

# **TYP03 CMS 7.5 LTS – What's New**

Résumé des nouvelles caractéristiques, changements et améliorations

Créé par :

Patrick Lobacher et Michael Schams

Traduction par :

Pierrick Caillon

# TYPO3 CMS 7.5 - What's New

---

## Sommaire

Introduction

Interface Utilisateur Backend

TSConfig & TypoScript

Changements en profondeur

Extbase & Fluid

Fonctions dépréciées et retirées

Sources et Auteurs

## Introduction

## Faits

# Introduction

---

## TYPO3 CMS 7.5 – Faits

- Date de sortie : 29 Septembre 2015
- Type de sortie : "Sprint Release"
- Slogan : Embrace, Innovate, Deliver
- Axe principal : Finalization



# Introduction

---

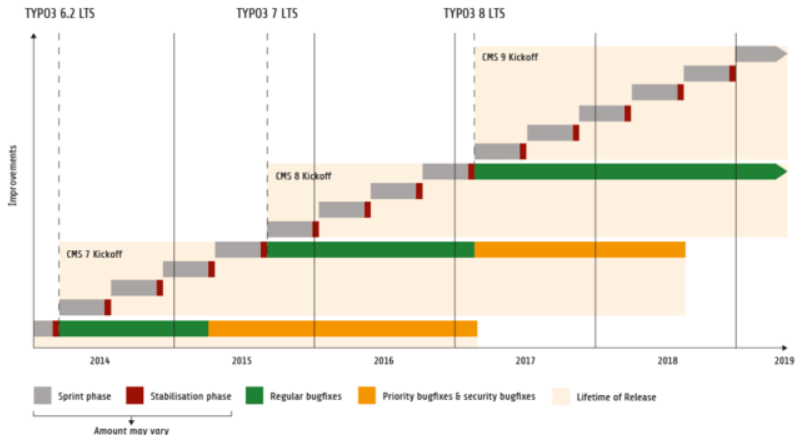
## Prérequis système

- PHP\* : v5.5.0 - v5.6.x
- MySQL : v5.5.x - v5.6.x (pas de mode strict)
- Espace disque : min. 200 Mo
- Configuration PHP :
  - `memory_limit`  $\geq$  128M
  - `max_execution_time`  $\geq$  240s
  - L'option de compilation `-disable-ipv6` NE doit PAS être utilisée
- Le backend nécessite IE  $\geq$  9 ou tout autre navigateur moderne

\*) Plus d'information : [Prérequis PHP minimum pour TYPO3 CMS 7 \(en anglais\)](#)

# Introduction

## Chronologie des développements et sorties



# Introduction

---

## Feuille de route TYPO3 CMS

Dates de sortie estimées et axes principaux :

- v7.0 02/Déc./2014 Backend Overhaul Vol 1
- v7.1 24/Fév./2015 Core Cleanup & Streamlining
- v7.2 28/Avr./2015 Frontend
- v7.3 16/Juin/2015 Package Ecosystem, Composer
- v7.4 04/Août/2015 Backend Overhaul Vol 2
- **v7.5 29/Sep./2015 Finalization**
- v7 LTS Oct./Nov./2015 **TYPO3 CMS 7 LTS** (Long Term Release)

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

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

# Introduction

---

## Installation

- Procédure officielle d'installation sous Linux/Mac OS X (DocumentRoot considéré /var/www/site/htdocs):

