

TYPO3 CMS 8.0 – What's New

Overzicht van nieuwe features, wijzigingen en verbeteringen

Gemaakt door:

Patrick Lobacher and Michael Schams

Nederlandse vertaling door:

Jigal van Hemert

TYPO3 CMS 8.0 - What's New

Inhoudsopgave

Inleiding

Gebruikersinterface backend

TSconfig & TypoScript

Systeemwijzigingen

Extbase & Fluid

Verouderde/verwijderde functies

Bronnen en auteurs

Inleiding

De feiten

Inleiding

TYPO3 CMS 8.0 – De feiten

- Publicatiedatum: 22 maart 2016
- Publicatietype: Sprint Release
- Slogan: Start de motoren



Start your engines

TYPO3 v8.0 released

TYPO3

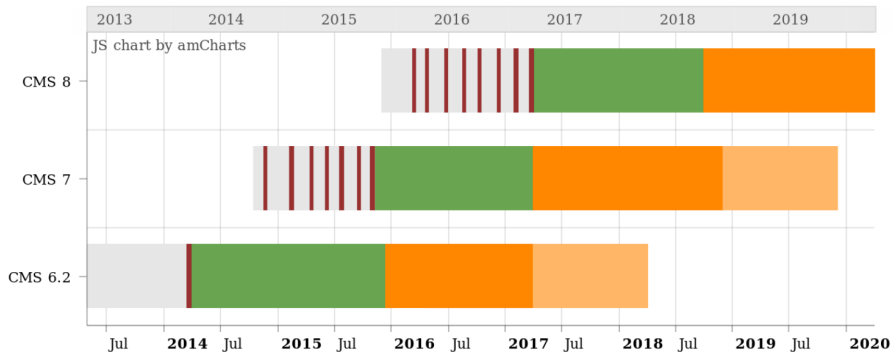
Inleiding

Systemeisen

- PHP: versie 7
- MySQL: versie 5.5 tot 5.7
- Schijfruimte: min 200 MB
- PHP-instellingen:
 - `memory_limit` \geq 128M
 - `max_execution_time` \geq 240s
 - `max_input_vars` \geq 1500
 - compilatieoptie `--disable-ipv6` mag niet gebruikt worden
- De backend vereist Microsoft Internet Explorer 11 of later, Microsoft Edge, Google Chrome, Firefox, Safari of een andere moderne compatibele browser

Inleiding

Planning voor ontwikkeling en publicatie



Inleiding

TYPO3 CMS Roadmap

Publicatiedatums en primaire focus:

- v8.0 22/Mar/2016 Last minute toevoegingen
- v8.1 03 mei 2016 Cloud-integratie
- v8.2 05 jul 2016 Rich Text Editor
- v8.3 30 aug 2016 Bewerken in Frontend powereditie
- v8.4 18 okt 2016 *onbekend*
- v8.5 20 dec 2016 Integrator-ondersteuning
- v8.6 14 feb 2017 *onbekend*
- v8.7 04 apr 2017 LTS Voorbereiding

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

<https://typo3.org/news/article/kicking-off-typo3-v8-development/>

Inleiding

Installatie

- Officiële installatieprocedure op Linux/Mac OS X
(DocumentRoot bijvoorbeeld `/var/www/site/htdocs`):

