

TYPO3 CMS 7.4 – What's New

Übersicht der neuen Funktionen, Änderungen und Verbesserungen

Patrick Lobacher und Michael Schams

Kapitelübersicht

Einführung

Backend User Interface

TSconfig & TypoScript

TSconfig & TypoScript

Änderungen im System

Extbase & Fluid

Veraltete/Entfernte Funktionen

Quellen und Autoren

Einführung (Die Fakten)

Einführung

TYPO3 CMS 7.4 - Die Fakten

- Veröffentlichungsdatum: 04. August 2015
- Releasetyp: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Hauptfokus: Backend Overhaul Vol 2

Einführung

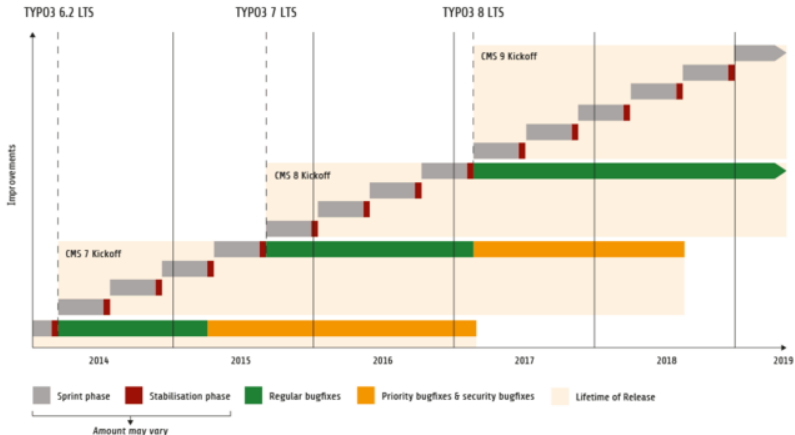
Systemvoraussetzungen

- 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 *(noch unbestimmt)*
- v7.6 xx/xxx/2015 **TYPO3 CMS 7 LTS** (Long Term Release)

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

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

Einführung

Installation

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

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/7.4
$ tar xzf typo3_src-7.4.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.4.0 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

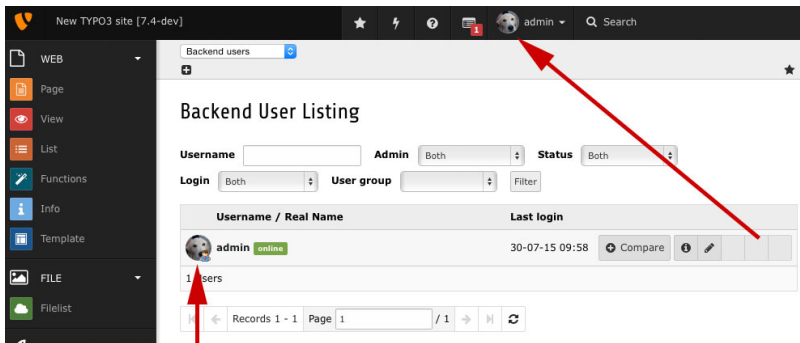
- Upgrades nur von TYPO3 CMS 6.2 LTS möglich
- TYPO3 CMS < 6.2 sollte man erst auf TYPO3 CMS 6.2 LTS aktualisieren
- Upgrade-Anleitung:
http://wiki.typo3.org/Upgrade#Upgrading_to_7.4
- Offizielles TYPO3 Guide "TYPO3 Installation and Upgrading":
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Generelles 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

Avatare für Backend Benutzer

Backend Benutzer können nun Avatare festlegen. Diese werden in den Benutzereinstellungen gepflegt und beispielsweise oben neben dem Anmeldenamen oder in den Benutzerlisten angezeigt.

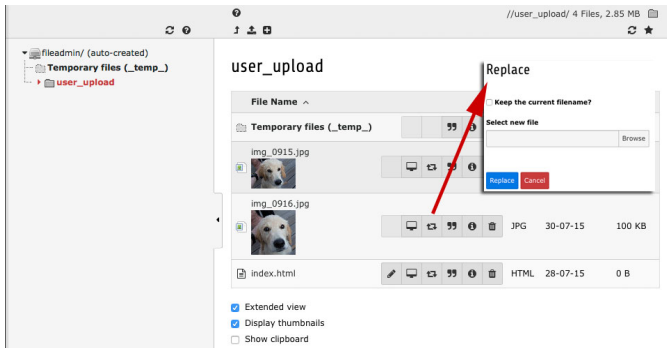


The screenshot shows the TYPO3 Backend User Listing interface. The top navigation bar includes a search icon, a lightning bolt icon, a question mark icon, a notification icon with a red '1', a user profile icon labeled 'admin', and a search input field. The main content area is titled 'Backend User Listing' and features a search bar with 'Backend users' selected. Below the search bar are filters for 'Username', 'Admin' (set to 'Both'), 'Status' (set to 'Both'), 'Login' (set to 'Both'), and 'User group'. A table displays the user listing with columns for 'Username / Real Name' and 'Last login'. The table contains one entry for the user 'admin', who is 'online' and last logged in on '30-07-15 09:58'. A red arrow points to the user profile icon in the top navigation bar, and another red arrow points to the user profile icon in the table row. The table footer shows '1 users' and a pagination control for 'Records 1 - 1' on 'Page 1' of '1'.

