

# **TYPO3 CMS 7 LTS – What's New**

## **In-Depth Changes**

Created by:  
Patrick Lobacher and Michael Schams

# Inleiding

---

## TYPO3 CMS 7 LTS - What's New

De volgende pagina's richten zich op een bepaald onderwerp. Afhankelijk van je rol kunnen de volgende onderwerpen interessant voor je zijn:

	<i>Interface backend</i>	<i>TypoScript</i>	<i>Systeemwijzigingen</i>	<i>Extbase/Fluid</i>	<i>Verouderd/verwijderd</i>	<i>Sys.Beheer</i>
<b>Redacteuren</b>	X					
<b>Sitebouwers</b>		X	X		X	
<b>Ontwikkelaars</b>			X	X	X	
<b>SysAdmins</b>						X

Download alle versies van de **What's New Slides** van [typo3.org](https://typo3.org)

## Systeemwijzigingen

Door het stroomlijnen van alle onderdelen en het verwijderen van verouderde technieken is de Backend van TYPO3 CMS 7 LTS razendsnel en beter dan alle voorgaande TYPO3 versies.

Niet alleen de Backend heeft een enorme versnelling gekregen, er zijn ook vele wijzigingen "onder de motorkap" doorgevoerd. PHP code is opgeschoond, sommige componenten zijn herschreven in Extbase en veel nieuwe functies zijn ingebouwd op basis van de nieuwste technieken en programmeerstandaarden.

# Systeemwijzigingen

---

## Integratie van jQuery UI versie 1.11.2

- jQuery UI 1.11 ondersteunt AMD ([Asynchronous Module Definition](#)), die JavaScript-bestanden alleen inlaadt als ze nodig zijn (extra performance)
- jQuery UI 1.11 vervangt jQuery UI 1.10 + Scriptaculous in TYPO3 CMS 7.0
- Alleen de core- en interactiecomponenten die nodig zijn om ExtJS en Scriptaculous te vervangen zijn opgenomen
- Widgets zijn niet opgenomen (die van Twitter Bootstrap worden gebruikt, zoals: DatePicker, Spinner, Dialog, Buttons, Tabs, Tooltip)

## Register voor klassen om bestanden te renderen

- Om alle soorten mediabestanden te kunnen renderen is een register voor het renderen van bestanden gemaakt.

Dit gebeurt als volgt (bijv. Video, MPEG, AVI, WAV, etc.):

```
<?php
namespace ...;

class NameTagRenderer implements FileRendererInterface {
    protected $possibleMimeTypes = array('audio/mpeg', 'audio/wav', ...);
    public function getPriority() {
        return 1; // prioriteit: hoe hoger, des te belangrijker (max: 100)
    }
    public function canRender(FileInterface $file) {
        return in_array($file->getMimeType(), $this->possibleMimeTypes, TRUE);
    }
    public function render(FileInterface $file, $width, $height, array $options = array(),
        $usedPathsRelativeToCurrentScript = FALSE) {
        ...

        return 'HTML code';
    }
}
```

# Systeemwijzigingen

---

## TCA: E-mailadressen valideren

- Nieuwe functie "email" checkt of de waarde een geldig e-mailadres is
- Als controle niet slaagt, verschijnt een statusmelding
- Voorbeeld:

```
'emailaddress' => array(  
    'exclude' => 1,  
    'label' => 'LLL:EXT:myextension/Resources/Private/Language/locallang_db.xlf:tx_myextension  
    'config' => array(  
        'type' => 'input',  
        'size' => 30,  
        'eval' => 'email,trim'  
    ),  
)
```

## AbstractCondition voor eigen TypoScript-condities

- Eigen TypoScript-condities kunnen afgeleid worden van AbstractCondition

```
class TestCondition
    extends \TYPO3\CMS\Core\Configuration\TypoScript\ConditionMatching\AbstractCondition {

    public function matchCondition(array $conditionParameters) {
        if ($conditionParameters[0] === '= 7' && $conditionParameters[1] === '!= 6') {
            throw new TestConditionException('Alles Ok', 1411581139);
        }
    }
}
```

- De bijbehorende TypoScript-code is:

```
[Vendor\Package\TestCondition]
[Vendor\Package\TestCondition = 7]
[Vendor\Package\TestCondition = 7, != 6]
```

- Operatoren die beschikbaar moeten zijn, worden in de klasse gedefinieerd

# Systemwijzigingen

---

## Signal voor manipulatie van IconUtility HTML tag

- Nieuw signal voor manipulatie van de IconUtility HTML tag voor sprite-icoenen:

```
dispatch(  
  'TYPO3\CMS\Backend\Utility\IconUtility',  
  'buildSpriteHtmlIconTag',  
  array($tagAttributes, $innerHtml, $tagName)  
);
```

- Aanroep:

```
TYPO3\CMS\Backend\Utility\IconUtility\buildSpriteHtmlIconTag
```



# Stysteemwijzigingen

---

## Signal Slots voor SoftReferenceIndex

- Twee nieuwe signal-slot aanroepen in SoftReferenceIndex:

```
protected function emitGetTypoLinkParts(  
    $linkHandlerFound, $finalTagParts, $linkHandlerKeyword, $linkHandlerValue) {  
    return $this->getSignalSlotDispatcher()->dispatch(  
        get_class($this),  
        'getTypoLinkParts',  
        array($linkHandlerFound, $finalTagParts, $linkHandlerKeyword, $linkHandlerValue)  
    );  
}  
protected function emitSetTypoLinkPartsElement(  
    $linkHandlerFound, $tLP, $content, $elements, $idx, $tokenId) {  
    return $this->getSignalSlotDispatcher()->dispatch(  
        get_class($this),  
        'setTypoLinkPartsElement',  
        array($linkHandlerFound, $tLP, $content, $elements, $idx, $tokenId, $this)  
    );  
}
```

- Aangeropen in:

```
TYPO3\CMS\Core\Database\SoftReferenceIndex->findRef_typolink  
TYPO3\CMS\Core\Database\SoftReferenceIndex->getTypoLinkParts
```

## afterPersistObjectt Signal Slot

- Nieuwe afterPersistObject signal-slot zendt uit voor de aggregate root na het persisteren van alle andere objecten

```
protected function emitAfterPersistObjectSignal(DomainObjectInterface $object) {  
    $this->signalSlotDispatcher->dispatch(__CLASS__, 'afterPersistObject', array($object));  
}
```

- Aangeropen in:

```
TYPO3\CMS\Extbase\Persistence\Generic\Backend->persistObject
```

- Hetzelfde signal is uitgezonden in de methode persistObject in de AbstractBackend-klasse in Flow

# Stysteemwijzigingen

---

## Signal in loadBaseTca

- Om de prestaties te verbeteren in de backend, wordt de hele TCA gecachet (niet alleen delen ervan)

```
protected function emitTcaIsBeingBuiltSignal(array $tca) {  
    list($tca) = static::getSignalSlotDispatcher()->dispatch(  
        __CLASS__,  
        'tcaIsBeingBuilt',  
        array($tca)  
    );  
    $GLOBALS['TCA'] = $tca;  
}
```

- Aangeropen in:

```
TYPO3\CMS\Core\Utility\ExtensionManagementUtility\Backend->buildBaseTcaFromSingleFiles
```

# Systeemwijzigingen

---

## API om gecachete TCA-wijzigingen toe te voegen

- PHP bestanden in `extkey/Configuration/TCA/Overrides/` worden direct uitgevoerd nadat de TCA cache is gebouwd
- Deze bestanden mogen alleen code bevatten die de TCA bewerkt, zoals: `addTCAColumns` of `addToAllTCATypes`
- Dit geeft de backend een prestatieverhoging zodra extensies hiervan gebruik maken

# Systeemwijzigingen

---

## Alleen-lezen startpunten voor bestanden

- Startpunten voor bestanden kunnen (weer) ingesteld worden als "alleen-lezen"
- Dit was al mogelijk in TYPO3 CMS 4.x, maar stilletjes verwijderd in 6.x
- Voorbeeld: voeg map "test" van opslag UID 3 als een alleen-lezen startpunt in de bestandslijst en elementbladerscherm.

```
options.folderTree.altElementBrowserMountPoints = 3:/test
```

Als geen opslag is ingesteld wordt aangenomen dat de map in de standaardopslag zit.