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/8.0
$ tar xzf typo3_src-8.0.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-8.0.0 typo3_src
$ ln -s typo3_src/index.php
$ ln -s typo3_src/typo3
$ touch FIRST_INSTALL
```

- Symbolische links op Microsoft Windows:
 - Gebruik `junction` op Windows XP/2000
 - Gebruik `mklink` op Windows Vista en Windows 7

Inleiding

Upgrade naar TYPO3 CMS 8.x

- Upgrades alleen mogelijk vanaf TYPO3 CMS 7.6 LTS
- TYPO3 CMS < 7.6 LTS moet eerst naar TYPO3 CMS 7.6 LTS bijgewerkt worden
- Upgrade-instructies:
http://wiki.typo3.org/Upgrade#Upgrading_to_8.0
- Officiële TYPO3-handleiding "TYPO3 Installation and Upgrading":
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Algemene aanpak:
 - Controleer minimale systeemeisen (PHP, MySQL, etc.)
 - Bekijk **deprecation_*.log** in oude TYPO3 installatie
 - Update alle extensies naar laatste versie
 - Zet nieuwe broncode neer en start Install Tool -> Upgrade Wizard
 - Bekijk startmodule voor backend gebruikers (optioneel)

Inleiding

PHP Versie 7

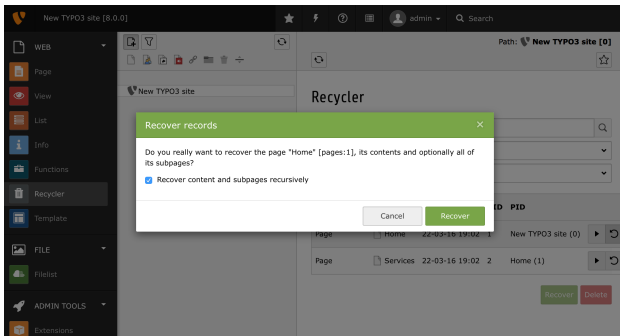
- PHP 7.0 is de minimale eis voor TYPO3 CMS 8.x
- TYPO3 zal volgende PHP 7 versies ondersteunen wanneer deze uitkomen
- Deze versie geeft significant meer prestaties op het hele systeem
- Niet alleen backendgebruikers merken een soepelere interface, maar ook het nieuwe record voor een volledig gecachte pagina in de frontend is nu minder dan 7 milliseconde, wat ongeveer 40% sneller is in vergelijking met dezelfde website met PHP versie 5.5
- We zijn ook begonnen nieuwe features van deze PHP-versie te gebruiken, bijvoorbeeld de cryptografisch veilige pseudo-random generatoren worden al ingezet

Hoofdstuk 1: Gebruikersinterface backend

Gebruikersinterface backend

Herstel pagina's en alle onderliggende pagina's

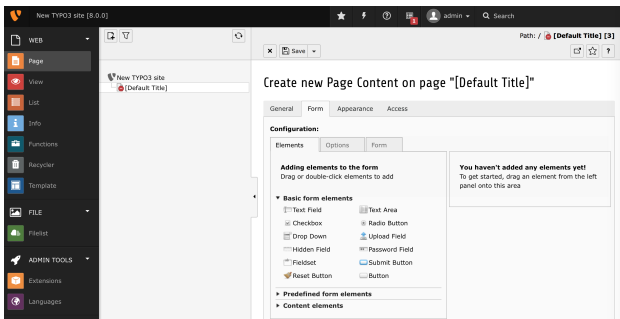
De Prullenbak ondersteunt het herstellen van verwijderde pagina's en alle onderliggende pagina's. Dit is wegens interne rechtenbeperkingen alleen beschikbaar voor admin-gebruikers.



Gebruikersinterface backend

Gebruik formulierassistent als interne assistent

De assistent van EXT:form wordt direct als inline assistent geladen. Het is niet nodig om het nieuwe inhoudselement op te slaan en te herladen om de assistent te kunnen openen. Dit is een grote verbetering in het gebruik.



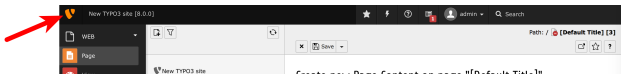
Gebruikersinterface backend

Alternatief logo in de backend

Het logo in de backend in de linkerbovenhoek kan nu via de configuratie van EXT:backend in het Extensiebeheer ingesteld worden.

Configuratieopties zijn:

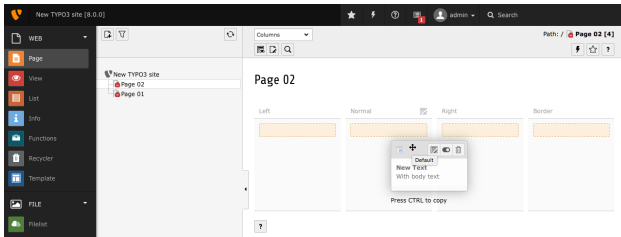
- **relatief pad binnen de TYPO3-installatie**
bijv. "fileadmin/images/my-background.jpg"
- **pad naar een extensie**
bijv. "EXT:my_theme/Resources/Public/Images/my-background.jpg"
- **extern bestand**
bijv. "//example.com/my-background.png"



Gebruikersinterface backend

Pagina's kopiëren via slepen

Naast het gebruikelijke slepen in de paginamodule (waarbij inhoudselementen *verplaatst* worden), kunnen nu kopieën gemaakt worden: houd de CTRL-toets ingedrukt om een kopie van het gesleepte element te maken. Na het loslaten wordt de pagina herladen om zeker te zijn dat het nieuwe element wordt aangemaakt met alle benodigde informatie.



Hoofdstuk 2: TSconfig & TypoScript

Sorteervolgorde van tabs van assistent nieuw inhoudselement

- Het is mogelijk om de volgorde van de tabbladen van de assistent voor nieuwe inhoudselementen in te stellen met `before-` en `after-`waardes in de `page-TSconfig`:

```
mod.wizards.newContentElement.wizardItems.special.before = common  
mod.wizards.newContentElement.wizardItems.forms.after = common,special
```

TScnfig & TypoScript

`HTMLparser.stripEmptyTags.keepTags`

- Nieuwe optie voor de configuratie van `HTMLparser.stripEmptyTags` is toegevoegd, voor het behouden van aangegeven tags
- Eerder kon alleen een serie tags verwijderd worden
- Het volgende voorbeeld verwijdert alle lege tags **behalve** `tr` en `td`:

```
HTMLparser.stripEmptyTags = 1
HTMLparser.stripEmptyTags.keepTags = tr,td
```

Belangrijk: als deze instelling gebruikt wordt, heeft `stripEmptyTags.tags` geen effect meer. Er kan maar één optie tegelijkertijd gebruikt worden.

TScnfig & TypoScript

EXT:form - integratie van voorgedefinieerde formulieren (1)

- Het inhoudselement van EXT:form ondersteunt nu het opnemen van voorgedefinieerde formulieren.
- Een integrator kan formulieren definiëren (bijv. in een site package) via `plugin.tx_form.predefinedForms`
- Een redacteur kan een nieuw `mailform`-inhoudselement op een pagina zetten en een formulier uit de voorgedefinieerde lijst kiezen
- Integrators kunnen formulieren maken met TypoScript, waarbij meer opties beschikbaar zijn dan via de assistent (integrators kunnen bijv. `stdWrap` functionaliteit gebruiken die via de assistent (uit veiligheidsoverwegingen) niet beschikbaar zijn.

Tsconfig & TypoScript

EXT:form - integratie van voorgedefinieerde formulieren (2)

- Het is voor redacteurs niet meer nodig om de formulierassistent te gebruiken. Redacteurs kunnen kiezen uit voorgedefinieerde formulier die qua lay-out geoptimaliseerd zijn.
- Formulieren kunnen hergebruikt worden in de hele installatie
- Formulieren kunnen buiten de database opgeslagen worden onder versiebeheer
- Om voorgedefinieerde formulier in de backend te kunnen kiezen, moet het formulier via PageTS geregistreerd worden:

```
TCEFORM.tt_content.tx_form_predefinedform.addItem.contactForm =  
LLL:EXT:my_theme/Resources/Private/Language/locallang.xlf:contactForm
```

TScnfig & TypoScript

EXT:form: integratie van voorgedefinieerde formulieren (3)

■ Voorbeeld:

```
plugin.tx_form.predefinedForms.contactForm = FORM
plugin.tx_form.predefinedForms.contactForm {
    enctype = multipart/form-data
    method = post
    prefix = contact
    confirmation = 1
    postProcessor {
        1 = mail
        1 {
            recipientEmail = test@example.com
            senderEmail = test@example.com
            subject {
                value = Contact form
                lang.nl = Contactformulier
            }
        }
    }
}
10 = TEXTLINE
10 {
    name = name
...

```

Hoofdstuk 3: Systeemwijzigingen

Stysteemwijzigingen

Ondersteuning PECL-memcached in MemcachedBackend

- Ondersteuning voor de PECL-module "memcached" is toegevoegd aan de MemcachedBackend van het Caching Framework
- Als zowel "memcache" als "memcached" zijn geïnstalleerd, wordt "memcache" gebruikt om geen brekende wijziging te zijn.
- Een integrator kan de optie `peclModule` gebruiken om de voorkeur in te stellen:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['my_memcached'] = [  
    'frontend' => \TYPO3\CMS\Core\Cache\Frontend\VariableFrontend::class  
    'backend' => \TYPO3\CMS\Core\Cache\Backend\MemcachedBackend::class,  
    'options' => [  
        'peclModule' => 'memcached',  
        'servers' => [  
            'localhost',  
            'server2:port'  
        ]  
    ]  
];
```

Systeemwijzigingen

Ondersteuning voor Symfony Console (1)