Backend User Interface

Dateien ersetzen

Es ist nun möglich, Dateien in der FAL Dateiliste zu **ersetzen**. Hierzu muss die "Erweiterte Ansicht" aktiviert sein. Je nach Bedarf kann der bisherige Dateinamen beibehalten oder der neue verwendet werden.



The screenshot displays the TYPO3 Backend File Administration List (FAL) interface. The left sidebar shows the file structure: fileadmin/ (auto-created) > Temporary files (_temp_) > user_upload. The main area shows the 'user_upload' folder containing three files: img_0915.jpg, img_0916.jpg, and index.html. The 'img_0916.jpg' file is selected, and a 'Replace' dialog box is open over it. The dialog box has the title 'Replace' and a checkbox 'Keep the current filename?' which is unchecked. Below this is a 'Select new file' field with a 'Browse' button. At the bottom of the dialog are 'Replace' and 'Cancel' buttons. A red arrow points from the 'Replace' button in the dialog to the 'Replace' button in the file's action menu. At the bottom of the interface, there are checkboxes for 'Extended view' (checked), 'Display thumbnails' (checked), and 'Show clipboard' (unchecked).

File Name	Size	Format	Date	Size
img_0915.jpg				
img_0916.jpg		JPG	30-07-15	100 KB
index.html		HTML	28-07-15	0 B

Backend User Interface

Onlinestatus anzeigen

Im Modul "Backend Benutzer" wird nun angezeigt, ob ein Benutzer momentan online ist.

Backend User Listing

Username Admin Status Login
User group

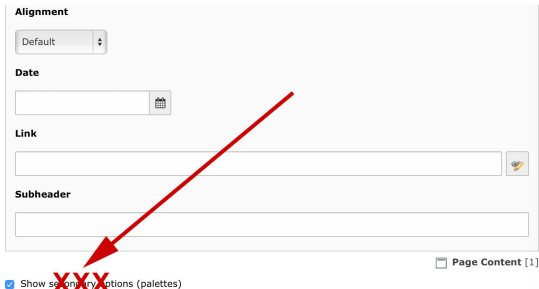
Username / Real Name	Last login	
 admin online Rocky	30-07-15 17:25	<input type="button" value="+ Compare"/> <input type="button" value="i"/> <input type="button" value="edit"/>
 admin0	Never	<input type="button" value="+ Compare"/> <input type="button" value="i"/> <input type="button" value="edit"/> <input type="button" value="toggle"/> <input type="button" value="trash"/> <input type="button" value="refresh"/>
2 Users		

⏪ ⏩ Records 1 - 2 Page / 1 ⏪ ⏩

Backend User Interface

Zweite Optionspalette entfernt

Die Checkbox "Show secondary options (palettes)" sowie die TSConfig options.enableShowPalettes und die zugehörigen TCA-Einstellungen wurden entfernt. Die "Paletten" sind nun immer sichtbar und können nicht mehr ausgeblendet werden.

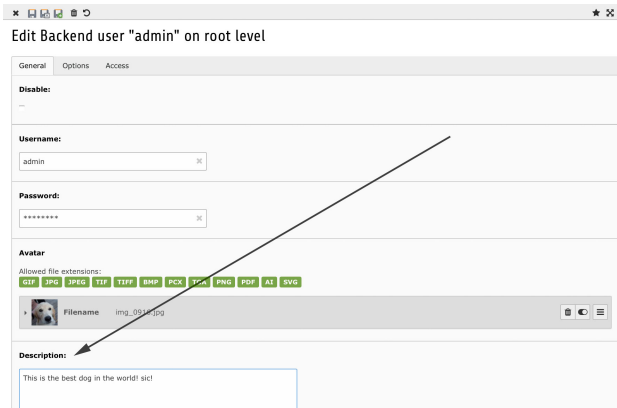


The screenshot shows a form section titled "Alignment". It contains several input fields: a dropdown menu set to "Default", a "Date" field with a calendar icon, a "Link" field with a magnifying glass icon, and a "Subheader" text area. Below the form, there is a checkbox labeled "Show secondary options (palettes)" which is checked. The text "XXX" is written in red over the checkbox label. A red arrow points from the top right towards the checkbox. In the bottom right corner of the form area, there is a small icon and the text "Page Content [1]".

Backend User Interface

Beschreibung für Backend Benutzer

Backend Benutzer können nun auch eine Beschreibung erhalten.

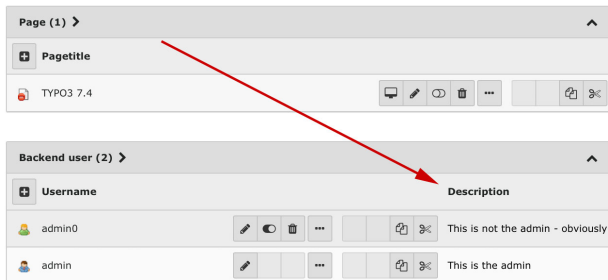


The screenshot shows the 'Edit Backend user "admin" on root level' configuration page. The page has three tabs: 'General', 'Options', and 'Access'. The 'General' tab is active. The 'Disable' field is empty. The 'Username' field contains 'admin'. The 'Password' field is masked with asterisks. The 'Avatar' section shows a list of allowed file extensions: GIF, JPG, JPEG, TIF, TIFF, BMP, PCX, TGA, PNG, PDF, AI, SVG. Below this, there is a preview of an avatar image with the filename 'img_0918.jpg'. The 'Description' field is a text area containing the text 'This is the best dog in the world! sic!'. An arrow points from the 'Description' field to the text area.