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/7.5
$ tar xzf typo3_src-7.5.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.5.0 typo3_src
$ ln -s typo3_src/index.php
$ ln -s typo3_src/typo3
$ touch FIRST_INSTALL
```

- Liens symboliques sous Microsoft Windows :
  - Utiliser junction sous Windows XP/2000
  - Utiliser mklink sous Windows Vista et Windows 7



# Introduction

---

## Mise à jour vers TYPO3 CMS 7.x

- Les mises à jour sont possibles seulement depuis TYPO3 CMS 6.2 LTS
- TYPO3 CMS < 6.2 doivent être mis à jour vers la 6.2 LTS en premier
- Instructions de mise à jour :  
[http://wiki.typo3.org/Upgrade#Upgrading\\_to\\_7.5](http://wiki.typo3.org/Upgrade#Upgrading_to_7.5)
- Guide TYPO3 officiel « TYPO3 Installation and Upgrading » :  
<http://docs.typo3.org/typo3cms/InstallationGuide>
- De manière générale :
  - Vérifier les prérequis système (PHP, MySQL, etc.)
  - Examiner **deprecation\_\*.log** de l'ancienne instance TYPO3
  - Mettre à jour toutes les extensions vers leurs dernières versions
  - Déployer les nouvelles sources et exécuter l'assistant de mise à jour de l'Install Tool
  - Examiner le module de démarrage des utilisateurs backend (optionnel)

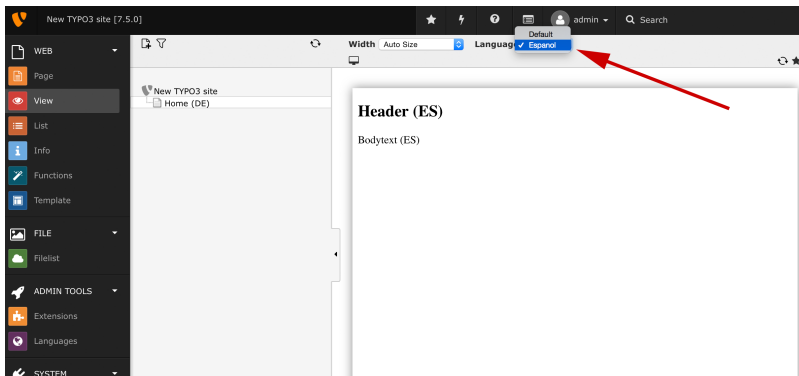
## Chapitre 1 :

# Interface Utilisateur Backend

# Interface Utilisateur Backend

## Sélection de la langue dans le module "Voir"

WEB→Voir Sélection de la langue pour la prévisualisation de la page.  
(se désactive à l'aide de `mod.SHARED.view.disableLanguageSelector = 1`)



# Interface Utilisateur Backend

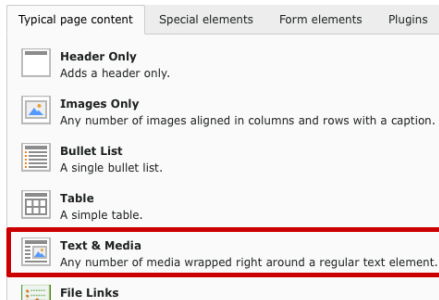
---

## Élément de contenu `textmedia`

Un nouvel élément de contenu **"Text & Media"** combine les éléments `text`, `image` et `textpic`.

### New content element

Please select the type of page content you wish to create:



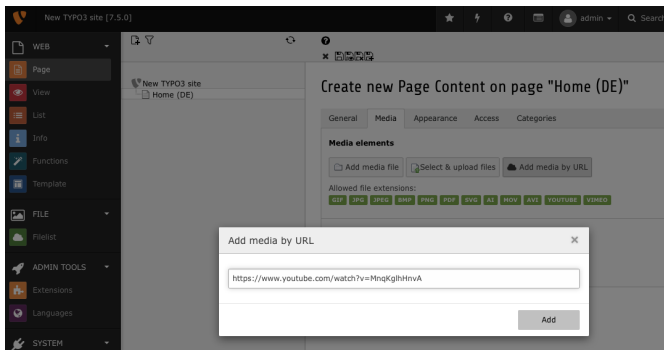
The screenshot shows a selection interface for content elements. At the top, there are four tabs: "Typical page content", "Special elements", "Form elements", and "Plugins". The "Typical page content" tab is selected. Below the tabs, a list of content elements is displayed, each with an icon and a description:

- Header Only**: Adds a header only.
- Images Only**: Any number of images aligned in columns and rows with a caption.
- Bullet List**: A single bullet list.
- Table**: A simple table.
- Text & Media**: Any number of media wrapped right around a regular text element. (This element is highlighted with a red border in the image.)
- File Links**

# Interface Utilisateur Backend

## Fichiers YouTube et Vimeo

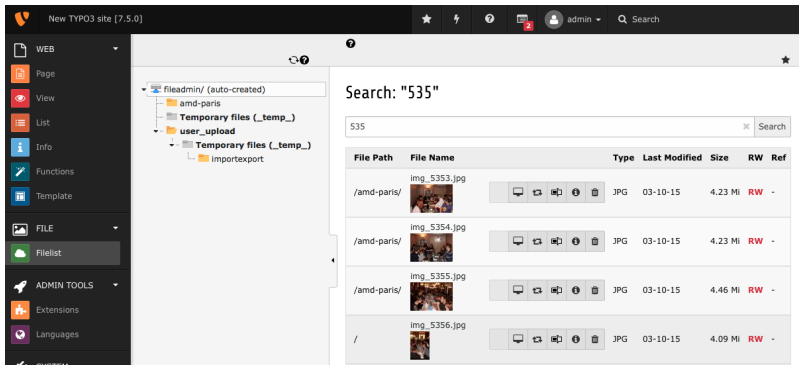
L'élément de contenu **"Text & Media"** permet aux éditeurs d'ajouter des fichiers externes YouTube et Vimeo, en plus des fichiers locaux.



# Interface Utilisateur Backend

## Recherche dans le module liste des fichiers

Le module liste des fichiers permet de rechercher par nom de fichier (récursivement depuis le dossier actuel).



The screenshot displays the TYPO3 Backend Filelist interface. The top navigation bar shows the site name "New TYPO3 site [7.5.0]", the user "admin", and a search icon. The left sidebar contains various modules like "WEB", "FILE", and "ADMIN TOOLS". The main area shows a file tree with "fileadmin/ (auto-created)" expanded to "amd-paris". A search bar at the top of the file list contains the text "535". Below the search bar, a table lists the search results.

File Path	File Name	Type	Last Modified	Size	RW	Ref
/amd-paris/	img_5353.jpg	JPG	03-10-15	4.23 Mi	RW	-
/amd-paris/	img_5354.jpg	JPG	03-10-15	4.23 Mi	RW	-
/amd-paris/	img_5355.jpg	JPG	03-10-15	4.46 Mi	RW	-
/	img_5356.jpg	JPG	03-10-15	4.09 Mi	RW	-

## Chapitre 2 : TSconfig & TypoScript

# Tsconfig & TypoScript

---

## Conditions pour les inclusions TypoScript

- La propriété supplémentaire « condition » est ajoutée à INCLUDE\_TYPOSCRIPT, permettant de n'inclure le fichier ou dossier seulement lorsque la condition est respectée