- TYPO3 ondersteunt de Symfony Console automatisch via een nieuw commandoregel script in `typo3/sysext/core/bin/typo3`. Bij TYPO3-installaties die via Composer zijn gemaakt wordt het uitvoerbare bestand gekoppeld aan de `bin`-folder, bijv. `bin/typo3`.
- Het nieuwe script ondersteunt nog steeds de bestaande argumenten als er geen correct Symfony Console-commando wordt gevonden.
- Een commando dat via de TYPO3-commandoregel beschikbaar moet zijn, kan geregistreerd worden door het plaatsen van een `Configuration/Commands.php`-bestand in een geïnstalleerde extensie. De lijst `Symfony/Console/Command` klassen die uitgevoerd kunnen worden door TYPO3, is een hash-tabel. De index is de naam van het commando dat als eerste argument voor TYPO3 gebruikt wordt.

Systeemwijzigingen

Ondersteuning voor Symfony Console (2)

- Een vereiste parameter bij het registreren van een commando is de `class`-eigenschap. Optioneel kan de `user`-parameter ingesteld worden zodat een backend gebruiker ingelogd is bij het aanroepen van het commando.
- Een `Configuration/Commands.php` kan er zo uitzien:

```
return [  
    'backend:lock' => [  
        'class' => \TYPO3\CMS\Backend\Command\LockBackendCommand::class  
    ],  
    'referenceindex:update' => [  
        'class' => \TYPO3\CMS\Backend\Command\ReferenceIndexUpdateCommand::class,  
        'user' => '_cli_lowlevel'  
    ]  
];
```

Systeemwijzigingen

Ondersteuning voor Symfony Console (3)

- Een aanroep kan er zo uitzien:

```
bin/typo3 backend:lock http://example.com/maintenance.html
```

- Voor een niet-Composer-installatie:

```
typo3/sysex/core/bin/typo3 backend:lock http://example.com/maintenance.html
```

Systeemwijzigingen

Cryptografisch veilige pseudotoevalsgenerator

- Een nieuwe cryptografisch veilige pseudotoevalsgenerator (CSPRNG) is geïmplementeerd in de TYPO3 core.
Het gebruikt de nieuwe CSPRNG-functies in PHP 7.
- De API zit in de klasse `\TYPO3\CMS\Core\Crypto\Random`
- Voorbeeld:

```
use \TYPO3\CMS\Core\Crypto\Random;
use \TYPO3\CMS\Core\Utility\GeneralUtility;

// Willekeurige bytes ophalen
$someRandomString = GeneralUtility::makeInstance(Random::class)->generateRandomBytes(64);

// Dobbelsteen rollen...
$tossedValue = GeneralUtility::makeInstance(Random::class)->generateRandomInteger(1, 6);
```

Stysteemwijzigingen

Assistentcomponent (1)

- Een nieuwe assistentcomponent is toegevoegd. Dit component kan gebruikt worden voor begeleide interacties
- De RequireJS module kan gebruikt worden door het invoegen van TYPO3\CMS\Backend\Wizard
- De assistent ondersteunt alleen simpele acties (vertakkingen zijn nog niet mogelijk)
- De API zit in de klasse \TYPO3\CMS\Core\Crypto\Random
- De assistentcomponent heeft de volgende publieke functies:

```
addSlide(identificer, title, content, severity, callback)
addFinalProcessingSlide(callback)
set(key, value)
show()
dismiss()
getComponent()
lockNextStep()
unlockNextStep()
```

Systemwijzigingen

Assistentcomponent (2)

- De "event" `wizard-visible` wordt afgevuurd als de assistent klaar is met renderen
- Assistenten worden gesloten door het afvuren van `wizard-dismiss`
- Assistenten vuren het `wizard-dismissed` "event" af als de assistent wordt gesloten
- Een eigen listener kan geregistreerd worden met `Wizard.getComponent()`

Systeemwijzigingen

Gegenererde assetbestanden verplaatst

- De mappenstructuur binnen typo3temp is gewijzigd om de assets die publiekelijk beschikbaar zijn, te scheiden van de tijdelijke bestanden (bijv. voor caching of blokkeringen die alleen voor de server bedoeld zijn).
- Deze assets zijn verplaatst van de mappen:
processed, compressor, GB, temp, Language, pics
naar:
 - typo3temp/assets/js/
 - typo3temp/assets/css/
 - typo3temp/assets/compressed/
 - typo3temp/assets/images/

Systeemwijzigingen

ImageMagick/GraphicsMagick wijzigingen (1)

- Instellingen voor grafische acties van Image- of GraphicsMagick zijn hernoemd (bestand: `LocalConfiguration.php`).
OUD: `im_`
NIEUW: `processor_`
- Negatieve naamgeving zoals `noScaleUp` is omgezet in positieve varianten. Tijdens de conversie worden de vorige waarden omgekeerd overeenkomstig de naamgeving van deze opties.
- Daarnaast zijn verwijzingen naar specifieke versies van ImageMagick/GraphicsMagick verwijderd uit de namen en waarden van de opties.