Backend User Interface

Beschreibung im Backend anzeigen

Über die TCA-Einstellung `['TCA']['ctrl']['descriptionColumn']` kann eine Spalte ausgewählt werden (meist `description`), die eine Beschreibung enthält. Ist diese vorhanden, wird der Inhalt beispielsweise im Listenmodul angezeigt.



The screenshot displays two sections of the TYPO3 Backend interface. The top section, titled "Page (1)", shows a table with one row for "Page (1)" and a "Pagetitle" column. The bottom section, titled "Backend user (2)", shows a table with two rows for "Backend user (2)". The first row is for "admin0" with a description "This is not the admin - obviously". The second row is for "admin" with a description "This is the admin". A red arrow points from the "Pagetitle" column in the top section to the "Description" column in the bottom section, illustrating the concept of a description column in the TCA configuration.

Page (1) >	
+	Pagetitle
TYPO3 7.4	

Backend user (2) >		
+	Username	Description
admin0		This is not the admin - obviously
admin		This is the admin

Backend User Interface

Beschreibung für Filemounts

Filemounts können ebenfalls eine Beschreibung erhalten.

Create new Filemount on root level

Label:

Disable:

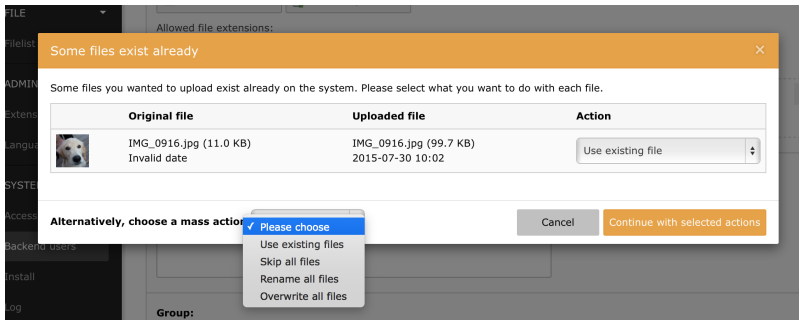
Description:

Remaining characters: 1928

Backend User Interface

Überschreiben Dialog beim Upload

Sofern bei einem Upload Dateien bereits auf dem Server existieren, werden in einem Dialog mehrere Optionen zur Auswahl angeboten.

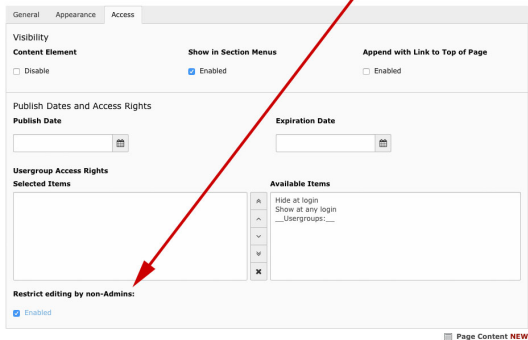


Backend User Interface

Editieren von Inhaltselementen für Nicht-Admins einschränken

Inhaltselemente können jetzt für die Bearbeitung durch Nicht-Admins eingeschränkt werden (ähnliche Funktion die es bereits bei Seiten gibt).

Create new Page Content on page "TYPO3 7.4"



The screenshot shows the 'Access' tab of the TYPO3 Backend interface for creating a new Page Content element. The interface is divided into several sections:

- Visibility:** Contains three checkboxes: 'Content Element' (unchecked), 'Show in Section Menus' (checked), and 'Append with Link to Top of Page' (unchecked).
- Publish Dates and Access Rights:** Contains two date pickers: 'Publish Date' and 'Expiration Date', both currently empty.
- Usergroup Access Rights:** Contains a 'Selected Items' list (empty) and an 'Available Items' list with the following entries: 'Hide at login', 'Show at any login', and '___Usergroups:___'. Navigation buttons (A, ^, v, W, X) are visible between the lists.
- Restrict editing by non-Admins:** A checkbox labeled 'Enabled' is checked.

A red arrow points from the top right towards the 'Restrict editing by non-Admins' checkbox. In the bottom right corner, there is a small icon and the text 'Page Content NEW'.

Backend User Interface

Statische TSconfig Dateien (1)

In den Seiteneigenschaften können nun statische TSconfig Dateien eingebunden werden.