```
// only include TypoScript, if current user is logged in:  
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypoScript/feuser.ts"  
  condition="[loginUser = *]">
```

```
// include TypoScript depending on application context:  
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypoScript/staging.ts"  
  condition="applicationContext = /^Production\\/Staging\\/Server\\d+$/">
```



# Tsconfig & TypoScript

---

## Option TCA : Afficher l'âge

- L'option TCA `disableAgeDisplay` désactive l'affichage de l'âge (par exemple : "2015-08-30 (-27 jours)")

```
$GLOBALS['TCA']['tt_content']['columns']['date']['config']['disableAgeDisplay'] = true;
```

- Bien évidemment, le type du champ doit être `input` et `eval` doit être défini à `date`

# TSconfig & TypoScript

---

## Configurer les libellés de langue exportés en TypoScript (1)

- Les fichiers de langue XLF peuvent être lus et exportés dans un tableau
- Ceci permet d'accéder aux libellés de langue en JavaScript, par exemple
- Les trois paramètres optionnels suivants sont supportés :
  - `selectionPrefix` :  
seul les identifiants de libellé commençant par ce préfix sont pris en compte
  - `stripFromSelectionName` :  
chaîne retirée des identifiants de libellé
  - `errorMode` :  
mode d'erreur si le fichier n'est pas trouvé :  
0 : entrée syslog (par défaut), 1 : ignorer, 3 : lever une exception

## Configurer les libellés de langue exportés en TypoScript (2)

### ■ Exemple :

```
page = PAGE
page.inlineLanguageLabelFiles {
    someLabels = EXT:myExt/Resources/Private/Language/locallang.xlf
    someLabels.selectionPrefix = idPrefix
    someLabels.stripFromSelectionName = strip_me
    someLabels.errorMode = 2
}
```

### ■ Sortie :