# Systeemwijzigingen

---

## Diversen

- jQuery is bijgewerkt van versie 1.11.0 naar versie 1.11.1
- Datatabellen zijn bijgewerkt van versie 1.9.4 naar versie 1.10.2
- Enige oude, ongebruikte variabelen zijn verwijderd uit EM\_CONF
- Extensie-iconen kunnen ook in SVG-formaat zijn (`ext_icon.svg`)
- Een verkeerde eID-identificer geeft nu een exceptie

# Systeemwijzigingen

---

## TCA: Maximaal aantal karakters in tekstelement

- TCA-type `text` biedt nu ondersteuning voor het HTML5 attribuut `maxlength` om de lengte van een tekst te beperken (opmerking: geregeleinden tellen gewoonlijk als twee karakters)

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

Merk op dat niet elke browser dit attribuut ondersteunt.  
Zie [Browser Support List](#) voor details.

# Stysteemwijzigingen

---

## Nieuwe implementatie van SplFileInfo

- Nieuwe klasse: TYPO3\CMS\Core\Type\File\FileInfo
- Deze klasse is een uitbreiding van klasse SplFileInfo die het mogelijk maakt om meta-informatie van bestanden op te halen

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

- Custom implementaties kunnen de volgende hook gebruiken:

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



# Systeemwijzigingen

---

## userFunc in de TCA Display Condition

- Een userFunc in de displayCondition maakt het mogelijk om op elke denkbare conditie of toestand te controleren
- Wanneer een situatie niet kan worden afgevangen met één van de bestaande controles, kunnen ontwikkelaars hun eigen gebruikerfuncties ontwikkelen (return TRUE/FALSE om het daarvoor bestemde veld te tonen/verbergen)

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

# Systeemwijzigingen

---

## API voor Twitter Bootstrap Modals (1)

- Twee nieuwe API-methoden om popup modals te creëren/verbergen:
  - `TYPO3.Modal.confirm(title, content, severity, buttons)`
  - `TYPO3.Modal.dismiss()`
- De opties `title` en `content` zijn verplicht
- Wanneer `buttons` wordt gebruikt, zijn ook de opties `buttons.text` en `buttons.trigger` verplicht
- Voorbeeld 1:

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

# Systeemwijzigingen

---

## API voor Twitter Bootstrap Modals (2)

### ■ Voorbeeld 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();
            }
        }
    ]
);
```

# Systeemwijzigingen

---

## API voor JavaScript Storage (1)

- Toegang tot de BE-user configuratie (`$BE_USER->uc`) kan worden afgehandeld in JavaScript door simpele sleutel/waarde-paren te gebruiken
- Bovendien kan HTML5's `localStorage` worden gebruikt om gegevens (client-side) op te slaan in de browser van de gebruiker
- Twee nieuwe globale TYPO3 objecten:
  - `top.TYPO3.Storage.Client`
  - `top.TYPO3.Storage.Persistent`
- Elk object heeft de volgende API-methoden:
  - `get(key)`: haal gegevens op
  - `set(key, value)`: schrijf gegevens weg
  - `isset(key)`: controleer of de sleutel bestaat
  - `clear()`: leeg de gehele storage

## API voor JavaScript Storage (2)

### ■ Voorbeeld:

```
// 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');
```

# Systeemwijzigingen

---

## Gealigneerde weergave van selectievakjes

- Bij selectievakjes kan bij 'cols' de instelling `inline` worden gebruikt om selectievakjes naast elkaar weer te geven om de gebruikte ruimte te beperken

```
'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'  
  )  
)  
),  
...
```

# Stysteemwijzigingen

---

## Registratie van Content Object

- Er is een nieuwe globale optie geïntroduceerd om cObjecten zoals TEXT te registreren, uit te breiden en/of te overschrijven
- Een lijst van alle beschikbare cObjecten is beschikbaar middels:  
`$GLOBALS['TYPO3_CONF_VARS']['FE']['ContentObjects']`
- Voorbeeld: registreer een nieuw cObject EXAMPLE  
`$GLOBALS['TYPO3_CONF_VARS']['FE']['ContentObjects']['EXAMPLE'] = Vendor\MyExtension\ContentObject\ExampleContentObject::class;`
- De geregistreerde klasse moet een subklasse zijn van  
`TYPO3\CMS\Frontend\ContentObject\AbstractContentObject`
- Bewaar je klasse in map  
`typo3conf/myextension/Classes/ContentObject/`  
om voorbereid te zijn op toekomstige autoloadmecanismen

# Systeemwijzigingen

---

## Hooks en Signals (1)

- Er is een nieuwe hook toegevoerd aan het einde van `PageRepository->init()`, die het mogelijk maakt om de zichtbaarheid van pagina's te beïnvloeden

- Registreer de hook als volgt:

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

- De hookklasse moet de volgende interface implementeren:

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



# Systeemwijzigingen

---

## Hooks en Signals (2)

- Er is een nieuwe hook toegevoegd aan de PageLayoutView om de weergave van de onderkant van contentelementen te bewerken.

- Voorbeeld:

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

- De hookklasse moet de volgende interface implementeren:

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

# Systeemwijzigingen

---

## Hooks en Signals (3)

- Er is een nieuwe hook toegevoegd als een post-processor van `BackendUtility::countVersionsOfRecordsOnPage`
- Dit kan bijvoorbeeld worden gebruikt om de toestand van de workspace in de paginaboom zichtbaar te maken
- Registreer de hook als volgt:

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

# Systeemwijzigingen

---

## Hooks en Signals (4)

- Een nieuw signal is toegevoegd aan het eind van de methode `DataPreprocessor::fetchRecord()`
- Dit kan bijvoorbeeld worden gebruikt om de array `regTableItems_data` te bewerken, teneinde gemanipuleerde data in `TCEForms` weer te geven

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

# Systeemwijzigingen

---

## Hooks en Signals (5)

- Er is een nieuw signal toegevoegd die extra bewerking mogelijk maakt na initialisatie van een mailer object, bijvoorbeeld de registratie van een Swift Mailer-plugin