Systeemwijzigingen

ImageMagick/GraphicsMagick wijzigingen (2)

- De niet gebruikte configuratieoptie `image_processing` is verwijderd zonder vervanging
- De processor-specifieke optie `colorspace` is voorzien van een *namespace* onder de `processor_` boom

Systeemwijzigingen

Hooks en Signals (1)

- Een extra hook is toegevoegd aan de methode `BackendUtility::viewOnClick()` voor het nabewerken van de voorvertoningsURL
- Registreer een hook-klasse die de methode `postProcess` bevat:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_befunc.php']['viewOnClickClass'][] =  
    \Vendor\MyExt\Hooks\BackendUtilityHook::class;
```

Systeemwijzigingen

Hooks en Signals (2)

- Vóór TYPO3 CMS 7.6 was het mogelijk om een overlapping van een record te overschrijven in Web -> List. Een nieuwe hook in TYPO3 CMS 8.0 biedt de oude functionaliteit.
- De hook wordt aangeroepen met de volgende argumenten:

```
/**
 * @param string $table
 * @param array $row
 * @param array $status
 * @param string $iconName
 * @return string the new (or given) $iconName
 */
function postOverlayPriorityLookup($table, array $row, array $status, $iconName) { ... }
```

- Registreer de hook-klasse die dit implementeert met de naam `postOverlayPriorityLookup`:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['IconFactory::class']['overrideIconOverlay'][] =
    \VENDOR\MyExt\Hooks\IconFactoryHook::class;
```

Systeemwijzigingen

Hooks en Signals (3)

- Een nieuw signal is geïmplementeerd voordat een opslag voor bronnen is geïnitieerd.
- Registreer de klasse met de logica in `ext_localconf.php`:

```
$dispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(  
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class);  
$dispatcher->connect(  
    \TYPO3\CMS\Core\Resource\ResourceFactory::class,  
    ResourceFactoryInterface::SIGNAL_PreProcessStorage,  
    \MY\ExtKey\Slots\ResourceFactorySlot::class,  
    'preProcessStorage'  
);
```

- De methode wordt aangeroepen met de volgende parameters:
 - `int $uid` de uid van het record
 - `array $recordData` alle record-gegevens als array
 - `string $fileIdentificer` de bestandsidentificer

Systemwijzigingen

Algoritme voor wachtwoordhash: PBKDF2

- Een nieuw hash-algoritme voor wachtwoorden, "PBKDF2", is toegevoegd aan de systeemextensie "saltedpasswords"
- PBKDF2 betekent: Password-Based Key Derivation Function 2
- Het algoritme kost veel rekenkracht om brute force password cracking te weerstaan

Hoofdstuk 4: Extbase & Fluid

Extbase & Fluid

Standalone Fluid

- De Fluid render-engine van TYPO3 CMS is vervangen door de standalone versie van Fluid die nu als een composer-afhankelijkheid opgenomen is
- De oude Fluid-extensie is omgezet in een zogenaamde *Fluid adapter* waarmee TYPO3 CMS standalone Fluid kan gebruiken
- Nieuwe features/mogelijkheden zijn in bijna elk onderdeel van Fluid toegevoegd
- Meest belangrijk: diverse onderdelen van Fluid die compleet intern waren en die onmogelijk overschreven konden worden, zijn nu simpel te vervangen en voorzien van een API

RenderingContext (1)

- Het belangrijkste onderdeel van de nieuwe API is de RenderingContext
- Voorheen was RenderingContext intern gebruikt door Fluid. Dit is nu uitgebreid en verantwoordelijk voor een nieuwe Fluid feature: **implementation provisioning**
- Hiermee kunnen ontwikkelaars een serie klassen wijzigen die Fluid gebruikt voor het parsen, oplossen, cachen, etc.
- Dit is te bereiken door ofwel een eigen RenderingContext of het bewerken van de standaard RenderingContext met publiek beschikbare functies.

