

# **TYPO3 CMS 7 LTS – What's New**

## **Extbase & Fluid**

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)

# Extbase & Fluid

Extbase en Fluid zijn de de facto standaarden op het gebied van moderne technieken binnen TYPO3 van de dag. Hoewel Extbase nieuwe functies gekregen heeft zijn de meeste wijzigingen in TYPO3 CMS 7 LTS gerelateerd aan Fluid.

Een aantal ViewHelpers is bijgewerkt, verbeterd of toegevoegd.

Bijvoorbeeld de **MediaViewHelper** die gebruikt kan worden om video, audio en andere bestandstypen in de frontend weer te geven.

De systeemextensie `form` is aangepast om Extbase/Fluid te ondersteunen.

## Alternatief pad voor sjabloon

- Fluid Standalone View en het TypoScript-objekt FLUIDTEMPLATE ondersteunen alternatieve paden voor sjablonen

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:myextension/Resources/Private/Templates/Main.html
page.10.partialRootPaths {
    10 = EXT:myextension/Resources/Private/Partials
    20 = EXT:fallback/Resources/Private/Partials
}
```

- Als de nieuwe en oude opties gebruikt worden (bijv. `partialRootPaths` and `partialRootPath`), wordt het pad op de eerste positie van de optie (index = 0) gebruikt.

## Typolink-ViewHelper

- Een nieuwe Typolink-ViewHelper accepteert een `typolink`-tekenreeks zoals gemaakt door de Link-wizard en de RTE

```
<f:link.typolink parameter="{link}" target="_blank" class="ico-class" title="some title"
    additionalAttributes="{type:'button'}">
```

`link` kan bevatten:

```
42 _blank - "Dit is de titel van de link" &foo=bar
```

Uitvoer:

```
<a href="index.php?id=42&foo=bar" title="Dit is de titel" target="_blank" class="ico-class" type
    ="button">
```

Let op: alleen `parameter` is vereist, rest is optioneel

# Extbase & Fluid

---

## Algemene data-\*-attribuut

- Alle ViewHelpers die HTML tags produceren ondersteunen de HTML5 data-\* attribuut
- Een array doorgegeven als data wordt omgezet en de sleutel/waarde combinaties vormen het attribuut: data-key="value"

### Voorbeeld:

```
<f:form.textfield data="{foo: 'bar', baz: 'foos'}" />
```

### Uitvoer:

```
<input data-foo="bar" data-baz="foos" ... />
```

## Waarden van klasse-tags via reflectie

- Extbase Reflection Service kan de tags en annotaties teruggeven die toegevoegd zijn aan een klasse

### Voorbeeld:

```
/**
 * @EenKlasseAnnotatie Een waarde
 */
class MyClass {
}
```

### Annotatie te benaderen via:

```
$service = new \TYPO3\CMS\Extbase\Reflection\ReflectionService();

// Geeft alle tags en hun waarden waarmee de betreffende klasse is getagt
$classValues = $service->getClassTagsValues('MyClass');

// Geeft de waarden van de gespecificeerde klasse-tag
$classValue = $service->getClassTagValue('MyClass', 'EenKlasseAnnotatie');
```

# Extbase & Fluid

---

## PaginateViewHelper

- Vanaf TYPO3 CMS 7.1 accepteert de PaginateViewHelper de volgende waardetypen:
  - QueryResultInterface
  - ObjectStorage
  - ArrayAccess
  - array
- Voorbeeld:

```
<f:widget.paginate objects="{blogs}" as="paginatedBlogs">
  <f:for each="{paginatedBlogs}" as="blog">
    <h4>{blog.title}</h4>
  </f:for>
</f:widget.paginate>
```



## ContainerViewHelper laadt RequireJS-modulen

- ContainerViewHelper kan RequireJS-modulen laden via het `includeRequireJsModules`-attribuut
- Voorbeeld:

```
<f:be.container pageTitle="Extension Module" loadJQuery="true"
  includeRequireJsModules="{
    0:'TYPO3\CMS\Extension\Module1',
    1:'TYPO3\CMS\Extension\Module2',
    2:'TYPO3\CMS\Extension\Module3',
    3:'TYPO3\CMS\Extension\Module4'
  }" >
```

# Extbase & Fluid

---

## Methode `has()` in `ObjectAccess`

- Voor het gebruik in Fluid ondersteunen `object.property` en `object.isProperty` reeds de volgende methoden:
  - `isProperty()`
  - `getProperty()`
- Nieuw vanaf TYPO3 CMS 7.1: `hasProperty()`
- Wanneer in Fluid `object.hasProperty` wordt gebruikt, wordt de methode `$object->hasProperty()` aangeroepen

## Upload van meerdere bestanden met de FormUpload-ViewHelper

- De FormUpload-Viewhelper ondersteunt nu een nieuw attribuut `multiple`, die het mogelijk maakt om meerdere bestanden tegelijkertijd te uploaden

```
<f:form.upload property="files" multiple="multiple" />
```

**Let op: ontwikkelaars moeten een eigen TypeConverter maken voor de property-mapping van de binnenkomende waarden!**

## Callouts vs. FlashMessages

- Callouts (content-info-blokken) vervangen FlashMessages op diverse plaatsen in de backend om contextuele informatie af te beelden
- Dit vereist een nieuwe Fluid ViewHelper `be.infobox`:

```
<f:be.infobox title="Titel van bericht">
    inhoud van blok
</f:be.infobox>
```

```
<f:be.infobox
    title="Titel van bericht"
    message="inhoud van blok"
    state="-2"
    iconName="check"
    disableIcon="TRUE" />
```

## format.case ViewHelper

- Nieuwe ViewHelper `format.case` wijzigt het hoofdlettergebruik van teksten:
  - `upper`: zet een tekst om naar "HOOFDLETTERS"
  - `lower`: zet een tekst om naar "kleine letters"
  - `capital`: zet een tekst om naar begin met hoofdletter
  - `uncapital`: zet een tekst om naar begin met kleine letter
- Voorbeeld:

```
// omzetten naar "EEN TEKST MET HOOFD- EN KLEINE LETTERS"  
<f:format.case>Een TeKsT mEt HoOfD- eN kLeInE lEtTeRs</f:format.case>
```

```
// omzetten naar "EenTekst"  
<f:format.case mode="capital">eenTekst</f:format.case>
```

# Extbase & Fluid

---

## Allerlei

- Parameter `cHash` wordt niet toegevoegd aan actie-URI's als de huidige aanroep niet gecached is en de doelactie niet gecached kan worden

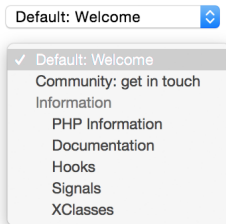
## ActionMenuItemGroupViewHelper (1)

- Met deze ViewHelper kunnen groepen met opties gebruikt worden in een selectievakje in de backend dat bepaalt welke actie is gekozen
- Example:

```
<f:be.menus.actionMenu>
  <f:be.menus.actionMenuItem label="Standaard: Welkom" controller="Default" action="index" />
  <f:be.menus.actionMenuItem label="Community: neem contact op" controller="Community"
    action="index" />
  <f:be.menus.actionMenuItemGroup label="Informatie">
    <f:be.menus.actionMenuItem label="PHP Informatie" controller="Information"
      action="listPhpInfo" />
    <f:be.menus.actionMenuItem label="Documentatie" controller="Information"
      action="documentation" />
    <f:be.menus.actionMenuItem label="Hooks" controller="Information" action="hooks" />
    <f:be.menus.actionMenuItem label="Signalen" controller="Information" action="signals" />
    <f:be.menus.actionMenuItem label="XClasses" controller="Information" action="xclass" />
  </f:be.menus.actionMenuItemGroup>
</f:be.menus.actionMenu>
```

## ActionMenuItemGroupViewHelper (2)

- Voorbeeld op vorige pagina resulteert in de volgende uitvoer:





## Sjabloonondersteuning voor FlashMessagesViewHelper

- De FlashMessagesViewHelper ondersteunt sjablonen
- Met het nieuwe attribuut `as` kan een variabelenaam ingesteld worden die gebruikt kan worden om binnen de onderliggende elementen van de ViewHelper de flash messages te benaderen
- Voorbeeld:

```
<f:flashMessages as="flashMessages">
  <ul class="myFlashMessages">
    <f:for each="{flashMessages}" as="flashMessage">
      <li class="alert {flashMessage.class}">
        <h4>{flashMessage.title}</h4>
        <span class="fancy-icon">{flashMessage.message}</span>
      </li>
    </f:for>
  </ul>
</f:flashMessages>
```

- Let op: optie `renderMode` is deprecated

## Nieuwe eigenschappen van cObject FLUIDTEMPLATE (1)

- cObject FLUIDTEMPLATE is uitgebreid met `templateRootPaths` en `templateName`
- Het is mogelijk om de naam van een sjabloon in te stellen en bij het renderen wordt dit sjabloon gebruikt samen met het ingesteld formaat om het sjabloon te vinden in de opgegeven `templateRootPaths`
- `templateRootPaths` biedt dezelfde fallback logica als `layoutRootPath` en `partialRootPath`
  - `templateName`: `string/stdWrap`
  - `templateRootPaths`: array van bestandspaden met ondersteuning voor "EXT:" prefix

## Nieuwe eigenschappen van cObject FLUIDTEMPLATE (2)

### ■ TypeScript voorbeeld:

```
lib.stdContent = FLUIDTEMPLATE
lib.stdContent {
    templateName = TEXT
    templateName.stdWrap {
        cObject = TEXT
        cObject {
            data = levelfield:-2,backend_layout_next_level,slide
            override.field = backend_layout
            split {
                token = frontend__
                1.current = 1
                1.wrap = |
            }
        }
    }
    ifEmpty = Default
}
templateRootPaths {
    10 = EXT:frontend/Resources/Private/Templates
    20 = EXT:sitemodification/Resources/Private/Templates
}
}
```

# Extbase & Fluid

---

## Verwijderen van `xmlns`-attributen en HTML Tags (1)

- Met de introductie van `xmlns:*` attributen om ViewHelpers in te voegen, is het mogelijk voor IDE's om Fluid sjablonen te ondersteunen. Het probleem is dat `xmlns:*` attributen en de overeenkomstige tag ook worden weergegeven wat meestal ongewenst is.
- Als workaround kunnen sections gebruikt worden, maar deze oplossing is niet vanzelfsprekend en niet beschikbaar in layouts. Ook geeft het meer werk bij de verwerking.
- `xmlns:*` attributen voor valide ViewHelper namespaces zullen nu verwijderd worden voordat het gerenderd wordt als ze de volgende syntax hebben: `http://typo3.org/ns/<phpNamespace>` (`xmlns` attributen voor namespaces die niet voor ViewHelpers zijn blijven bewaard)

## Verwijderen van `xmlns`-attributen en HTML Tags (2)

- Neem ViewHelper namespaces op in de HTML tag en het `data-namespace-typo3-fluid="true"` attribuut om te voorkomen dat de hele HTML tag gerenderd wordt

```
<html data-namespace-typo3-fluid="true"
  xmlns:f="http://typo3.org/ns/TYPO3/CMS/Fluid/ViewHelpers"
  xmlns:n="http://typo3.org/ns/GeorgRinger/News/ViewHelpers">

  <f:if condition="{newsItem.title}">
    <f:then>
      <n:titleTag>{newsItem.title}</n:titleTag>
    </f:then>
    <f:else>
      <n:titleTag>Nieuwsdetail</n:titleTag>
    </f:else>
  </f:if>

</html>
```

## Nieuwe methodes in Fluid-StandaloneView

- StandaloneView is uitgebreid met `setTemplateRootPaths($templatePaths)` en `setTemplate($templateName, $throwException = TRUE)`
- Zelfde functionaliteit als `cObject FLUIDTEMPLATE`
- Voorbeeld (maak e-mail van een sjabloon):

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

# Extbase & Fluid

---

## Gegevensverwerking voor FLUIDTEMPLATE cObject (1)

- cObject FLUIDTEMPLATE is uitgebreid met dataProcessing
- Deze instelling kan gebruikt worden om een of meerdere processors toe te voegen voor het bewerken van de \$data variabele van het cObject dat momenteel afgehandeld wordt (bijv. tt\_content of page)
- Processor moet de interface FluidTemplateDataProcessorInterface implementeren moet de volgende functie bevatten:

```
function process(array &$data, array $processorConfiguration,  
    array $configuration, StandaloneView $view) {  
    [...]  
}
```

## Gegevensverwerking voor FLUIDTEMPLATE cObject (2)

### ■ Voorbeeld:

```
my_custom_ctype = FLUIDTEMPLATE
my_custom_ctype {
    templateRootPaths {
        10 = EXT:your_extension_key/Resources/Private/Templates
    }
    templateName = CustomName
    settings {
        extraParam = 1
    }
    dataProcessing {
        1 = Vendor\YourExtensionKey\DataProcessing\MyFirstCustomProcessor
        2 = AnotherVendor\AnotherExtensionKey\DataProcessing\MySecondCustomProcessor
        2 {
            options {
                myOption = SomeValue
            }
        }
    }
}
```



## Fragment voor Paginatie-widget

- Met deze feature kan een item `section` aan de configuratie van een Fluid paginatie widget toegevoegd worden
- Het fragment wordt toegevoegd aan elke link in de paginering
- De volgende code voegt het fragment `#archive` toe:

```
<f:widget.paginate objects="{plantpestWarnings}" as="paginatedWarnings"  
  configuration="{section: 'archive', itemsPerPage: 10, insertAbove: 0, insertBelow: 1,  
  maximumNumberOfLinks: 10}">
```

```
[...]
```

```
</f:widget.paginate>
```

# Extbase & Fluid

---

## Attribuut base voor DateViewHelper

- De DateViewHelper is uitgebreid met een optioneel attribuut base
- Het attribuut definieert de basis voor relatieve datumaanduidingen
- Als de datum een DateTime object is wordt base genegeerd
- Het volgende voorbeeld geeft "2016", als dateObject een datum is in 2017:

```
<f:format.date format="Y" base="{dateObject}">-1 year</f:format.date>
```

(zie [PHP documentatie](#) voor geldige waarden)

## Optie dataProcessing voor FLUIDTEMPLATE

- In TYPO3 CMS 7.3 was de optie dataProcessing voor cObject FLUIDTEMPLATE toegevoegd
- De FluidTemplateDataProcessorInterface is herbouwd naar DataProcessorInterface, wat ook van invloed is op de functie process()

```
public function process(  
    ContentObjectRenderer $cObj,  
    array $contentObjectConfiguration,  
    array $processorConfiguration,  
    array $processedData  
);
```

**BREAKING CHANGE!**

## Filter voor FlashMessageQueue

- Tot TYPO3 CMS 7.5 werden alle berichten van de FlashMessageQueue opgehaald of verwijderd
- Vanaf TYPO3 CMS 7.5 kan men de berichten filteren op severity (ernst):

```
FlashMessageQueue::getAllMessages($severity);  
FlashMessageQueue::getAllMessagesAndFlush($severity);  
FlashMessageQueue::removeAllFlashMessagesFromSession($severity);  
FlashMessageQueue::clear($severity);
```

# Extbase & Fluid

---

## Ondersteuning voor "between" in query's

- Er is ondersteuning toegevoegd voor het gebruik van `between` in de Extbase Query object
- Dit heeft geen voordelen voor de prestaties aangezien databases "between" intern sowieso al converteren naar:  
`min <= expr AND expr <= max`
- De nieuwe Extbase-feature repliceert het gedrag van de databases door gebruik te maken van een logische AND-conditie, dus dit werkt in alle databases

```
$query->matching(  
    $query->between('uid', 3, 5)  
);
```

## Meerdere FlashMessageQueues

- Het is nu mogelijk om meerdere FlashMessageQueues te implementeren:

```
$queueIdentifier = 'myQueue';  
$this->controllerContext->getFlashMessageQueue($queueIdentifier);
```

- Dit kan als volgt in Fluid worden gebruikt:

```
<f:flashMessages queueIdentifier="myQueue" ></f:flashMessages>
```

## MediaViewHelper (1)

- Voor de eenvoudige weergave van video, audio en andere bestandstypen met een Renderer, is de MediaViewHelper beschikbaar gemaakt
- MediaViewHelper kijkt eerst of er een Renderer beschikbaar is voor het betreffende bestand. Zo niet, dan wordt er teruggevallen op een image-tag
- Voorbeelden:

```
<code title="Image Object">  
  <f:media file="{file}" width="400" height="375" ></f:media>  
</code>
```

```
<output>  
    
</output>
```

## MediaViewHelper (2)

### ■ Voorbeelden (vervolg):

```
<code title="MP4 Video Object">
  <f:media file="{file}" width="400" height="375" ></f:media>
</code>
```