```
$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'  
);
```

# Systemwijzigingen

---

## Meerdere UID's in PageRepository::getMenu()

- De methode PageRepository::getMenu() accepteert nu arrays om meerdere rootpagina's te kunnen definiëren

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

# Wijzigingen nader bekeken

---

## SVG-ondersteuning in core

- TYPO3 CMS Core ondersteunt SVG-afbeeldingen ("Scalable Vector Graphics")
- Als een SVG-afbeelding wordt geschaald wordt een record met de nieuwe maten in `sys_file_processedfile` opgeslagen en geen nieuw bestand aangemaakt (behalve als de afbeelding verder bewerkt wordt, bijv. afgesneden).
- Als ImageMagick/GraphicsMagick de maten niet kan vaststellen wordt teruggevallen op het uitlezen van de XML.
- SVG is ook toegevoegd aan de lijst met geldige afbeeldingsbestanden: `$GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext']`

# Wijzigingen nader bekeken

---

## Uitbreiding FAL-driver

- Om de prestaties van de bestandslijst te verbeteren bij gebruik van opslag op niet-lokale systemen moet de FAL-driver zorgen voor sorteren en het vaststellen van het aantal bestanden/mappen. Twee nieuwe parameters `sort` en `sortRev` zijn toegevoegd hiervoor:

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

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

- Daarnaast zijn er twee nieuwe functies:

```
public function getFilesInFolderCount($folderIdentifier, $recursive = FALSE,
    array $filenameFilterCallbacks = array());
```

```
public function getFoldersInFolderCount($folderIdentifier, $recursive = FALSE,
    array $folderNameFilterCallbacks = array());
```

# Wijzigingen nader bekeken

---

## API Backend-routing (1)

- Een API voor backend routing is toegevoegd die de toegangspunten voor de backend afhandelt.
- De API is voor een groot deel compatible met de Symfony Routing Framework, waarop het is gebaseerd (TYPO3 gebruikt momenteel ca. 20%)
- Drie klassen zorgen voor de functionaliteit:
  - `class Route`: bevat details over paden en opties
  - `class Router`: API om de route te detecteren
  - `class UrlGenerator`: bouwt de URL op



# Wijzigingen nader bekeken

---

## API Backend-routering (2)

- Routes worden gedefinieerd in het volgende bestand of extensie:  
`Configuration/Backend/Routes.php`  
(zie systeemextensie `backend` als voorbeeld)
- Verdere details over de API voor Backend-routering:  
<http://wiki.typo3.org/Blueprints/BackendRouting>

# Wijzigingen nader bekeken

---

## Nieuwe systeemextensie voor Media Contentelementen

- Nieuwe systeemextensie "mediace" bevat de volgend cObjects:
  - MULTIMEDIA
  - MEDIA
  - SWFOBJECT
  - FLOWPLAYER
  - QTOBJECT
- Contentelementen `media` en `multimedia` zijn verplaatst naar deze systeemextensie, net zoals de "Media Wizard Provider"
- Deze extensie is **niet** standaard geïnstalleerd!

# Wijzigingen nader bekeken

---

## Locatie van bibliotheken van derden

- Bibliotheken van derden die door composer geïnstalleerd zijn staan onder `typo3/contrib/vendor` (TYPO3 CMS < 7.2: in map `Packages/Libraries`)
- Hierdoor kan het inpakproces voor de publicatie van TYPO3 CMS als tar of zip een compleet werkende installatie samenstellen zonder `Packages/` voor bibliotheken van derden mee te hoeven leveren
- Er kunnen problemen ontstaan bij installaties die via composer opgezet zijn en `phpunit` gebruiken tenzij composer afhankelijkheden opnieuw worden opgebouwd. Om dit op te lossen:

```
# cd htdocs/  
# rm -rf typo3/contrib/vendor/ bin/ Packages/Libraries/ composer.lock  
# composer install
```

# Wijzigingen nader bekeken

---

## JavaScript Notificaties

- Een nieuwe JavaScript API voor Notificaties API is beschikbaar:

```
// oud en deprecated:
```

```
top.TYPO3.Flashmessages.display(TYPO3.Severity.notice)
```

```
// nieuw en de enige juiste manier sinds TYPO3 CMS 7.2:
```

```
top.TYPO3.Notification.notice(title, message)
```

- Overzicht API functies:

(parameter duur is optioneel en is standaard 5 seconde)

- `top.TYPO3.Notification.notice(titel, bericht, duur)`
- `top.TYPO3.Notification.info(titel, bericht, duur)`
- `top.TYPO3.Notification.success(titel, bericht, duur)`
- `top.TYPO3.Notification.warning(titel, bericht, duur)`
- `top.TYPO3.Notification.error(titel, bericht, duur)`

# Wijzigingen nader bekeken

---

## Uitklapscherm systeeminformatie (1)

- Extra systeeminformatie kan worden toegevoegd door een slot aan te maken
- Het slot moet geregistreerd worden in het bestand `ext_localconf.php`:

```
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class);  
  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Backend\Backend\ToolBarItems\SystemInformationToolBarItem::class,  
    'getSystemInformation',  
    \Vendor\Extension\SystemInformation\Item::class,  
    'getItem'  
);
```

# Wijzigingen nader bekeken

---

## Uitklapscherm systeeminformatie (2)

- Extra systeeminformatie kan worden toegevoegd door een slot aan te maken
- Hiervoor is de klasse `Item` en de functie `getItem()` nodig in bestand `EXT:extension\Classes\SystemInformation\Item.php`:

```
class Item {
    public function getItem() {
        return array(array(
            'title' => 'Titel als muis erboven staat',
            'value' => 'Beschrijving voor de lijstt',
            'status' => SystemInformationHookInterface::STATUS_OK,
            'count' => 4,
            'icon' => \TYPO3\CMS\Backend\Utility\IconUtility::getSpriteIcon(
                'extensions-example-information')
        ));
    }
}
```

# Wijzigingen nader bekeken

---

## Uitklapscherm systeeminformatie (3)

- Icoon extensions-example-information-icon moet geregistreerd worden in `ext_localconf.php`:

```
\TYPO3\CMS\Backend\Sprite\SpriteManager::addSingleIcons(  
    array(  
        'information-icon' => \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extRelPath(  
            $_EXTKEY) . 'Resources/Public/Images/Icons/information-icon.png'  
        ),  
        $_EXTKEY  
    );
```

# Wijzigingen nader bekeken

---

## Uitklapscherm systeeminformatie (4)

- Berichten worden getoond onderaan het uitklapscherm
- Extensies kunnen een eigen slot hebben om de berichten te vullen:

```
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class);  
  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Backend\Backend\ToolBarItems\SystemInformationToolBarItem::class,  
    'loadMessages',  
    \Vendor\Extension\SystemInformation\Message::class,  
    'getMessage'  
);
```



# Wijzigingen nader bekeken

---

## Uitklapscherm systeeminformatie (5)

- Berichten worden getoond onderaan het uitklapscherm
- Dit vereist de klasse Message met de functie getMessage() in bestand EXT:extension\Classes\SystemInformation\Message.php:

```
class Message {  
    public function getMessage() {  
        return array(array(  
            'status' => SystemInformationHookInterface::STATUS_OK,  
            'text' => 'Er ging iets fout. In de reports module is meer informatie.'  
        ));  
    }  
}
```

# Wijzigingen nader bekeken

---

## Configuratieopties voor het bewerken van afbeeldingen (1)

### ■ Nieuwe TypeScript configuratieopties:

```
# afsnijden uitschakelen voor alle afbeeldingen  
tt_content.image.20.1.file.crop =
```