General Access Metadata Appearance Behaviour Resources Categories

Files

Media

Create new relation Select & upload files

TypoScript Configuration

Include Page TSConfig (from extensions):

Selected Items

Available Items

Backend User Interface

Statische TSconfig Dateien (2)

Die TSconfig Dateien werden wie folgt registriert:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::registerPageTSconfigFile(  
    'extension_name',  
    'Configuration/PageTS/myPageTSconfigFile.txt',  
    'My special configuration'  
);
```

Backend User Interface

Echte Sprachkopien

Es ist nun möglich, "richtige" Kopien von Inhaltselementen in Sprachversionen anzulegen (und nicht nur Referenzen).

TYPO3 7.4

Default



Left

+ Content

Normal

+ Content

Content editor for Default language, showing a header element with the text "Header Content".

+ Content

Right

+ Content

Austrian



Austrian Page Title

Left

+ Content

Normal

+ Content

Content editor for Austrian language, showing a header element with the text "This is austrian content".

+ Content

Translate default content elements [1]

Right

+ Content

Klingon



Klingon Page Title

Left

+ Content

Normal

+ Content

Content editor for Klingon language, showing a dropdown menu for "Translate default content elements [1]" with options: "Copy from default language" and "Copy from language 'Austrian'".

Right

+ Content

Kapitel 2: TScnfig & TypoScript

Data-Provider für Backend Layouts (1)

- Backend-Layouts können jetzt per PageTScnfig definiert und damit auch in Dateien ausgelagert werden. Zum Beispiel:

```
mod {
  web_layout {
    BackendLayouts {
      exampleKey {
        title = Example
        config {
          backend_layout {
            colCount = 1
            rowCount = 2
            rows {
              1 {
                columns {
                  1 {
                    name = LLL:EXT:frontend/ ... /locallang_ttc.xlf:colPos.I.3
                    colPos = 3
                    colspan = 1
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  [...]
}
```


TScnfig & TypoScript

Erweiterung der Option `page.meta`

- Die Option `page.meta` unterstützt nun auch **Open Graph** Attributnamen

```
page {
  meta {
    X-UA-Compatible = IE=edge,chrome=1
    X-UA-Compatible.attribute = http-equiv
    keywords = TYPO3
    # <meta property="og:site_name" content="TYPO3" />
    og:site_name = TYPO3
    og:site_name.attribute = property
    description = Inspiring people to share
    og:description = Inspiring people to share
    og:description.attribute = property
    og:locale = en_GB
    og:locale.attribute = property
    og:locale:alternate {
      attribute = property
      value.1 = fr_FR
      value.2 = de_DE
    }
    refresh = 5; url=http://example.com/
    refresh.attribute = http-equiv
  }
}
```

TSconfig & TypoScript

languageField wird automatisch gesetzt

- In der TypoScript-Option `select` (die beispielsweise beim `cObject CONTENT` verwendet wird) musste man bisher das `languageField` explizit setzen
- Jenes wird nun automatisch gesetzt und kann daher weglassen werden

```
config.sys_language_uid = 2
page.10 = CONTENT
page.10 {
    table = tt_content
    select.where = colPos=0

    # Die nachfolgende Zeile ist nicht notwendig:
    #select.languageField = sys_language_uid

    renderObj = TEXT
    renderObj.field = header
    renderObj.htmlSpecialChars = 1
}
```

Individuelles Content Caching

- Es gibt nun ein individuelles Content Caching, welches im Gegensatz zu `stdWrap.cache` auch mit COA-Objekten funktioniert (ähnlich dem "Magento Block Caching")

```
page = PAGE
page.10 = COA
page.10 {
    cache.key = coaout
    cache.lifetime = 60
    #stdWrap.cache.key = coastdWrap
    #stdWrap.cache.lifetime = 60
    10 = TEXT
    10 {
        cache.key = mycurrenttimestamp
        cache.lifetime = 60
        data = date : U
        strftime = %H:%M:%S
        noTrimWrap = |10: | |
    }
}
```

```
[...]
20 = TEXT
20 {
    data = date : U
    strftime = %H:%M:%S
    noTrimWrap = |20: | |
}
}
```

Zähler für listNum

- Es gibt eine neue Eigenschaft `returnCount` für die `stdWrap` Eigenschaft `split`
- Damit kann die Anzahl der Elemente in einer kommaseparierten Liste ermittelt werden
- Das folgende Beispiel gibt 9 zurück:

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

TSconfig & TypoScript

Sortierung von Tabellen im Backend

- Über die TSconfig Option `mod.web_list.tableDisplayOrder` kann eingestellt werden, wie die Tabellen im List-Modul sortiert werden
- Dafür werden die Schlüsselworte `before` und `after` verwendet

Anwendung:

```
mod.web_list.tableDisplayOrder {
  <tableName> {
    before = <tableA>, <tableB>, ...
    after = <tableA>, <tableB>, ...
  }
}
```

Zum Beispiel:

```
mod.web_list.tableDisplayOrder {
  be_users.after = be_groups
  sys_filemounts.after = be_users
  pages_language_overlay.before = pages
  fe_users.after = fe_groups
  fe_users.before = pages
}
```

Content Language im HTTP Header

- Es wird nun standardmäßig `Content-language: XX` im HTTP Response Header an den Client gesendet, wobei "XX" dem ISO-Code entspricht, der via `sys_language_content` konfiguriert wurde
- Dabei kann `sys_language_content` unterschiedlich zu `sys_language_uid` sein, wenn der Inhalt von der Fallback-Sprache ermittelt wird
(jenes hängt von der Einstellung `sys_language_mode` ab)
- Über die Einstellung `config.disableLanguageHeader = 1` kann der Header bei Bedarf auch deaktiviert werden

TSconfig & TypoScript

Rekursive Option für ordner-basierte File Collections

- Ordner-basierte File Collections haben nun eine Option um rekursiv alle Dateien für einen gegebenen Ordner zu ermitteln
- Die Option ist ebenfalls für das TypoScript Objekt FILES verfügbar