Rendering Context (2)

- De features op de volgende pagina's worden aangestuurd door het wijzigen van de RenderingContext. Standaard zijn ze niet ingeschakeld, maar door het aanroepen van een simpele methode (via de View instantie) kunnen ze ingeschakeld worden:

```
$view->getRenderingContext()->setLegacyMode(false);
```


ExpressionNodes (1)

- ExpressionNodes zijn een nieuw soort syntaxstructuren van Fluid die alle een eigenschap gemeen hebben: ze werken alleen binnen accolades

```
$view->getRenderingContext()->setExpressionNodeTypes(array(  
    'Class\Number\One',  
    'Class\Number\Two'  
));
```

- Ontwikkelaars kunnen hun eigen soorten ExpressionNode toevoegen
- Elk bestaat uit een patroon en methodes bepaald door een interface om de overeenkomsten af te handelen
- Een bestaand soort ExpressionNode kan gebruikt worden als referentie

ExpressionNodes (2)

ExpressionNodeTypes maken nieuwe syntaxen mogelijk:

- **CastingExpressionNode**

maakt het omzetten van variabelen naar bepaalde types mogelijk; bijvoorbeeld om zeker te zijn van een geheel getal of een boolean. Het wordt gebruikt met een `as` sleutelwoord: `{myStringVariable as boolean}` of `{myBooleanVariable as integer}` enzovoort. Pogingen om een variabele om te zetten naar een incompatibel type resulteren in een standaard Fluid-foutmelding.

- **MathExpressionNode**

maakt basisberekeningen met variabelen mogelijk, bijvoorbeeld `{myNumber + 1}`, `{myPercent / 100}` of `{myNumber * 100}` enzovoort. Een onmogelijke uitdrukking geeft geen uitvoer.

Extbase & Fluid

ExpressionNodes (3)

ExpressionNodeTypes maken nieuwe syntax mogelijk:

- **TernaryExpressionNode**

maakt een drieweg-voorwaarde mogelijk die werkt met variabelen. Typisch gebruik: "als deze variabele dan gebruik die variabele anders gebruik andere variabele". Het wordt gebruikt als:

```
{myToggleVariable ? myThenVariable : myElseVariable}
```

Let op: ondersteunt geen geneste uitdrukkingen, inline ViewHelper syntax of vergelijkbaar erbinnen. Het moet alleen met standaard variabelen gebruikt worden.

Extbase & Fluid

Namespaces zijn uitbreidbaar (1)

- Fluid maakt het mogelijk om een namespace alias (bijvoorbeeld `f:`) uit te breiden door door extra PHP namespaces eraan toe te voegen
- PHP namespaces worden ook gecontroleerd op de aanwezigheid van ViewHelper-classes
- Dit betekent ook dat ontwikkelaars specifieke ViewHelpers kunnen overschrijven met eigen versies en eigen ViewHelpers laten aanroepen als de `f:` namespace gebruikt wordt
- Deze wijziging betekent ook dat namespaces niet langer eenduidig zijn. Bij het gebruik van `{namespace f=My\Extension\ViewHelpers\}` verschijnt er niet langer de foutmelding "namespace al geregistreerd". Fluid zal de PHP namespace toevoegen en ook daar zoeken naar ViewHelpers.

Namespaces zijn uitbreidbaar (2)

- Extra namespaces worden van onder af nagelopen waardoor extra namespaces ViewHelper-klassen kunnen overschrijven door ze in dezelfde scope te plaatsen.
- Voorbeeld: `f:format.nl2br` kan worden overschreven door `My\Extension\ViewHelpers\Format\Nl2brViewHelper`, met de namespace-registratie van de vorige pagina

Extbase & Fluid

Renderen via `f:render` (1)

Sta standaardinhoud toe op optionele `f:render`:

- Waar `f:render` wordt gebruikt met de vlag `optional = TRUE` resulteert een missende section in lege uitvoer.
- In plaats van deze lege uitvoer kan een nieuw attribuut `default` (gemengd) ervoor zorgen dat de inhoud daarvan als standaarduitvoer wordt gebruikt.
- Tevens kan de inhoud van de tag gebruikt worden als standaarduitvoer, zoals bij vele ViewHelpers waarbij zowel het attribuut als de inhoud gebruikt kunnen worden

Extbase & Fluid

Renderen via `f:render` (2)

Tag-inhoud doorgeven van `f:render` naar `partial/section`:

- Een nieuwe benadering van de structuur van een Fluid sjabloon
- `Partials` en `sections` kunnen gebruikt worden als "omsluiting" van willekeurige delen van een sjabloon.
- Voorbeeld:

```
<f:section name="MyWrap">
  <div>
    <!-- meer HTML, eventueel met variabelen -->
    <!-- tag-inhoud voor f:render uitvoer: -->
    {contentVariable -> f:format.raw()}
  </div>
</f:section>

<f:render section="MyWrap" contentAs="contentVariable">
  Deze inhoud wordt omsloten. Hier kan willekeurige Fluid komen.
</f:render>
```

Complexe voorwaarderlijke statements

- Fluid ondersteunt elke mate van complexe voorwaarderlijke statements zowel genest als gegroepeerd:

```
<f:if condition="{variableOne} && {variableTwo} || {variableThree} || {variableFour}">
  // Uitgevoerd als zowel variabele een en twee 'waar' opleveren,
  // of variabele drie of vier dat opleveren.
</f:if>
```

- Daarbij kent `f:else` nu een "elseif"-achtige functie:

```
<f:if condition="{variableOne}">
  <f:then>Doe dit</f:then>
  <f:else if="{variableTwo}">
    Doe dit als variabele twee 'waar' oplevert
  </f:else>
  <f:else if="{variableThree}">
    Doe dit als variabele twee 'waar' oplevert
  </f:else>
  <f:else>
    Of doe dit als niets waar is
  </f:else>
</f:if>
```


Dynamische delen van namen van variabelen(1)

- Een andere, nieuwe feature, ook backwards compatibel, is de mogelijkheid om sub-variabele referenties te gebruiken bij variabelen. Neem bijvoorbeeld de volgende Fluid variabelen-array:

```
$mykey = 'foo'; // of 'bar', vanuit elke bron
$view->assign('data', ['foo' => 1, 'bar' => 2]);
$view->assign('key', $mykey);
```

- Met het volgende Fluid-sjabloon:

```
U koos: {data.{key}}.
(uitvoer: "1" als de index "foo" is, "2" als de index "bar" is)
```

Dynamische delen van namen van variabelen (2)

- Dezelfde benadering kan ook gebruikt worden om dynamisch delen van de naam van een stringvariabele te genereren:

```
$mijndynamischdeel = 'Eerste'; // of 'Tweede', vanuit elke bron
$view->assign('mijnEersteVariabele', 1);
$view->assign('mijnTweedeVariabele', 2);
$view->assign('welke', $mijndynamischdeel);
```

- Met dit Fluid-sjabloon:

```
U koos: {mijn{welke}Variabele}.
(uitvoer: "1" als 'welke' "Eerste" is, of "2" als 'welke' "Tweede" is)
```

Nieuwe ViewHelpers

- Er zijn een paar nieuwe ViewHelpers toegevoegd als onderdeel van standalone Fluid en daardoor ook beschikbaar in TYPO3:
 - `f:or`

Dit is een korte vorm om (geschakelde) voorwaarden te schrijven. Het ondersteunt de volgende syntax die elke variabele naloopt en de eerste die niet leeg is als uitvoer gebruikt.

```
{variableOne -> f:or(alternative: variableTwo) -> f:or(alternative: variableThree)}
```
 - `f:spaceless`

Dit kan gebruikt worden als tag rond sjablooncode om overvloedige witruimte te verwijderen evenals lege regels die bijvoorbeeld ontstaan door het inspringen van ViewHelpers.

ViewHelper-namespaces kunnen ook vanuit PHP uitgebreid worden

- Via de ViewHelperResolver van de RenderingContext kunnen ontwikkelaars de ViewHelper-namespaces wijzigen op globale (lees: per View-instantie) schaal:

```
$resolver = $view->getRenderingContext()->getViewHelperResolver();  
// equivalent van registratie van namespace in sjablo(en):  
$resolver->registerNamespace('news', 'GeorgRinger\News\ViewHelpers');  
// extra PHP namespace toevoegen om te controleren bij het opzoeken van ViewHelpers:  
$resolver->extendNamespace('f', 'My\Extension\ViewHelpers');  
// alle namespaces vooraf, globaal, voor het verwerken van de sjabloon instellen:  
$resolver->setNamespaces(array(  
    'f' => array(  
        'TYPO3Fluid\Fluid\ViewHelpers', 'TYPO3\CMS\Fluid\ViewHelpers',  
        'My\Extension\ViewHelpers'  
    ),  
    'vhs' => array(  
        'FluidTYPO3\Vhs\ViewHelpers', 'My\Extension\ViewHelpers'  
    ),  
    'news' => array(  
        'GeorgRinger\News\ViewHelpers',  
    );  
));
```