```
# afsnijden overschrijven/instellen voor alle afbeeldingen  
# offsetX,offsetY,width,height  
tt_content.image.20.1.file.crop = 50,50,100,100
```

### ■ Fluid heeft ook ondersteuning voor afsnijdfuncties:

```
# afsnijden uitschakelen voor alle afbeeldingen  
<f:image image="{imageObject}" crop="" ></f:image>
```

```
# afsnijden overschrijven/instellen voor alle afbeeldingen  
# offsetX,offsetY,width,height  
<f:image image="{imageObject}" crop="50,50,100,100" ></f:image>
```

# Wijzigingen nader bekeken

---

## Configuratieopties voor het bewerken van afbeeldingen (2)

- TCA kent ook de afsnijdfunctionaliteit:
  - Column Type: image\_manipulation
  - Config file\_field: string (default: uid\_local)
  - Config enableZoom: boolean (default: FALSE)
  - Config allowedExtensions: string (default: \$GLOBALS['TYPO3\_CONF\_VARS']['GFX']['imagefile\_ext'])
  - Config ratios: array, default:

```
array(  
    '1.7777777777777777' => '16:9',  
    '1.3333333333333333' => '4:3',  
    '1' => '1:1',  
    'NaN' => 'Free'  
)
```

# Wijzigingen nader bekeken

---

## Extra parameters voor HTMLparser userFunc

- Extra parameters kunnen doorgegeven worden aan een userFunc van de HTMLparser:

```
myobj = TEXT
myobj.value = <a href="/" class="myclass">MyText</a>
myobj.HTMLparser.tags.a.fixAttrib.class {
    userFunc = Tx\MyExt\Myclass->htmlUserFunc
    userFunc.myparam = test
}
```

- In een extensie zijn deze parameters te bereiken via:

```
function htmlUserFunc(array $params, HtmlParser $htmlParser) {
    // $params['attributeValue'] bevat de waarde van het attribuut "myclass"
    // $params['myparam'] bevat "test" in dit voorbeeld
    ...
}
```

# Wijzigingen nader bekeken

---

## API voor vergrendelingen (1)

- Nieuwe API voor vergrendelingen is gemaakt die diverse blokkeringsmethodes bevat (SimpleFile, Semaphore, ...)
- Een blokkeringsmethode moet LockingStrategyInterfaceimplementeren:

```
$lockFactory = GeneralUtility::makeInstance(LockFactory::class);  
$locker = $lockFactory->createLocker('someId');  
$locker->acquire() || die('Geen vergrendeling mogelijk.');
```

...

```
$locker->release();
```

# Wijzigingen nader bekeken

---

## API voor vergrendelingen (2)

- Sommige functies ondersteunen ook niet-blokkerende vergrendelingen:

```
$lockFactory = GeneralUtility::makeInstance(LockFactory::class);
$locker = $lockFactory->createLocker(
    'someId',
    LockingStrategyInterface::LOCK_CAPABILITY_SHARED |
    LockingStrategyInterface::LOCK_CAPABILITY_NOBLOCK
);
try {
    $result = $locker->acquire(LockingStrategyInterface::LOCK_CAPABILITY_SHARED |
        LockingStrategyInterface::LOCK_CAPABILITY_NOBLOCK);
    catch (\RuntimeException $e) {
        if ($e->getCode() === 1428700748) {
            // een proces bezit de vergrendeling
            // laten we ondertussen iets anders doen
            ...
        }
    }
    if ($result) {
        $locker->release();
    }
}
```

# Wijzigingen nader bekeken

---

## Signaal installeren van extensie

- Een nieuw signaal is beschikbaar in methode

`\TYPO3\CMS\Extensionmanager\Utility\InstallUtility::install()` die wordt verstuurd zodra een extensie is geïnstalleerd en alle imports/updates zijn afgerond

```
// execution
$this->emitAfterExtensionInstallSignal($extensionKey);

// methode
protected function emitAfterExtensionInstallSignal($extensionKey) {
    $this->signalSlotDispatcher->dispatch(
        __CLASS__,
        'afterExtensionInstall',
        array($extensionKey, $this)
    );
}
```

# Wijzigingen nader bekeken

---

## Registry voor uitlezen van tekst (1)

- Meerdere tekstlezers kunnen geregistreerd worden om diverse bestandstypes (bijv. Office, PDF-bestanden, etc.) af te handelen
- TYPO3 core heeft een lezer voor gewone tekstbestanden
- Elke geregistreerde tekstlezer moet `TextExtractorInterface` implementeren
- ...en de volgende functies:
  - `canExtractText()`  
bepaalt of het uitlezen van het bestand mogelijk is
  - `extractText()`  
geeft de inhoud van het bestand terug als tekst



# Wijzigingen nader bekeken

---

## Registry voor uitlezen van tekst (2)

- Registratie tekstuitlezers in bestand `ext_localconf.php`:

```
$textExtractorRegistry = \TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::
    getInstance();
$textExtractorRegistry->registerTextExtractor(
    \TYPO3\CMS\Core\Resource\TextExtraction\PlainTextExtractor::class
);
```

- Gebruik als volgt:

```
$textExtractorRegistry = \TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::
    getInstance();
$extractor = $textExtractorRegistry->getTextExtractor($file);
if($extractor !== NULL) {
    $content = $extractor->extractText($file);
}
```

# Wijzigingen nader bekeken

---

## Allerlei

- Web-bibliotheken (zoals Twitter Bootstrap, jQuery, Font Awesome, etc.) gebruiken "Bower" (<http://bower.io>) en zitten niet meer in de TYPO3 core Git repository

```
# bower install      voert installatie uit
# bower update       voert update uit
(bestand bower.json zit in map Build/)
```
- Taakplanner CLI heeft optie "-s" om een lopende taak te stoppen
- De map voor het bewerken van een niet-lokale opslag kan zich buiten de opslag bevinden (handig voor een alleen-legen opslag)
- De pagina ID van de oorspronkelijk opgevraagde pagina is uit te lezen: `$TSFE->getRequestedId()`

# Wijzigingen nader bekeken

---

## Symfony/Console integratie in de CommandController (1)

De CommandController maakt nu intern gebruik van Symfony/Console en biedt diverse functies:

- 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)`

# Wijzigingen nader bekeken

---

## Symfony/Console integratie in de CommandController (2)

- ProgressHelper

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

(zie volgende pagina's voor voorbeelden)

# Wijzigingen nader bekeken

---

## Symfony/Console integratie in de CommandController (3)

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

class MyCommandController extends CommandController {
    public function myCommand() {

        // render a table
        $this->output->outputTable(array(
            array('Bob', 34, 'm'),
            array('Sally', 21, 'v'),
            array('Blake', 56, 'm')
        ),
        array('Naam', 'Leeftijd', 'Geslacht'));

        // select
        $colors = array('rood', 'blauw', 'geel');
        $selectedColorIndex = $this->output->select('Kies een kleur', $colors, 'red');
        $this->outputLine('Gekozen kleur: %s.', array($colors[$selectedColorIndex]));

        [...]
    }
}
```