```
filecollection = FILES
filecollection {
    folders = 1:images/
    folders.recursive = 1
    renderObj = IMAGE
    renderObj {
        file.import.data = file:current:uid
    }
}
```


TSconfig & TypeScript

Extension `.ts` für Static Templates

- Bislang waren für statische TypeScript Templates nur die folgenden Dateinamen zugelassen:
 - `constants.txt`
 - `setup.txt`
 - `include_static.txt`
 - `include_static_files.txt`
- Als Extension kann nun auch `.ts` verwendet werden
- Dabei hat `.ts` Vorrang vor `.txt`

TSconfig & TypoScript

save & view Button

- Der "save & view" Button ist nun via TSconfig konfigurierbar
- Der folgende Schlüssel nimmt eine kommaseparierte Liste an "doktypes" auf: `TCEMAIN.preview.disableButtonForDokType`
- Der Standardwert ist "254, 255, 199" (Storage Folder, Recycler und Menu Seperator)
- In Foldern und Recycler-Seiten ist der "save & view" Button daher standardmäßig nicht mehr sichtbar

TSconfig & TypoScript

stdWrap für treatIdAsReference

- Für das Objekt `getImgResource` existiert die Option `treatIdAsReference`, die ggf. definiert, dass die angegebenen UIDs als UIDs von `sys_file_reference`, anstatt von `sys_file` gelten
- Die Option `treatIdAsReference` besitzt nun `stdWrap` Funktionalität

Kapitel 3:

TSconfig & TypoScript: Data Processors

Einführung von Daten-Prozessoren

- Es wurden einige Daten-Prozessoren eingeführt, die das flexible Verarbeiten von kommaseparierten Listen erlauben
 - SplitProcessor
 - CommaSeparatedValueProcessor
 - FilesProcessor
 - GalleryProcessor
 - DatabaseQueryProcessor
- Siehe: `TYPO3\CMS\Frontend\DataProcessing`

Der Split-Processor

- Der "Split-Prozessor" zerteilt Werte in ein Array.
Dieses kann dann in einer Schleife verarbeitet werden:

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:site_default/Resources/Private/Template/Default.html
page.10.dataProcessing.2 = TYPO3\CMS\Frontend\DataProcessing\SplitProcessor
page.10.dataProcessing.2 {
    if.isTrue.field = bodytext
    delimiter = ,
    fieldName = bodytext
    removeEmptyEntries = 1
    filterIntegers = 1
    filterUnique = 1
    as = keywords
}
```

- Mögliche Anwendung in Fluid:

```
<f:for each="{keywords}" as="keyword">
  <li>Keyword: {keyword}</li>
</f:for>
```

Der CommaSeparatedValue-Prozessor (1)

- Der "CommaSeparatedValue-Prozessor" zerteilt Werte in ein zwei-dimensionales Array:

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:site_default/Resources/Private/Template/Default.html
page.10.dataProcessing.4 = TYPO3\CMS\Frontend\DataProcessing\CommaSeparatedValueProcessor
page.10.dataProcessing.4 {
    if.isTrue.field = bodytext
    fieldName = bodytext
    fieldDelimiter = |
    fieldEnclosure =
    maximumColumns = 2
    as = table
}
```

- Jenes eignet sich z.B. für CSV-Dateien oder für tt_content Datensätze vom CType "table"

Beispiel für eine Anwendung in Fluid auf der folgenden Slide

Der CommaSeparatedValue-Prozessor (2)

■ Mögliche Anwendung in Fluid:

```
<table>
  <f:for each="{table}" as="columns">
    <tr>
      <f:for each="{columns}" as="column">
        <td>
          {column}
        </td>
      </f:for>
    <tr>
  </f:for>
</table>
```


Der Files-Prozessor (1)

- Der "File Prozessor" löst Datei-Referenzen, Dateien oder Dateien innerhalb eines Ordners oder eine Collection auf, die für die Ausgabe im Frontend verwendet werden

```
tt_content.image.20 = FLUIDTEMPLATE
tt_content.image.20 {
    file = EXT:myextension/Resources/Private/Templates/ContentObjects/Image.html
    dataProcessing.10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
    dataProcessing.10 {
        references.fieldName = image
        references.table = tt_content
        files = 21,42
        collections = 13,14
        folders = 1:introduction/images/,1:introduction/posters/
        folders.recursive = 1
        sorting = description
        sorting.direction = descending
        as = myfiles
    }
}
```

Beispiel für eine Anwendung in Fluid auf der folgenden Slide

Der Files-Prozessor (2)

■ Mögliche Anwendung in Fluid:

```
<ul>
  <f:for each="{myfiles}" as="file">
    <li>
      <a href="{file.publicUrl}">{file.name}</a>
    </li>
  </f:for>
</ul>
```

Der Gallery-Prozessor

- Der "Galerie-Prozessor" berechnet beispielsweise die Größen von Assets