ViewHelpers kunnen allerlei argumenten accepteren (1)

- Hiermee kan een ViewHelper-klasse elk aantal extra argumenten ontvangen met elke gewenste naam
- Het werkt door het scheiden van de argumenten die doorgegeven worden naar elke ViewHelper in twee groepen: diegene die gedeclareerd zijn via `registerArgument` (of de argumenten van de rendermethode) en diegene die dat niet zijn
- De niet-gedeclareerde argumenten worden doorgegeven aan de speciale functie `handleAdditionalArguments` in de ViewHelper-klasse, die in de standaardimplementatie een foutmelding geeft als er extra argumenten zijn

ViewHelpers kunnen allerlei argumenten accepteren (2)

- Door het overschrijven van deze methode in de ViewHelper, is te wijzigen of en wanneer er een foutmelding moet komen als er ongeregistreerde argumenten zijn
- Hiermee kunnen TagBasedViewHelpers ook willekeurige argumenten met een `data-`prefix accepteren zonder foutmelding
- bij TagBasedViewHelpers voegt de methode `handleAdditionalArguments` eenvoudigweg de nieuwe argumenten toe aan de tag die opgebouwd wordt en geeft een foutmelding als er extra argumenten zijn die noch geregistreerd zijn, noch een `data-`prefix hebben.

Argument "allowedTags" voor `f:format.stripTags`

- Het argument `allowedTags` bevat een lijst met HTML tags die niet verwijderd worden bij `f:format.stripTags`
- De syntax voor de lijst met tags is identiek aan de tweede parameter van de PHP functie `strip_tags` (zie: http://php.net/strip_tags)

Toegang tot ObjectStorage als array in Fluid

- Maakt een alias van `toArray()` aan waarmee de methode die aangeroepen wordt als `toArray()` (die het op zijn beurt mogelijk maakt om de methode transparant aan te roepen vanuit `ObjectAccess::getPropertyPath`) toegang mogelijk maakt in Fluid en andere plaatsen
- Door het maken van een simpele alias van `toArray()` op de `ObjectStorage`, kan het aangeroepen worden als `toArray()`
- Voorbeeld: haal het 4e element op

```
// in PHP:  
ObjectAccess::getPropertyPath($subject, 'objectstorageproperty.array.4')
```

```
// in Fluid:  
{myObject.objectstorageproperty.array.4}  
{myObject.objectstorageproperty.array.{dynamicIndex}}
```


Hoofdstuk 5: Verouderde/verwijderde functies

Verouderde/verwijderde functies

Diversen

- De volgende configuratieopties zijn verwijderd:

- `$TYPO3_CONF_VARS['SYS']['t3lib_cs_utils']`
- `$TYPO3_CONF_VARS['SYS']['t3lib_cs_convMethod']`

(functionaliteit wordt nu automatisch gedetecteerd en mbstring wordt standaard gebruikt indien beschikbaar)

- De verouderde TypoScript-optie `page.includeJSlibs` is verwijderd. Gebruik in plaats hiervan de TypoScript-optie `page.includeJSLibs` (hoofdletter "L")
- De TypoScript-optie `config.renderCharset`, die werd gebruikt als tekenset voor interne conversies is verwijderd

Hoofdstuk 6: Bronnen en auteurs

Bronnen en auteurs

Bronnen

TYPO3-Nieuws:

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

Publicatie-informatie:

- http://wiki.typo3.org/TYPO3\CMS_8.0.0
- [INSTALL.md](#) en [Changelog](#)
- `typo3/sysex/core/Documentation/Changelog/8.0/*`

TYPO3 Bug-/Issuetracker:

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

TYPO3 en Fluid Git Repositories:

- <https://git.typo3.org/Packages/TYPO3.CMS.git>
- <https://github.com/TYPO3Fluid/Fluid>

Bronnen en auteurs

TYPO3 CMS What's New Team:

Andrey Aksenov, Pierrick Caillon, Sergio Catala, Jigal van Hemert,
Patrick Lobacher, Michel Mix, Sinisa Mitrovic, Angeliki Plati,
Nena Jelena Radovic, Michael Schams en Roberto Torresani

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

Licentie: Creative Commons BY-NC-SA 3.0