```
<script type="text/javascript">
/**/
    var TYPO3 = TYPO3 || {};
    TYPO3.lang = {"firstLabel":[{"source":"first Label","target":"erstes Label"}],
        "secondLabel":[{"source":"second Label","target":"zweites Label"}]};
/*]]&gt;*/
&lt;/script&gt;</pre></div><div data-bbox="22 936 234 958" data-label="Page-Footer"><p>TYPO3 CMS 7.5 LTS - What's New</p></div><div data-bbox="806 886 970 951" data-label="Page-Footer"><img alt="TYPO3 logo" data-bbox="806 886 970 951"/>The logo for TYPO3, featuring an orange stylized 'T' icon to the left of the text 'TYPO3' in a bold, black, sans-serif font.</div>
```

## Prévisualisation dans les espaces de travail par TSconfig

- TYPO3 CMS génère des liens de prévisualisation seulement pour les tables `tt_content`, `pages` et `pages_language_overlay` par défaut
- La configuration est possible en TSconfig de page :

```
# use page 123 for previewing workspaces records (in general)
options.workspaces.previewPageId = 123
```

```
# use the pid field of each record for previewing (in general)
options.workspaces.previewPageId = field:pid
```

```
# use page 123 for previewing workspaces records (for table tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = 123
```

```
# use the pid field of each record for previewing (for table tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = field:pid
```

## Qualité des images dans `sourceCollection`

- La spécification de la qualité des images des entrées `sourceCollection` est ajoutée
- L'option est prioritaire sur la configuration de l'Install Tool (enregistrée dans le fichier `LocalConfiguration.php`)
- Exemple :

```
# for small retina images
tt_content.image.20.1.sourceCollection.smallRetina.quality = 80

# for large retina images
tt_content.image.20.1.sourceCollection.largeRetina.quality = 65
```

# Tsconfig & TypeScript

---

## Compte des éléments d'une liste

- La propriété `returnCount` est ajoutée à la propriété `stdWrap split`
- Elle permet de récupérer le nombre d'éléments d'une liste ayant la virgule comme séparateur
- Par exemple, le code suivant retourne 9 :

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

# TScnfig & TypoScript

---

## Prise en charge des dispositions backend (1)

- La prise en charge des dispositions backend a été simplifiée pour le frontend
- La nouvelle option pagelayout est utilisable en TypoScript
- Exemple :

```
page.10 = FLUIDTEMPLATE
page.10 {
    file.stdWrap.cObject = CASE
    file.stdWrap.cObject {
        key.data = pagelayout
        default = TEXT
        default.value = EXT:sitepackage/Resources/Private/Templates/Home.html
        3 = TEXT
        3.value = EXT:sitepackage/Resources/Private/Templates/1-col.html
        4 = TEXT
        4.value = EXT:sitepackage/Resources/Private/Templates/2-col.html
    }
}
```

(continue sur la page suivante)

# TScnfig & TypoScript

---

## Prise en charge des dispositions backend (2)

- ... où `key.data = pagelayout` remplace le code suivant :

```
field = backend_layout
ifEmpty.data = levelfield:-2,backend_layout_next_level,slide
ifEmpty.ifEmpty = default
```



# Tsconfig & TypoScript

---

## Divers

- La fonction `stdWrap bytes` a été introduite par TYPO3 CMS 7.4
- La possibilité de définir la base est ajoutée par TYPO3 CMS 7.5, permettant de définir s'il on veut calculer avec la base 1000 ou 1024

```
bytes.labels = " | K| M| G"
```

```
bytes.base = 1000
```

## Chapitre 3 : Changements en profondeur

# Changements en profondeur

---

## Fluid-based Content Elements (1)

- La nouvelle extension système "**Fluid-based Content Elements**" est implémentée
- Des modèles Fluid sont utilisés pour le rendu des éléments de contenu à la place du TypoScript
- Elle pourra être une alternative à *CSS Styled Content* dans le futur
- Les gabarits statiques suivants sont à inclure pour utiliser la fonctionnalité :
  - Content Elements (`fluid_styled_content`)
  - Content Elements CSS (optional) (`fluid_styled_content`)

# Changements en profondeur

---

## Fluid-based Content Elements (2)

- De plus le gabarit TSconfig de page suivant doit être inclus :  
Fluid-based Content Elements (fluid\_styled\_content)
- Surchargez les modèles par défaut en ajoutant vos propres chemins en configuration TypoScript :