# Wijzigingen nader bekeken

---

## Symfony/Console integratie in de CommandController (4)

```
[...]
// vraag
$name = $this->output->ask('Wat is je naam?' . PHP_EOL, 'Bob', array('Bob', 'Sally', 'Blake'));
$this->outputLine('Hallo %s.', array($name));

// bevestiging
$likesDogs = $this->output->askConfirmation('Hou je van honden?');
if ($likesDogs) {
    $this->outputLine('Je houdt van honden!');
}

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

# Wijzigingen nader bekeken

---

## Backend aanmelding API (1)

- De aanmelding voor de Backend is compleet verbouwd en heeft een nieuwe API
- Het OpenID-formulier is eruit gehaald en gebruikt de nieuwe API (nu onafhankelijk van de klassen uit de core)
- Het concept van de backend aanmelding is gebaseerd op "login providers", die als volgt geregistreerd worden in `ext_localconf.php`:

```
$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'  
];
```

# Wijzigingen nader bekeken

---

## Backend aanmelding API (2)

- Opties zijn als volgt:
  - provider:  
login provider klassenaam, die moet implementeren:  
`TYPO3\CMS\Backend\LoginProvider>LoginProviderInterface`
  - sorting:  
volgorde van de links naar de beschikbare login providers op het aanmeldscherm
  - icon-class:  
naam van het font-awesome icoon voor de link naar het aanmeldscherm
  - label:  
label voor de link naar de login provider op het aanmeldscherm



# Wijzigingen nader bekeken

---

## Backend aanmelding API (3)

- De `LoginProviderInterface` bevat slechts de functie  

```
public function render(StandaloneView $view, PageRenderer $pageRenderer, LoginController $loginController);
```
- Parameters zijn als volgt gedefinieerd:
  - `$view`:  
Fluid `StandaloneView` die het aanmeldscherm rendert. Het sjabloonbestand moet ingesteld worden en optioneel kunnen variabelen toegevoegd worden.
  - `$pageRenderer`:  
`PageRenderer` instantie geeft mogelijkheid om JavaScript toe te voegen.
  - `$loginController`:  
`LoginController` instantie.

# Wijzigingen nader bekeken

---

## Backend aanmelding API (4)

- Sjabloon moet `<f:layout name="Login">` en `<f:section name="loginFormFields">` (voor formulervelden) bevatten:

```
<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')}"/>
      <div class="input-group-addon">
        <span class="fa fa-openid"></span>
      </div>
    </div>
  </div>
</f:section>
```

# Wijzigingen nader bekeken

---

## CategoryRegistry met nieuwe opties

- Functie CategoryRegistry->addTcaColumn heeft extra opties om l10n\_mode en l10n\_display in te stellen:

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

# Wijzigingen nader bekeken

---

## Sprites in Backend modules

- Backend modules (hoofdmodules zoals "Web" maar ook submodules zoals "Bestandslijst") kunnen nu sprites gebruiken als iconen (alleen sprite-iconen die bekend zijn bij TYPO3 zijn beschikbaar!)
- Example:

```
\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',  
        ),  
    ),  
);
```

# Wijzigingen nader bekeken

---

## FormEngine NodeFactory API (1)

- Het is nu mogelijk om nieuwe knooppunten te registreren en bestaande knooppunten te overschrijven

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

- Voorbeeld hierboven registreert een nieuwe klasse `MyVendor\MyExtension\Form\Element\T3editorElement` voor het renderen van TCA-type `input`, die de interface `TYPO3\CMS\Backend\Form\NodeInterface` moet implementeren
- De index van de array is de Unix timestamp van het moment dat het element is toegevoegd

# Wijzigingen nader bekeken

---

## FormEngine NodeFactory API (2)

- Als er meerdere registry-elementen zijn geregistreerd voor hetzelfde type, wordt degene met de hoogste prioriteit (0 to 100) gebruikt
- A nieuw TCA type kan geregistreerd worden met:

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
);
```

# Wijzigingen nader bekeken

---

## Signaal `postProcessMirrorUrl`

- Het Signaal `postProcessMirrorUrl` is naar een nieuwe klasse verplaatst

### **BREAKING CHANGE!**

- Dit voorbeeld houdt rekening met de TYPO3-versie:

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

# Wijzigingen nader bekeken

---

## Interface Stuurprogramma

- De volgende functies zijn toegevoegd aan de DriverInterface:
  - getFolderInFolder
  - getFileInFolder
- Elk FAL-stuurprogramma moet deze nieuwe functies bevatten:

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

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

**BREAKING CHANGE!**



# Wijzigingen nader bekeken

---

## IEC/SI ondersteuning in formattering bestandsgrootte

- Formattering van bestandsgrootte ondersteunt twee extra opties met een lijst labels:
  - `iec` (standaard)  
(macht van 2, labels: | Ki | Mi | Gi | Ti | Pi | Ei | Zi | Yi)
  - `si`  
(macht van 10, labels: | k | M | G | T | P | E | Z | Y)
- Voorbeeldinstelling voor formattering in TYPOscript:  
`bytes.labels = iec`

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

# Wijzigingen nader bekeken

---

## Service voor volgorde van afhankelijkheden (1)

- Vaak is het nodig om een geordende lijst te maken van "dependencies". De lijst wordt gebruikt om acties in die volgorde uit te voeren.
- Voorbeelden voor gebruik in de TYPO3 core:
  - volgorde uitvoeren van hooks,
  - volgorde laden van extensies,
  - lijst menu-items,
  - etc.
- De `DependencyResolver` is herbouwd en bevat nu een `DependencyOrderingService`

# Wijzigingen nader bekeken

---

## Service voor volgorde van afhankelijkheden (2)

### ■ Gebruik:

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

### ■ Voorbeeld:

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

# Wijzigingen nader bekeken

---

## Hooks en Signals (1)

- Hook voor post-proces `InlineRecordContainer::checkAccess` resultaten is toegevoegd
- `InlineRecordContainer::checkAccess` kan gebruikt worden om toegang te regelen voor gerelateerde inline records
- De volgende code registreert de hook:

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

# Wijzigingen nader bekeken

---

## Hooks en Signals (2)

- Hook voor post-process van mislukte aanmeldingen in `AbstractUserAuthentication::checkAuthentication` is toegevoegd
- Proces stopt standaard gedurende 5 sec bij een mislukte aanmelding
- Met deze hook kunnen alternatieve oplossingen gemaakt worden (bijv. ter voorkoming van brute force aanvallen)
- De volgende code registreert de hook:

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

# Wijzigingen nader bekeken

---

## Hooks en Signals (3)

- Nieuw signal `recordMarkedAsMissing` wordt verzonden als de FAL indexer een `sys_file` record vindt dat geen overeenkomstig bestand heeft en markeert het als vermist. Het signaal geeft het record UID van `sys_file` door.
- Dit is handig in extensies die bestandsbeheer bieden of uitbreiden, zoals versiebeheer, synchronisatie, herstel, enz.
- Signaal `afterMappingSingleRow` wordt verzonden als de `DataMapper` een object aanmaakt

# Wijzigingen nader bekeken

---

## HTML in titels in TypoLink

- Aanhalingstekens in TypoLink titels worden automatisch *escaped*
- Dit betekent dat installaties waar HTML code handmatig escaped wordt incorrect output tonen bij TYPO CMS 7.4  
Voor:       'Een &quot;speciale&quot; titel'  
Wordt:       'Een &amp;quot;speciale&amp;quot; titel'
- Aanbevolen wordt om escape-acties uit te schakelen omdat TYPO3 nu zorgt voor het escaperen van HTML in TypoLink titels