```
tt_content.text_media.20 = FLUIDTEMPLATE
tt_content.image.20 {
    file = EXT:myextension/Resources/Private/Templates/ContentObjects/Image.html
    dataProcessing {
        10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
        20 = TYPO3\CMS\Frontend\DataProcessing\GalleryProcessor
        20 {
            filesProcessedDataKey = files
            mediaOrientation.field = imageorient
            numberOfColumns.field = imagecols
            equalMediaHeight.field = imageheight
            equalMediaWidth.field = imagewidth
            maxGalleryWidth = 1000
            maxGalleryWidthInText = 1000
            columnSpacing = 0
            borderEnabled.field = imageborder
            borderWidth = 0
            borderPadding = 10
            as = gallery
        }
    }
}
```

Der DatabaseQuery-Prozessor (1)

■ Der "DatabaseQuery-Prozessor" spricht die Datenbank an

```
tt_content.mycontent.20 = FLUIDTEMPLATE
tt_content.mycontent.20 {
    file = EXT:myextension/Resources/Private/Templates/ContentObjects/MyContent.html
    dataProcessing.10 = TYPO3\CMS\Frontend\DataProcessing\DatabaseQueryProcessor
    dataProcessing.10 {
        if.isTrue.field = records
        table = tt_address
        colPos = 1
        pidInList = 13,14
        as = myrecords
        dataProcessing {
            10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
            10 {
                references.fieldName = image
            }
        }
    }
}
```

Beispiel für die Verarbeitung in Fluid auf der folgenden Slide

Der DatabaseQuery-Prozessor (2)

■ Mögliche Anwendung in Fluid:

```
<ul>
  <f:for each="{myrecords}" as="record">
    <li>
      <f:image image="{record.files.0}" />
      <a href="{record.data.www}">{record.data.first_name} {record.data.last_name}</a>
    </li>
  </f:for>
</ul>
```

Kapitel 4: Änderungen im System

Änderungen im System

Driver Interface

- Zum `DriverInterface` wurden die folgenden Methoden hinzugefügt:
 - `getFolderInFolder`
 - `getFileInFolder`
- Jeder eigene FAL-Driver muss daher diese beiden Methoden nachimplementieren:

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

```
public function getFileInFolder(  
    $fileName,  
    $folderIdentifier  
);
```

BREAKING CHANGE!

Änderungen im System

Unterstützung von IEC/SI-Keywords für Größen

- Die Formatierung von Größen unterstützt nun zwei Keywords, um die Einheiten festzulegen:

- `iec` (default)

(Basis: 2, Labels: | Ki | Mi | Gi | Ti | Pi | Ei | Zi | Yi)

- `si`

(Basis: 10, Labels: | k | M | G | T | P | E | Z | Y)

- Gesetzt werden kann die Formatierung z.B. via TypoScript:

```
bytes.labels = iec
```

```
echo GeneralUtility::formatSize(85123);  
// => Vorher "83.1 K"  
// => Nachher "83.13 Ki"
```


Änderungen im System

Dependency Ordering Service (1)

- Oftmals ist es notwendig eine sortierte Liste an Items zur Verfügung zu stellen, deren Einträge einerseits Abhängigkeiten haben und andererseits dazu verwendet werden, um Aktionen in eben dieser Reihenfolge auszuführen.
- Im Core findet jenes beispielsweise Verwendung bei:
 - Reihenfolge der Hook-Ausführung,
 - Ladereihenfolge von Extensions,
 - Reihenfolge der Anzeige von Menü-Einträgen,
 - usw.
- Eine Überarbeitung des bisherigen DependencyResolver stellt nun den DependencyOrderingService zur Verfügung

Änderungen im System

Dependency Ordering Service (2)

■ Anwendung:

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['someExt']['someHook'] [<some id>] = [
    'handler' => someClass::class,
    'runBefore' => [ <some other ID> ],
    'runAfter' => [ ... ],
    ...
];
```

■ Zum Beispiel:

```
$hooks = $GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['someExt']['someHook'];
$sorted = GeneralUtility::makeInstance(DependencyOrderingService::class)->orderByDependencies(
    $hooks, 'runBefore', 'runAfter'
);
```

Änderungen im System

Hooks und Signals (1)

- Ein neuer Hook wurde am Ende von `InlineRecordContainer::checkAccess` hinzugefügt, mit dem der Zugriff von Inline-Records geprüft werden kann
- Der Hook kann wie folgt registriert werden:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_tceforms_inline.php']  
  ['checkAccess'][] = 'My\\Package\\HookClass->hookMethod';
```

Änderungen im System

Hooks und Signals (2)

- Ein neuer Hook wurde am Ende von `AbstractUserAuthentication::checkAuthentication` hinzugefügt, mit dem man fehlgeschlagene Anmeldeversuche verarbeiten kann
- Standardmäßig wartet der Prozess 5 Sekunden nachdem eine Anmeldung fehlgeschlagen ist
- Über den Hook kann ein anderes Verhalten implementiert werden (z.B. zur Abwehr von Brute Force Angriffen)
- Der Hook kann wie folgt registriert werden:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_userauth.php']  
    ['postLoginFailureProcessing'][] = 'My\\Package\\HookClass->hookMethod';
```

Änderungen im System

Hooks und Signals (3)