```
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/
```

# Changements en profondeur

---

## Fluid-based Content Elements (3)

- Migration de *CSS Styled Content* vers *Fluid-based Content Elements* :
  - Désinstaller l'extension `css_styled_content`
  - Installer l'extension `fluid_styled_content`
  - Utiliser l'assistant de migration de l'Install Tool pour migrer les éléments `text`, `image` et `textpic` vers `textmedia`

Note : "*Fluid-based Content Elements*" est dans un stage préliminaire et des changements importants peuvent toujours avoir lieu avant TYPO3 CMS 7 LTS. Des conflits avec *CSS Styled Content* peuvent toujours exister.

# Changements en profondeur

---

## Méthode `SELECTmmQuery`

- La méthode `SELECT_mm_query` est ajoutée à la classe `DatabaseConnection`
- Extraite de `exec_SELECT_mm_query` pour séparer la construction et l'exécution des requêtes M:M.
- Permettant l'usage de la construction de requête dans la couche d'abstraction des bases de données.

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

# Changements en profondeur

---

## Optimiser les tables MySQL

- Nouvelle tâche du planificateur pour exécuter la commande MySQL `OPTIMIZE TABLE` sur les tables sélectionnées
- La commande réorganise physiquement le stockage des données des tables et de leurs indexes afin de réduire l'occupation et améliorer les performances d'E/S
- Ces types de tables sont supportés :  
MyISAM, InnoDB et ARCHIVE
- L'utilisation de cette tâche avec DBAL et d'autre SGBD n'est pas supportée car la commande est spécifique à MySQL

**Note :** L'optimisation des tables est coûteuse en performances d'E/S. Aussi, avant MySQL 5.6.17 le processus verrouille les tables pendant son exécution, pouvant impacter le site.

# Changements en profondeur

---

## Gestion des contenus en ligne (1)

- Les contenus externes (online media) sont supportés par défaut
- Comme exemples, le support des vidéos YouTube et Vimeo est ajouté au noyau
- Les ressources s'ajoutent en tant qu'URL dans le contenu **"Text & Media"**, par exemple
- La classe d'assistance correspondante récupère les métadonnées et fournit l'image de vignette si disponible



# Changements en profondeur

---

## Gestion des contenus en ligne (2)

Les motifs d'URL suivants sont disponibles :

### 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>`

# Changements en profondeur

---

## Gestion des contenus en ligne (3)

- L'accès aux ressources par Fluid s'effectue comme suit :

```
<!-- enable js api and set no-cookie support for YouTube videos -->  
<f:media file="{file}" additionalConfig="{enablejsapi:1, 'no-cookie': true}" ></f:media>  
  
<!-- show title and uploader for YouTube and Vimeo before video starts playing -->  
<f:media file="{file}" additionalConfig="{showinfo:1}" ></f:media>
```

- Options de configuration pour les vidéos YouTube :  
autoplay, controls, loop, enablejsapi, showinfo, no-cookie
- Options de configuration pour les vidéos Vimeo :  
autoplay, loop, showinfo

# Changements en profondeur

---

## Gestion des contenus en ligne (4)

- Pour inscrire votre propre gestionnaire de contenu en ligne, vous avez besoin d'une classe `OnlineMediaHelper` implémentant `OnlineMediaHelperInterface` et une classe `FileRenderer` implémentant `FileRendererInterface`

```
// register your own online video service (the used key is also the bind file extension name)
$GLOBALS['TYPO3_CONF_VARS']['SYS']['OnlineMediaHelpers']['myvideo'] =
    \MyCompany\Myextension\Helpers\MyVideoHelper::class;

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

// register an custom mime-type for your videos
$GLOBALS['TYPO3_CONF_VARS']['SYS']['FileInfo']['fileExtensionToMimeType']['myvideo'] =
    'video/myvideo';