**BREAKING CHANGE!**

# Wijzigingen nader bekeken

---

## Diversen (1)

- Via de rechten van een Backendgebruiker Files->replace kan het een gebruiker toegestaan of verboden worden om bestanden in de Bestandslijst module te vervangen
  - Een hash wordt gebruikt in de bestandsnaam van bestanden die FileWriter maakt als er geen ander logbestand is geconfigureerd
    - voor: `typo3temp/logs/typo3.log`
    - na: `typo3temp/logs/typo3_<hash>.log`
- (waarde van <hash> wordt gebaseerd op de encryptiesleutel)



# Wijzigingen nader bekeken

---

## Diversen (2)

- Klassen gebruikt in hooks moeten het autoload-mechanisme volgen
- Daarom kan de hook-definitie ingekort worden:

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

**BREAKING CHANGE!**

# Systeemwijzigingen

---

## Contentelementen gebaseerd op Fluid (1)

- De nieuwe systeemextensie "**Fluid-based Content Elements**" is geïmplementeerd
- Hierbij worden i.p.v. TypoScript nu Fluid-sjablonen gebruikt voor de weergave van contentelementen
- Dat kan op termijn een alternatief worden voor *CSS Styled Content*
- Voeg de volgende statische sjablonen toe om hiervan gebruik te maken:
  - Content Elements (`fluid_styled_content`)
  - Content Elements CSS (optional) (`fluid_styled_content`)

## Contentelementen gebaseerd op Fluid (2)

- Ook moet de volgende PageTSconfig-template worden toegevoegd aan de pagina-eigenschappen:

Fluid-based Content Elements (fluid\_styled\_content)

- De paden naar de sjablonen kunnen met TypoScript worden gewijzigd:

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

## Contentelementen gebaseerd op Fluid (3)

- Migratie van *CSS Styled Content* naar *Fluid-based Content Elements*:
  - Deactiveer extensie `css_styled_content`
  - Activeer extensie `fluid_styled_content`
  - Gebruik de Upgrade Wizard in de Installatie-module om de contentelementen `text`, `image` en `textpic` naar `textmedia` te migreren

Let op: De ontwikkeling van "*Fluid-based Content Elements*" bevindt zich nog in een vroeg stadium en mogelijk werkt de extensie tot TYPO3 CMS 7 LTS nog niet goed. Ook is het nog mogelijk dat er conflicten zijn met *CSS Styled Content*.

# Systeemwijzigingen

---

## Methode `SELECTmmQuery`

- De nieuwe methode `SELECT_mm_query` is toegevoegd aan klasse `DatabaseConnection`
- Gekopieerd uit `exec_SELECT_mm_query` om het bouwen en uitvoeren van M:M-query's van elkaar te scheiden.
- Dit maakt het mogelijk om in de database abstraction layer een query op te bouwen

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

# Systeemwijzigingen

---

## Optimaliseren van MySQL-databasetabellen

- Aan de scheduler is een nieuwe taak toegevoegd die het MySQL-commando `OPTIMIZE TABLE` uitvoert op de geselecteerde databasetabellen
- Dit commando ordent de fysieke opslag van de database en de bijbehorende index om de benodigde opslagruimte te reduceren en de I/O-efficiëntie te verbeteren
- De volgende type tabellen worden ondersteund:  
MyISAM, InnoDB en ARCHIVE
- Het uitvoeren van de taak op DBAL en andere databases wordt niet ondersteund aangezien het commando alleen werkt in MySQL.

**Let op:** het optimaliseren van tabellen is een intensief I/O-proces. Bovendien worden in MySQL < 5.6.17 de tabellen gelocked, wat invloed kan hebben op de website.

# Systeemwijzigingen

---

## Ondersteuning van online media (1)

- Standaard worden nu externe online media ondersteund
- Zo is er bijvoorbeeld ondersteuning ingebouwd voor YouTube- en Vimeovideo's
- Aan het contentelement "**Text & Media**" kunnen URL's worden toegevoegd
- Een bijbehorende helper-klasse haalt de metadata op en zorgt dat er (indien beschikbaar) een voorafbeelding wordt getoond

# Systemwijzigingen

---

## Ondersteuning van online media (2)

Voor de URL's worden de volgende schrijfwijzen ondersteund:

### YouTube:

`youtu.be/<code>`

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

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

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

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

### Vimeo:

`vimeo.com/<code>`

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



# Systeemwijzigingen

---

## Ondersteuning van online media (3)

- In Fluid kunnen de elementen op de volgende manier worden gebruikt:

```
<!-- gebruik js api en no-cookie-support voor YouTube-video's -->  
<f:media file="{file}" additionalConfig="{enablejsapi:1, 'no-cookie': true}" ></f:media>
```

```
<!-- toon titel en uploader van YouTube- of Vimeo-video voor het afspelen -->  
<f:media file="{file}" additionalConfig="{showinfo:1}" ></f:media>
```

- Mogelijke instellingen voor YouTube-video's:  
autoplay, controls, loop, enablejsapi, showinfo, no-cookie
- Mogelijke instellingen voor Vimeo-video's:  
autoplay, loop, showinfo

## Ondersteuning van online media (4)

- Om gebruik te maken van een eigen online mediaservice, is er een `OnlineMediaHelper`-klasse nodig die de `OnlineMediaHelperInterface` implementeert en een `FileRenderer`-klasse die de `FileRendererInterface` implementeert.

```
// registreer een eigen online videoservice (de gebruikte sleutel is de extensienaam)
$GLOBALS['TYPO3_CONF_VARS']['SYS']['OnlineMediaHelpers']['myvideo'] =
    \MyCompany\Myextension\Helpers\MyVideoHelper::class;

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

// registreer een eigen mime-type voor video's
$GLOBALS['TYPO3_CONF_VARS']['SYS']['FileInfo']['fileExtensionToMimeType']['myvideo'] =
    'video/myvideo';

// registreer een eigen bestandsextensie als toegestaan mediabestand
$GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] .= ',myvideo';
```

# Stysteemwijzigingen

---

## Backend Routing

- Er is een nieuwe routingcomponent toegevoegd aan de TYPO3-backend die de aanroepen van de verschillende modulen binnen TYPO3 CMS afhandelt
- Routes kunnen worden gedefinieerd in de volgende klasse:

Configuration/Backend/Routes.php

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

- De objecten die aan de methode worden meegegeven zijn conform de PSR-7-standaard:

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

# Systeemwijzigingen

---

## Autoload-definities in `ext_emconf.php`

- Extensies mogen nu meerdere PSR-4-definities toevoegen aan `ext_emconf.php`
- Dit was al mogelijk in `composer.json`, maar met deze nieuwe functionaliteit hoeft er niet meer speciaal hiervoor een `composer`-bestand te worden toegevoegd

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

(dit is de nieuwe aanbevolen manier om in TYPO3 klassen te registreren)

# Systeemwijzigingen

---

## Nieuwe Icon-factory (1)

- Logica voor het werken met iconen, icoonformaten en -overlays is nu gebundeld in de nieuwe klasse `IconFactory`
- De nieuwe Icon-factory vervangt de oude API voor iconen
- Alle iconen uit de core worden in de `IconRegistry`-klasse geregistreerd
- Om bestaande iconen te vervangen of om iconen aan de Icon-factory toe te voegen moeten extensies gebruik maken van `IconRegistry::registerIcon()`:

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

# Systeemwijzigingen

---

## Nieuwe Icon-factory (2)

- De TYPO3 CMS-core gebruikt 3 IconProviders:  
BitmapIconProvider, FontawesomeIconProvider en SvgIconProvider

- Voorbeeld:

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

- Geldige waarden voor `Icon::SIZE_...` zijn:  
SIZE\_SMALL, SIZE\_DEFAULT en SIZE\_LARGE
- Geldige waarden voor `Icon::STATE_...` zijn:  
STATE\_DEFAULT en STATE\_DISABLED

## Nieuwe Icon-factory (3)

- In de TYPO3 CMS-core is een ViewHelper beschikbaar waarmee het eenvoudig is om iconen te gebruiken in Fluid:

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

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

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

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

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

# Systeemwijzigingen

---

## Hooks en signals

- Er is een nieuw signal toegevoegd aan LinkValidator, die een aanvullende verwerking na het initialiseren van een bepaald record mogelijk maakt  
(bijv. om gegevens uit de pluginconfiguratie in te laden)
- Registreer de signal als volgt in `ext_localconf.php`:

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



# Systeemwijzigingen

---

## Systemextensie JumpUrl (1)

- Het genereren en verwerken van JumpURL's is verplaatst naar de nieuwe systeemextensie `jumpurl`
- Er zijn nieuwe hooks beschikbaar die het mogelijk maken om eigen URL's te genereren en te verwerken (zie volgende pagina)

**BREAKING CHANGE!**

# Systeemwijzigingen

---

## Systemextensie JumpUrl (2)

- Hook 1: manipuleren van een **URL** tijdens het maken van een link

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlHandlers']  
    ['myext_myidentifier']['handler'] = \Company\MyExt\MyUrlHandler::class;  
  
// klasse moet UrlHandlerInterface implementeren:  
class MyUrlHandler implements \TYPO3\CMS\Frontend\Http\UrlHandlerInterface {  
    ...  
}
```

- Hook 2: verwerken van **links**

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlProcessors']  
    ['myext_myidentifier']['processor'] = \Company\MyExt\MyUrlProcessor::class;  
  
// klasse moet UrlProcessorInterface implementeren:  
class MyUrlProcessor implements \TYPO3\CMS\Frontend\Http\UrlProcessorInterface {  
    ...  
}
```

# Systeemwijzigingen

---

## Command-line-interface (CLI)

- Wanneer vanaf de opdrachtregel `typo3/cli_dispatch.phpsh` wordt aangeroepen zonder of met een verkeerde parameter, dan wordt er een gekleurde foutmelding getoond
- Extbase-commandcontrollers kunnen nu ook in submappen van de Command-map bewaard worden
- Voorbeeld:

De controller in bestand:

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

...kan als volgt vanaf de opdrachtregel worden aangeroepen:

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

# Systeemwijzigingen

---

## Diversen (1)

- De verplaatsknoppen van TCA-type group kun nu expliciet worden uitgezet met de optie `hideMoveIcons = TRUE`
- Methode `makeCategorizable` is uitgebreid met een nieuwe parameter `override` waarmee een nieuwe categorieconfiguratie kan worden ingesteld voor een reeds geregistreerde tabel/veld-combinatie
- Voorbeeld:

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

Met de laatste parameter (hier: `TRUE`) wordt het overschrijven geforceerd (standaardwaarde is `FALSE`).

# Systeemwijzigingen

---

## Diversen (2)

- Er is een nieuwe functie die een unieke ID genereert:

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

- Bestandsextensie `.typoscript` is toegevoegd aan de lijst van geldige extensies voor tekstbestanden
- Een nieuwe configuratie-optie definieert de bestandsextensies van mediabestanden

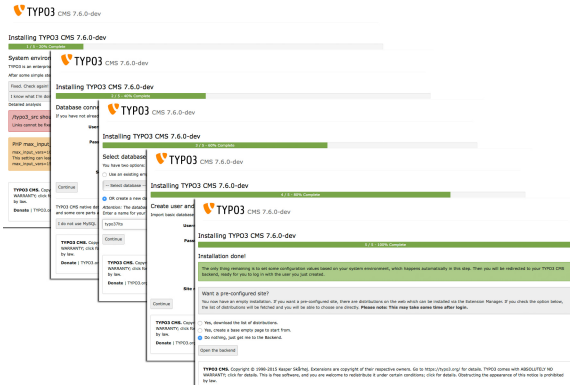
```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] =  
    'gif,jpg,jpeg,bmp,png,pdf,svg,ai,mov,avi';
```

**BREAKING CHANGE!**

# Stysteemwijzigingen

## Bootstrap in de Installatie-module (1)

- De Installatie-module is nu gebaseerd op Bootstrap - voor het installatiedeel:



## Bootstrap in de Installatie-module (2)

- De Installatie-module is nu gebaseerd op Bootstrap - voor de configuratie:

- Important actions
- Configuration Presets
- All configuration
- Upgrade Wizard
- System environment
- Folder structure **3**
- Test setup
- Clean up
- About

Logout from Install Tool

### Change configuration values

Changed values are written to LocalConfiguration.php. The optional file AdditionalConfiguration.php is not controlled by the TYPO3 CMS core and may override single settings again. Administrators must maintain AdditionalConfiguration.php on their own and should use it with care.

Filter by:

**Backend [BE]**

**[BE][disable\_exec\_function] = 0**

Boolean: Don't use exec() function (except for ImageMagick which is disabled by [GFX][im]=0). If set, all fileoperations are done by the default PHP-functions. This is necessary under Windows! On Unix the system commands by exec() can be used, unless this is disabled.

**Database [DB]**

**Extension Installation [EXT]**

**Frontend [FE]**

**Image Processing [GFX]**

**[GFX][im\_path\_lzw] = /usr/bin/**

Path to the IM tool 'convert' with LZW enabled! See 'gif\_compress'. If your version 4.2.9 of ImageMagick is compiled with LZW you may leave this field blank AND disable the flag 'gif\_compress'! Tip: You can call LZW 'convert' with a prefix like 'myver\_convert' by setting this path with it, eg. '/usr/bin/myver\_' instead of just '/usr/bin/'.

