden kann

- Der Hook kann wie folgt registriert werden:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']  
    [\TYPO3\CMS\Frontend\Page\PageRepository::class]['init']
```

- Die Hook-Klasse muss das folgende Interface implementieren:

```
\TYPO3\CMS\Frontend\Page\PageRepositoryInitHookInterface
```


Änderungen im System

Hooks und Signals (2)

- Neuer Hook wurde zu PageLayoutView hinzugefügt, um das Rendering des Footers von Inhaltselementen im Backend manipulieren zu können

- Beispiel:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']  
    ['cms/layout/class.tx_cms_layout.php']['tt_content_drawFooter'];
```

- Die Hook-Klasse muss das folgende Interface implementieren:

```
\TYPO3\CMS\Backend\View\PageLayoutViewDrawFooterHookInterface
```

Änderungen im System

Hooks und Signals (3)

- Es wurde ein Hook als Post-Prozessor zu `BackendUtility::countVersionsOfRecordsOnPage` hinzugefügt
- Dieser wird z.B. verwendet, um Workspace-Zustände im Seitenbaum zu visualisieren
- Der Hook kann wie folgt registriert werden:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']  
    ['t3lib/class.t3lib_befunc.php']['countVersionsOfRecordsOnPage'][] =  
    'My\Package\HookClass->hookMethod';
```

Änderungen im System

Hooks und Signals (4)

- Neues Signal wurde am Ende der Methode `DataPreprocessor::fetchRecord()` hinzugefügt
- Jenes kann dazu verwendet werden, um das Array `regTableItems_data` zu manipulieren, damit die manipulierten Daten in TCEForms angezeigt werden können

```
$this->getSignalSlotDispatcher()->dispatch(  
    \TYPO3\CMS\Backend\Form\DataPreprocessor::class,  
    'fetchRecordPostProcessing',  
    array($this)  
);
```

Änderungen im System

Hooks und Signals (5)

- Neues Signal wurde eingeführt, um zusätzlichen Code bei der Registrierung des Mailer-Objekts auszuführen (z.B. Swift Mailer Plugins)

```
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class  
);  
  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Core\Mail\Mailer::class,  
    'postInitializeMailer',  
    \Vendor\Package\Slots\MailerSlot::class,  
    'registerPlugin'  
);
```

Änderungen im System

Multiple UID in PageRepository::getMenu()

- Methode PageRepository::getMenu() kann nun auch ein Array aufnehmen, um mehrerer Root-Seiten zu definieren

```
$pageRepository = new \TYPO3\CMS\Frontend\Page\PageRepository();  
$pageRepository->init(FALSE);  
$rows = $pageRepository->getMenu(array(2, 3));
```

Kapitel 4: Extbase & Fluid

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!

Kapitel 5:

Veraltete und entfernte Funktionen

Veraltete/Entfernte Funktionen

`$TYPO3_CONF_VARS[SYS][compat_version]`

- Die Option `$TYPO3_CONF_VARS[SYS][compat_version]` (gesetzt beim Update im Install Tool wizard) wurde entfernt
- Alle Prüfungen gegen `GeneralUtility::compat_version` werden nun gegen die Konstante `TYP03_branch` gemacht

Hinweis: TypeScript Conditions, die auf `compat_version` prüfen, haben nun keine Wirkung mehr!

Veraltete/Entfernte Funktionen

Inline styles of <blockquote> tag

- CSS Styled Content rendert <blockquote> über die TypoScript Option `lib.parseFunc_RTE`
- Diese Zeilen wurden ersatzlos entfernt:

```
lib.parseFunc_RTE.externalBlocks.blockquote.callRecursive.tagStdWrap.HTMLparser = 1
lib.parseFunc_RTE.externalBlocks.blockquote.callRecursive.tagStdWrap.HTMLparser.tags.blockquote.overrideAttribs = style="margin-bottom:0;margin-top:0;"
```

- Das bedeutet, die Inline-Styles "margin-bottom:0;margin-top:0;" werden dem <blockquote>-Tag nicht mehr hinzugefügt

Hinweis: nach einem Update auf TYPO3 CMS 7.1 könnte sich das Styling von <blockquote> geändert haben

Veraltete/Entfernte Funktionen

Workspaces: Feld `disable_autocreate`

- Das Feld `disable_autocreate` wurde von `EXT:workspaces` entfernt
- Sollten TYPO3 Extensions dieses Feld verwenden, wird ein SQL Fehler erzeugt

Veraltete/Entfernte Funktionen

Funktion: `include_once`

- Die Funktionalität, um PHP-Dateien mittels `include_once` innerhalb von Modul-Funktionen (wie z.B. dem Info-Modul) zu inkludieren, wurde entfernt
- Das gilt für folgende Module:
 - Web => Page
 - Web => Page - New Content Element Wizard
 - Web => Functions
 - Web => Info
 - Web => Template
 - Web => Recycler
 - User => Task Center
 - System => Scheduler

Veraltete/Entfernte Funktionen

TypoScript Option: `config.meaningfulTempFilePrefix`

- Früher war es möglich, per TypoScript Teile des Original-Dateinamens zum Dateinamen hinzuzufügen, der vom GIFBUILDER generiert wird
- Jenes war mit folgender TypoScript Option möglich:
`config.meaningfulTempFilePrefix`
(standardmäßig verwendete der GIFBUILDER lediglich ein Hash-Wert als Dateinamen)
- Diese Option wurde entfernt
(Dateinamen im Verzeichnis `typo3temp/GB/` enthalten nun den Original-Dateinamen automatisch)

Veraltete/Entfernte Funktionen

Removed files

- Die folgenden **Dateien** wurden entfernt:
 - typo3/file_edit.php
 - typo3/file_newfolder.php
 - typo3/file_rename.php
 - typo3/file_upload.php
 - typo3/show_rechis.php
 - typo3/listframe_loader.php
- Deren Funktionalität wurde in Backend Module integriert, z.B. `typo3/file_edit.php` in `BackendUtility::getModuleUrl('file_edit');`

Veraltete/Entfernte Funktionen

ExtJS DateTimePicker

- Die ExtJS Komponente `Ext.ux.DateTimePicker` wurde entfernt und gegen die Twitter Bootstrap Alternative ersetzt (siehe Kapitel "Backend User Interface")
- Das betrifft zum Beispiel die System Extensions `EXT:blog` und `EXT:scheduler`

Hinweis: Extensions, die die (als "deprecated" markierte) Komponente `Ext.ux.DateTimePicker` benötigen, werden mit hoher Wahrscheinlichkeit nicht mehr funktionieren.

Veraltete/Entfernte Funktionen

Änderungen beim Access List Render Mode

- Die folgende **Variable** wurde entfernt:
`$GLOBALS[TYPO3_CONF_VARS][BE][accessListRenderMode]`
- Die entsprechenden Felder in den TCA Tabellen `be_users` und `be_groups` besitzen nun die Standardwert "checkbox"
- Jenes kann in der Datei `typo3conf/extTables.php` bei Bedarf angepasst werden:

```
$GLOBALS['TCA']['be_users']['columns']['file_permissions']['config']['renderMode'] = 'singlebox';  
$GLOBALS['TCA']['be_users']['columns']['userMods']['config']['renderMode'] = 'singlebox';
```

```
$GLOBALS['TCA']['be_groups']['columns']['file_permissions']['config']['renderMode'] = 'singlebox';  
$GLOBALS['TCA']['be_groups']['columns']['pagetypes_select']['config']['renderMode'] = 'singlebox';  
$GLOBALS['TCA']['be_groups']['columns']['tables_select']['config']['renderMode'] = 'singlebox';  
$GLOBALS['TCA']['be_groups']['columns']['tables_modify']['config']['renderMode'] = 'singlebox';  
$GLOBALS['TCA']['be_groups']['columns']['non_exclude_fields']['config']['renderMode'] = 'singlebox';  
$GLOBALS['TCA']['be_groups']['columns']['userMods']['config']['renderMode'] = 'singlebox';
```

Veraltete/Entfernte Funktionen

Content Element "Mailform"

- Die Mailform Funktionalität, die das cObject FORM bereitstellt, wurde vom TYPO3 Core entfernt
(diese ist jedoch weiterhin in EXT:compatibility6 vorhanden)
- Die folgenden Optionen wurden als "deprecated" markiert:

```
$TYPO3_CONF_VARS [FE] [secureFormmail]  
$TYPO3_CONF_VARS [FE] [strictFormmail]  
$TYPO3_CONF_VARS [FE] [formmailMaxAttachmentSize]
```

- Die folgenden Methoden im TypoScriptFrontendController wurden entfernt:

```
protected checkDataSubmission()  
protected sendFormmail()  
public extractRecipientCopy()  
public codeString()  
protected roundTripCryptString()
```

Veraltete/Entfernte Funktionen

Funktionalität überarbeitet (1)

- Die `EXT:indexed_search` wird automatisch aktiviert, sobald sie installiert wird. Das bedeutet auch, dass die TypoScript Optionen `config.index_enable` und `config.index_externals` ebenfalls automatisch aktiviert werden
- TSconfig `web_func.menu.wiz` ändert sich zu `web_func.menu.functions`
- Extensions, die sich in die Toolbar oben rechts einklinken, müssen das folgende neue Interface implementieren:
`TYPO3\CMS\Backend\Toolbar\ToolbarItemInterface`
und müssen unter folgendem Schlüssel registriert werden:
`$GLOBALS['TYPO3_CONF_VARS']['BE']['toolbarItems']`

Veraltete/Entfernte Funktionen

Funktionalität überarbeitet (2)

- Die Datei `typo3/js/tree.js` wurde ersetzt gegen `EXT:backend/Resources/Public/JavaScript/LegacyTree.js` (basierend auf jQuery)
- Die Variable `$GLOBALS['WEBMOUNTS']` wurde ersetzt gegen `$GLOBALS['BE_USER']->returnWebmounts()`
- Die Unterstützung von `.t3-table` und `.t3-button` wurde entfernt (Twitter Bootstrap CSS Klassen implementieren die Styles jetzt)
- Länderflaggen (PNG-Bilder) wurden von `typo3/gfx/flags/` und `typo3/sysex/t3skin/images/flags/` nach `typo3/sysex/core/Resources/Public/Icons/flags/` verschoben

Veraltete/Entfernte Funktionen

Funktionalität überarbeitet (3)

- CSS Styled Content TypoScript Templates der TYPO3 CMS Versionen 4.5 bis 6.1 wurden entfernt
- Die folgenden TypoScript cObjects wurden in die Legacy-Extension `EXT:compatibility6` verschoben:

| | | | |
|---------------|---------|--------|----------|
| SEARCHRESULTS | COLUMNS | OTABLE | CLEARGIF |
| IMGTEXT | CTABLE | HRULER | |

- Das Inhaltselement `search` wurde in die Legacy-Extension `EXT:compatibility6` verschoben
- Die folgenden TCA-Wizard-Optionen wurden entfernt:

| | | |
|-----------------------|----------------------|------------------------|
| <code>_PADDING</code> | <code>_VALIGN</code> | <code>_DISTANCE</code> |
|-----------------------|----------------------|------------------------|

Veraltete/Entfernte Funktionen

TypoScript-Option andWhere

- Die TypoScript-Option andWhere wurde als "deprecated" markiert
- Integratoren sollten die Eigenschaften where und markers verwenden:

```
page.30 = CONTENT
page.30 {
    table = tt_content
    select {
        pidInList = this
        orderBy = sorting
        where {
            dataWrap = sorting>{field:sorting}
        }
    }
}
```

```
page.60 = CONTENT
page.60 {
    table = tt_content
    select {
        pidInList = 73
        where = header != ###whatever###
        orderBy = ###sortfield###
        markers {
            whatever.data = GP:first
            sortfield.value = sor
            sortfield.wrap = |ting
        }
    }
}
```


Veraltete/Entfernte Funktionen

Entry-Points

- Die folgenden Entry-Points wurden als "deprecated" markiert:

- typo3/tce_file.php
- typo3/move_el.php
- typo3/tce_db.php
- typo3/login_frameset.php
- typo3/sysextd/cms/layout/db_new_content_el.php
- typo3/sysextd/cms/layout/db_layout.php

- Stattdessen kann nun folgendes verwendet werden:

```
\TYPO3\CMS\Backend\Utility\BackendUtility::getModuleUrl('<parameter>')
```

Wobei *<parameter>* eines der folgenden Elemente sein kann:

tce_file, move_element, tce_db, login_frameset,
new_content_element, web_layout

Veraltete/Entfernte Funktionen

Diverses (1)

- Die TypoScript-Option `config.xhtmlDoctype = xhtml_2` wurde als "deprecated" markiert

- Die folgenden Methoden wurden als "deprecated" markiert:

```
TYPO3\CMS\Backend\Utility\BackendUtility::implodeTSParams()  
TYPO3\CMS\Backend\Controller::makeLoginBoxImage()
```

- Die folgende Methode wurde als "deprecated" markiert:

```
LocalImageProcessor::getTemporaryImageWithText()
```

...und ersetzt gegen:

```
TYPO3\CMS\Core\Imaging\GraphicalFunctions::getTemporaryImageWithText()
```

- Die stdWrap Eigenschaften `textStyle` und `tableStyle` wurde als "deprecated" markiert

Veraltete/Entfernte Funktionen

Diverses (2)

- Die TypoScript-Option `page.includeJSlibs` wurde umbenannt zu `page.includeJSLibs` (Großbuchstabe "L") und die alte Option als "deprecated" markiert
- Die Condition `device` wurde als "deprecated" markiert
- Die Methode `DocumentTable::table()` wurde als "deprecated" markiert (Entwickler sollten dafür Fluid verwenden)
- Die folgende Methode wurde als "deprecated" markiert:

```
TYPO3\CMS\Frontend\Controller\  
    TypeScriptFrontendController::doXHTML_cleaning()
```

...ebenso die dazugehörige TypoScript-Option `config.xhtml_cleaning`

- Der folgende Hook wurde als "deprecated" markiert:
`$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['GLOBAL']['softRefParser_GL']`

Veraltete/Entfernte Funktionen

Diverses (3)

- Die folgenden Methoden wurden als "deprecated" markiert:

```
TypoScriptTemplateObjectBrowserModuleFunctionController::  
    verify_TSobjects()  
ExtendedTemplateService::ext_getKeyImage()  
ConfigurationForm::ext_getKeyImage()
```

- Die Ausführung von `contentObject->COBJECT()` wurde als "deprecated" markiert
(benutze stattdessen `$cObj->cObjGetSingle('...', $conf);`)
- Der direkte Zugriff auf `FormEngine::$renderReadOnly` wurde als "deprecated" markiert
(benutze stattdessen `AbstractFormElement::setRenderReadOnly(TRUE);`)

Veraltete/Entfernte Funktionen

Diverses (4)

- Die folgenden FormEngine-Methoden wurden als "deprecated" markiert:
 - `FormEngine::insertDefStyle`
 - `FormEngine::getAvailableLanguages()`
 - `FormEngine::sL()`
 - `FormEngine::renderVDEFDiff()`
 - `FormEngine::getLL()`
 - `FormEngine::getTSCpid()`
 - `FormEngine::getSingleField_typeFlex_langMenu()`
 - `FormEngine::getSingleField_typeFlex_sheetMenu()`
 - `FormEngine::getSpecConfFromString()`

Kapitel 6: Quellen und Autoren

Quellen und Autoren

Quellennachweis

TYPO3 News:

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

Release Infos:

- http://wiki.typo3.org/TYPO3\CMS_7.1.0
- [INSTALL.md](#) and [Changelog](#)
- [typo3/sysex/core/Documentation/Changelog/7.1/](http://typo3.org/sysex/core/Documentation/Changelog/7.1/)*

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, Ben van't Ende,
Michel Mix, Sinisa Mitrovic, Nena Jelena Radovic und Roberto Torresani

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

Lizenziert unter Creative Commons BY-NC-SA 3.0