```
<output>
  <video width="400" height="375" controls>
    <source src="fileadmin/user_upload/my-video.mp4" type="video/mp4">
  </video>
</output>
```

```
<code title="MP4 Video Object with loop and autoplay option set">
  <f:media file="{file}" width="400" height="375"
    additionalConfig="{loop: '1', autoplay: '1'}" ></f:media>
</code>
```

```
<output>
  <video width="400" height="375" controls loop>
    <source src="fileadmin/user_upload/my-video.mp4" type="video/mp4">
  </video>
</output>
```



# Extbase & Fluid

---

## Systemextensie `form` (1)

- Systemextensie `form` (inclusief het eigen datamodel, de controllers, validatie en sjablonen) is aangepast aan het MVC-model van Extbase/Fluid
- Doordat de weergave nu volledig op Fluid is gebaseerd, is het eenvoudiger om eigen sjablonen en ViewHelpers te gebruiken
- Elk `form`-element heeft een eigen Partial, die kan worden geconfigureerd met de TypoScript-optie `partialPath = ...`

# Extbase & Fluid

---

## Systemextensie form (2)

- Er zijn 3 nieuwe ViewHelpers beschikbaar:
  - AggregateSelectOptionsViewHelper (voor optgroup-tags)
  - SelectViewHelper (voor optgroup-tags)
  - PlainMailViewHelper (voor het renderen van e-mail in platte tekst)
- Bovendien zijn er 3 Views:
  - show (het formulier zelf)
  - confirmation (de bevestigingspagina)
  - postProcessor/mail (de e-mail)
- Sjabloonpaden en de zichtbaarheid van velden kunnen voor elke View afzonderlijk worden ingesteld

# Extbase & Fluid

---

## Annotatie @cli

- Door gebruik te maken van de nieuwe annotatie @cli, kan worden aangegeven dat commando's in Extbase-CommandControllers alleen gebruikt mogen worden vanaf de command-line-interface
- Deze commando's kunnen niet als taak worden geselecteerd in de scheduler
- Typische use-cases zijn commando's zoals `extbase:help:help`

## Relatie met dezelfde tabel

- Er kan nu in een domeinmodel direct een relatie gemaakt worden met een object van dezelfde klasse

```
namespace \Vendor\Extension\Domain\Model;
class A {
    /**
     * @var \Vendor\Extension\Domain\Model\A
     */
    protected $parent;
}
```

```
namespace \Vendor\Extension\Domain\Model;
class A {
    /**
     * @var \Vendor\Extension\Domain\Model\B
     */
    protected $x;

    /**
     * @var \Vendor\Extension\Domain\Model\B
     */
    protected $y;
}
```

## Optie absolute voor Image-ViewHelpers

- Nieuwe optie absolute forceert ImageViewhelper en Uri/ImageViewHelper om een **absolute** URL te geven
- Voorbeeld 1 (ImageViewhelper):

```
<f:image image="{file}" width="400" height="375" absolute="1" ></f:image>
```

```
// Output
```

```

```

- Voorbeeld 2 (Uri/ImageViewHelper):

```
<f:uri.image image="{file}" width="400" height="375" absolute="1" ></f:uri>
```

```
// Uitvoer
```

```
http://example.com/fileadmin/_processed_/323223424.png
```

# Extbase & Fluid

---

## Verwijder witruimte tussen HTML-tags

- Nieuwe ViewHelper `spaceless` verwijdt overbodige spaties tussen HTML tags terwijl de witruimte binnen HTML tags bewaard blijft:

```
<f:spaceless>
<div>
  <div>
    <div>tekst

tekst</div>
</div>
</div>
```

- Uitvoer:

```
<div><div><div>tekst

tekst</div></div></div>
```

# Extbase & Fluid

---

## Configuratie op root-niveau

- Het root-niveau van een tabel kan nu in de TCA ingesteld worden (dit bepaalt waar in het systeem records van een tabel gevonden kunnen worden)
  - 0: alleen paginaboom
  - 1: alleen op de rootpagina (PID 0)
  - -1: beide, rootpagina en paginaboom
- TCA configuratie:

```
$GLOBALS['TCA']['tx_myext_domain_model_record']['ctrl']['rootLevel'] = -1;
```

## 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/sysexst/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