## CSRF-bescherming voor frontend plug-ins

- Nieuwe klasse ondersteunt gebruik van formulierbeschermings-API in de frontend
- Dit implementeert een CSRF-bescherming (Cross-Site Request Forgery)

```
$formToken = \TYPO3\CMS\Core\FormProtection\FormProtectionFactory::get()->getFormProtection()->
    generateToken('news', 'edit', $uid);
if (
    $dataHasBeenSubmitted
    && \TYPO3\CMS\Core\FormProtection\FormProtectionFactory::get()->validateToken(
        \TYPO3\CMS\Core\Utility\GeneralUtility::_POST('formToken'), 'User setup', 'edit')) {
    // bewerk de gegevens
}
else {
    // ongeldige token!
}
```



# Systeemwijzigingen

---

## Tabbladen voor linkbladerscherm (1)

- Het linkbladerscherm is uit te breiden met nieuwe tabbladen
- Elk tabblad wordt afgehandeld door een zogenoemde "LinkHandler", die de volgende Interface moet implementeren:  
`\TYPO3\CMS\Recordlist\LinkHandler\LinkHandlerInterface`
- LinkHandlers worden als volgt in PageTSconfig geregistreerd:

```
file {
    handler = TYPO3\CMS\Recordlist\LinkHandler\FileLinkHandler
    label = LLL:EXT:lang/locallang_browse_links.xlf:file
    displayAfter = page
    scanAfter = page
    configuration {
        customConfig = doorgegeven aan handler
    }
}
```

# Systeemwijzigingen

---

## Tabbladen voor linkbladerscherm (2)

- Opties `displayBefore` en `displayAfter` bepalen de posities van de tabbladen
- Opties `scanBefore` en `scanAfter` bepalen de volgorde waarin de handlers worden uitgevoerd bij het scannen van bestaande links

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['LinkBrowser']['hooks'][1444048118] = [  
    'handler' => \Vendor\Ext\MyClass::class,  
    'before' => [], // optioneel  
    'after' => [] // optioneel  
];
```

# Systeemwijzigingen

---

## API Modulesjabloon (1)

- Een nieuwe API voor de modulesjabloon heeft als doel de implementatie van DocHeaders normalizeren
- Voorbeeld 1: knop toevoegen

```
$openInNewWindowButton = $this->moduleTemplate->getDocHeaderComponent()->getButtonBar()  
->makeLinkButton()  
->setHref('#')  
->setTitle($this->getLanguageService()->sL(  
    'LLL:EXT:lang/locallang_core.xlf:labels.openInNewWindow', TRUE  
))  
->setIcon($this->iconFactory->getIcon('actions-window-open', Icon::SIZE_SMALL))  
->setOnClick($aOnClick);
```

```
$this->moduleTemplate->getDocHeaderComponent()->getButtonBar()  
->addButton($openInNewWindowButton, ButtonBar::BUTTON_POSITION_RIGHT);
```

## API Modulesjabloon (2)

### ■ Voorbeeld 2: menu met menu-items toevoegen

```
$languageMenu = $this->moduleTemplate->getDocHeaderComponent()
->getModuleMenuRegistry()->makeMenu()
->setIdentifier('_langSelector')
->setLabel($this->getLanguageService()->sL(
    'LLL:EXT:lang/locallang_general.xlf:LGL.language', TRUE
));

$menuitem = $languageMenu->makeMenuItem()
->setTitle($lang['title'] . $newTranslation)
->setHref($href);

if((int)$lang['uid'] === $currentLanguage) {
    $menuItem->setActive(TRUE);
}

$languageMenu->addMenuItem($menuItem);
$this->moduleTemplate->getDocHeaderComponent()->getModuleMenuRegistry()->addMenu($languageMenu);
```

## PSR-7 Routing voor Backend AJAX Requests

- Om een route voor een AJAX-request toe te voegen, kan het bestand `Configuration/Backend/AjaxRoutes.php` worden aangemaakt met de volgende inhoud:

```
return [  
    // doe iets  
    'unique_route_name' => [  
        'path' => '/toolcollection/some-action',  
        'target' => \Vendor\Controller\SomeController::class . '::myAction',  
    ]  
];
```

# Systeemwijzigingen

---

## OpenID Hook `getUserRecord`

Twee hooks zijn toegevoegd aan de OpenID-service (1/2)

- Hook 1:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['openid']['getUserRecord']
```

- Wijzig het gebruikersrecord nadat het is opgehaald, of
- Maak een nieuw record aan als er geen is gevonden
- Parameters `record`, `response` en `authInfo` worden doorgegeven aan de hook

# Systeemwijzigingen

---

## OpenID Hook `authRequest`

Twee hooks zijn toegevoegd aan de OpenID-service (2/2)

- Hook 2:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['openid']['authRequest']
```

- Wijzigt het Authentication Request, voordat het wordt verzonden
- Kan worden gebruikt om extra attributen aan te vragen uit de OpenID-Server zoals een bijnaam
- Parameters `authRequest` en `authInfo` worden doorgegeven aan de hook

# Systeemwijzigingen

---

## Hooks en Signals (1)

- Het is nu mogelijk om de upload-map te wijzigen die terugkomt uit `BackendUserAuthentication::getDefaultUploadFolder()`
- Registreer de hook in bestand `ext_localconf.php` als volgt:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_userauthgroup.php']  
  ['getDefaultUploadFolder'][] =  
    \Vendor\MyExtension\Hooks\DefaultUploadFolder::class . '->getDefaultUploadFolder';
```



# Systeemwijzigingen

---

## Hooks en Signals (2)

### Voorbeeld:

```
<?php
namespace Vendor\MyExtension\Hooks;
use TYPO3\CMS\Core\Authentication\BackendUserAuthentication;
use TYPO3\CMS\Core\Resource\Folder;

/**
 * Class DefaultUploadFolder
 */
class DefaultUploadFolder {

    /**
     * Haal de standaard upload-map op
     * Als er een map beschikbaar is met dezelfde naam als het laatste deel van de tabelnaam gebruik dan
     * die map
     * @param array $params
     * @param BackendUserAuthentication $backendUserAuthentication
     * @return Map
     */
    public function getDefaultUploadFolder($params, BackendUserAuthentication $backendUserAuthentication)
    {
        [...]
    }
}
```

# Systeemwijzigingen

---

## Hooks en Signals (3)

### Voorbeeld (vervolg):

```
[...]  
  
/** @var Folder $uploadFolder */  
$uploadFolder = $params['uploadFolder'];  
$pid = $params['pid'];  
$table = $params['table'];  
$field = $params['field'];  
  
$matches = [];  
if (!empty($uploadFolder) && preg_match('/_([a-z]+)/', $table, $matches)) {  
    $folderName = $matches[1];  
    if ($uploadFolder->hasFolder($folderName)) {  
        $uploadFolder = $uploadFolder->getSubfolder($folderName);  
    }  
}  
return $uploadFolder;  
}  
}
```

# Systemwijzigingen

---

## Diversen

- Het gebruik van het TCA-veldtype `select` vereist het gebruik van de optie `renderType`
- Geldige waarden zijn:

```
'renderType' => 'selectMultipleSideBySide',  
'renderType' => 'selectCheckBox',  
'renderType' => 'selectSingle',  
'renderType' => 'selectSingleBox',  
'renderType' => 'selectTree',
```

# Bronnen en auteurs

# Bronnen en auteurs

---

## Bronnen

### TYPO3 Nieuws:

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

### Publicatie-informatie:

- [https://wiki.typo3.org/Category:ReleaseNotes/TYPO3\\_7.x](https://wiki.typo3.org/Category:ReleaseNotes/TYPO3_7.x)
- [INSTALL.md](#) en [Changelog](#)
- `typo3/sysext/core/Documentation/Changelog/*`

### TYPO3 Bug-/Issuetracker:

- <https://forge.typo3.org/projects/typo3cms-core>

### TYPO3 Git Repositories:

- <https://git.typo3.org/Packages/TYPO3.CMS.git>
- <https://git.typo3.org/Packages/TYPO3.Fluid.git>

## **TYPO3 CMS What's New Slides:**

Patrick Lobacher

(Onderzoek, Informatieverzamelen en Duitse versie)

Michael Schams

(Projectleider en Engelse versie)

### **Vertalingen en bijdragen door:**

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

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

Licensed under Creative Commons BY-NC-SA 3.0