- Das neue Signal `recordMarkedAsMissing` wird ausgesendet, wenn der FAL Indexer auf einen `sys_file` Eintrag stößt, dessen Datei im Dateisystem aber nicht auffindbar ist. Dabei wird die `sys_file` UID übermittelt.
- Jenes kann in Extensions verwendet werden, die Dienste rund um das Datei-Management anbieten (wie beispielsweise Versionierung, Synchronisation, Recovery, usw.)
- Das Signal `afterMappingSingleRow` wird ausgesendet, wann immer der `DataMapper` ein Objekt erstellt

Änderungen im System

HTML in TypoLink-Titeln

- Anführungszeichen in TypoLink-Titeln werden nun automatisch "escaped"
- Ein eventuell bereits existierendes Escaping wird daher nun falsch dargestellt:
Aus `'Some "special" title'`
wird `'Some &quot;special&quot; title'`
- Es wird empfohlen, hier auf Escaping komplett zu verzichten, da sich TYPO3 nun darum kümmert

BREAKING CHANGE!

Änderungen im System

Diverse Änderungen (1)

- Mit `Files->replace` gibt eine neue Berechtigung für Backend Benutzer, um Dateien im Modul Dateiliste zu ersetzen
 - Der Dateinamen des Logfiles, welches der FileWriter schreibt, ändert sich wie folgt:
 - bisher: `typo3temp/logs/typo3.log`
 - neu: `typo3temp/logs/typo3_<hash>.log`
- (der Wert `<hash>` berechnet sich aus dem Encryptionkey)

Änderungen im System

Diverse Änderungen (2)

- Die in Hooks verwendeten Klassen müssen ab sofort dem Autoloading-Mechanismus folgen
- Daher kann die Hook-Definition auch verkürzt werden:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['tce']['formevals']  
  [\TYPO3\CMS\Saltedpasswords\Evaluation\FrontendEvaluator::class] = '';
```

BREAKING CHANGE!

Kapitel 5: Extbase & Fluid

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!

Kapitel 6:

Veraltete und entfernte Funktionen

Veraltete/Entfernte Funktionen

Systemextension `cms` entfernt (1)

- Die Systemextension `cms` wurde entfernt
- Entwickler von Extensions sollten prüfen, ob Abhängigkeit zu `cms` in der Datei `ext_emconf.php` vorhanden sind und diese ggf. korrigieren

```
[...]
'constraints' => array(
  'depends' => array(
    // 'cms' => ' ... ',          <= FALSCH!
    'typo3' => '7.0.0-7.99.99',
  ),
),
[...]
```

- Die meiste Funktionalität wurde zur Systemextension `frontend` migriert (daher müssen ggf. Referenzen zu Sprachdateien angepasst werden, siehe folgende Slide)

Veraltete/Entfernte Funktionen

Systemextension cms entfernt (2)

■ Notwendige Anpassungen der Referenzen zu Sprachdateien:

ALT: typo3/sysext/cms/web_info/locallang.xlf

NEU: typo3/sysext/frontend/Resources/Private/Language/locallang_webinfo.xlf

ALT: typo3/sysext/cms/locallang_ttc.xlf

NEU: typo3/sysext/frontend/Resources/Private/Language/locallang_ttc.xlf

ALT: typo3/sysext/cms/locallang_tca.xlf

NEU: typo3/sysext/frontend/Resources/Private/Language/locallang_tca.xlf

ALT: typo3/sysext/cms/layout/locallang_db_new_content_el.xlf

NEU: typo3/sysext/backend/Resources/Private/Language/locallang_db_new_content_el.xlf

ALT: typo3/sysext/cms/layout/locallang.xlf

NEU: typo3/sysext/backend/Resources/Private/Language/locallang_layout.xlf

ALT: typo3/sysext/cms/layout/locallang_mod.xlf

NEU: typo3/sysext/backend/Resources/Private/Language/locallang_mod.xlf

ALT: typo3/sysext/cms/locallang_csh_webinfo.xlf

NEU: typo3/sysext/frontend/Resources/Private/Language/locallang_csh_webinfo.xlf

ALT: typo3/sysext/cms/locallang_csh_weblayout.xlf

NEU: typo3/sysext/frontend/Resources/Private/Language/locallang_csh_weblayout.xlf

Veraltete/Entfernte Funktionen

PageRenderer ist veraltet

- Die folgenden PageRenderer-Methoden wurden als veraltet deklariert:

```
TYPO3\CMS\Backend\Controller\BackendController::getPageRenderer()  
TYPO3\CMS\Backend\Template\DocumentTemplate::getPageRenderer()  
TYPO3\CMS\Backend\Template\FrontendDocumentTemplate::getPageRenderer()  
TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController::getPageRenderer()
```

- Stattdessen ist nun folgender Code zu verwenden, um eine Instanz des PageRenderers zu erhalten:

```
\TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(\TYPO3\CMS\Core\Page\PageRenderer::class)
```


Veraltete/Entfernte Funktionen

Veraltete GeneralUtility-Methoden

- Die folgenden GeneralUtility-Methoden wurden als veraltet deklariert und werden in TYPO3 CMS version 8 entfernt:

```
GeneralUtility::modifyHTMLColor()  
GeneralUtility::modifyHTMLColorAll()  
GeneralUtility::isBrokenEmailEnvironment()  
GeneralUtility::normalizeMailAddress()  
GeneralUtility::formatForTextarea()  
GeneralUtility::getThisUrl()  
GeneralUtility::cleanOutputBuffers()  
GeneralUtility::readLLfile()
```

- Methode `readLLfile()` kann durch folgenden Code ersetzt werden:

```
/** @var $languageFactory \TYPO3\CMS\Core\Localization\LocalizationFactory */  
$languageFactory = GeneralUtility::makeInstance(  
    \TYPO3\CMS\Core\Localization\LocalizationFactory::class  
);  
$languageFactory->getParsedData($fileToParse, $language, $renderCharset, $errorMode);
```

Veraltete/Entfernte Funktionen

JavaScript Bibliotheken entfernt

- Die JavaScript-Bibliotheken `prototype.js` und `scriptaculous` wurden entfernt. Somit haben die folgenden TypoScript-Eigenschaften keine Funktion mehr:
 - `page.javascriptLibs.Prototype`
 - `page.javascriptLibs.Scriptaculous.*`
- Im ViewHelper `be.container` liefern die entsprechenden Attribute Fehler:

```
<f:be.container loadPrototype="false" loadScriptaculous="false"
  scriptaculousModule="someModule,someOtherModule">
```
- Stattdessen wird empfohlen `jQuery` und `RequireJS` zu verwenden (die im Backend bereits standardmäßig geladen werden)

Veraltete/Entfernte Funktionen

`init.php`, `mod.php` und `ajax.php` sind veraltet

- Da alle nicht benötigten Dateien aus typo3 aufgeräumt werden sollen, wurden die Dateien `init.php`, `mod.php` und `ajax.php` als veraltet markiert
- Will man eigene Init Entry Points verwenden, so geht dies über den folgenden Code:

```
call_user_func(function() {  
    $classLoader = require __DIR__ . '/vendor/autoload.php';  
    (new \TYPO3\CMS\Backend\Http\Application($classLoader))->run();  
});
```

- Anstelle des Zugriffs auf `mod.php` verwendet man nun:

```
BackendUtility::getModuleUrl()
```

Veraltete/Entfernte Funktionen

TCA: Zusätzliche Palette entfernt

- Der `showitem` String des TCA-Schlüssels `types` sah die Möglichkeit vor, eine zusätzliche Palette zu definieren. Diese wurde nach dem Hauptfeld gerendert
- Jenes wurde nun entfernt und in die normale Paletten-Definition migriert
- Bisher:

```
'types' => array(  
    'aType' => array(  
        'showitem' => 'aField;aLabel;anAdditionalPaletteName',  
    ),  
),
```

- Neu:

```
'types' => array(  
    'aType' => array(  
        'showitem' => 'aField;aLabel, --palette--;;anAdditionalPaletteName',  
    ),  
).
```

Veraltete/Entfernte Funktionen

Diverse Änderungen (1)

- Die Content-Objekte "Text" und "Text mit Bild" hatten bisher eine Checkbox "RTE enabled". Diese wurde, zusammen mit der dazugehörigen TCA-Option `flag`, entfernt.
- Die folgenden TypoScript-Optionen zum Einbinden von PHP-Dateien wurden entfernt:
 - `config.includeLibrary`
 - `config.includeLibs`
- Das Verzeichnis `typo3/ext` wurden entfernt (nicht aber die Möglichkeit, globale Extensions zu verwenden: das Verzeichnis kann manuell angelegt werden)

Veraltete/Entfernte Funktionen

Diverse Änderungen (2)

- ExtCore (ein schlanker ExtJS Adapter) wurde entfernt und damit die folgenden TypeScript-Optionen:
 - `page.javascriptLibs.ExtCore.*`
 - `page.javascriptLibs.ExtJs.*`

Außerdem die entsprechende Option im `<f:be.container>`-ViewHelper

- Die sogenannten "BigButtons" ("Edit Page Properties", "Move Page",...) wurden entfernt und mit ihnen die TSconfig-Einstellung `mod.we_layout.disableBigButtons`

Veraltete/Entfernte Funktionen

Diverse Änderungen (3)

- Die Konfiguration für das Error- und Exception-Handling kann nun nicht mehr in der Datei `ext_localconf.php` der Extension überschrieben werden, sondern muss in einer der Dateien `LocalConfiguration.php` oder `AdditionalConfiguration.php` gesetzt werden.
- Das Feld "General Record Storage Page" welches die Storage-PID für die Seite aufgenommen hat, wurde entfernt. Stattdessen muss man die Storage-PID nun per TypoScript (oder FlexForm) setzen.
- Die Funktion `IconUtility::getIcon()` wurde als veraltet gekennzeichnet - stattdessen verwendet man `IconUtility::getSpriteIconForRecord()`

Kapitel 7: Quellen und Autoren

Quellen und Autoren

Quellennachweis

TYPO3 News:

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

Release Infos:

- http://wiki.typo3.org/TYPO3\CMS_7.4.0
- [INSTALL.md](#) and [Changelog](#)
- [typo3/sysex/core/Documentation/Changelog/7.4/*](http://typo3.org/sysex/core/Documentation/Changelog/7.4/*)

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

TYP03 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, Pierrick Caillon, Sergio Catala, Jigal van Hemert, Michel Mix, Sinisa Mitrovic, Angeliki Plati, Nena Jelena Radovic, Roberto Torresani

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

Lizenziert unter Creative Commons BY-NC-SA 3.0