// register your custom file extension as allowed media file
$GLOBALS['TYPO3_CONF_VARS']['SYS']['mediafile_ext'] .= ',myvideo';
```

# Changements en profondeur

---

## Routage Backend

- Un nouveau composant de routage est ajouté au backend de TYPO3 prenant en charge l'adressage des différents appels et modules de TYPO3 CMS
- Les routes sont à définir dans la classe suivante :

Configuration/Backend/Routes.php

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

- Les méthodes appelées contiennent des objets de requête et réponse compatible avec PSR-7 :

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

# Changements en profondeur

---

## Définition de l'auto-chargement dans `ext_emconf.php`

- Les extensions peuvent contenir une ou plusieurs définitions PSR-4 dans le fichier `ext_emconf.php`
- C'était déjà possible dans `composer.json`, mais avec cette nouvelle fonctionnalité, les développeurs d'extension n'ont plus besoin de fournir un fichier `composer` juste pour ça

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

(c'est la méthode d'enregistrement des classes recommandée de TYPO3)

# Changements en profondeur

---

## Nouvelle fabrique d'icônes (1)

- La logique pour travailler avec les icônes, leurs tailles et les icônes de recouvrement est maintenant dans la classe `IconFactory`
- La nouvelle fabrique remplacera l'ancienne API de thème petit à petit
- Tous les icônes du noyau seront enregistrés avec la classe `IconRegistry`
- Les extensions doivent utiliser `IconRegistry::registerIcon()` pour surcharger un icône ou en ajouter un nouveau à la fabrique :

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

# Changements en profondeur

---

## Nouvelle fabrique d'icônes (2)

- Le noyau de TYPO3 CMS implémente trois fournisseurs d'icônes :  
BitmapIconProvider, FontawesomeIconProvider et SvgIconProvider

- Usages d'exemple :

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

- Les valeurs valides pour `Icon::SIZE_...` sont :  
`SIZE_SMALL`, `SIZE_DEFAULT` et `SIZE_LARGE`
- Les valeurs valides pour `Icon::STATE_...` sont :  
`STATE_DEFAULT` and `STATE_DISABLED`

# Changements en profondeur

---

## Nouvelle fabrique d'icônes (3)

- Le noyau de TYPO3 CMS fourni un ViewHelper Fluid permettant l'utilisation facile d'un icône dans une vue Fluid :

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

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

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

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

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



# Changements en profondeur

---

## Hooks et Signals

- Un nouveau signal est ajouté à LinkValidator, permettant des opérations supplémentaires lors de l'initialisation d'un enregistrement (ex. récupérer du contenu depuis la configuration du plugin dans l'enregistrement)
- Enregistrer le signal dans `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'  
);
```

# Changements en profondeur

---

## JumpUrl en extension système (1)

- La génération et la prise en charge des JumpURLs sont déplacées dans l'extension système `jumpurl`
- De nouveaux hooks sont introduits pour permettre la génération et la prise en charge personnalisée (voir page suivante)

**BREAKING CHANGE!**

# Changements en profondeur

---

## JumpUrl en extension système (2)

- Hook 1 : manipuler les **URLs** pendant la génération de liens

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlHandlers']  
    ['myext_myidentifiier']['handler'] = \Company\MyExt\MyUrlHandler::class;  
  
// class needs to implement the UrlHandlerInterface:  
class MyUrlHandler implements \TYPO3\CMS\Frontend\Http\UrlHandlerInterface {  
    ...  
}
```

- Hook 2 : prise en charge des liens **links**

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['urlProcessing']['urlProcessors']  
    ['myext_myidentifiier']['processor'] = \Company\MyExt\MyUrlProcessor::class;  
  
// class needs to implement the UrlProcessorInterface:  
class MyUrlProcessor implements \TYPO3\CMS\Frontend\Http\UrlProcessorInterface {  
    ...  
}
```

# Changements en profondeur

---

## Interface ligne de commande (CLI)

- Appeler `typo3/cli_dispatch.phpsh` en ligne de commande affiche un message d'erreur colorisé si aucune ou une clé invalide a été spécifiée en premier paramètre
- Les contrôleurs de commande d'Extbase peuvent être placés dans des sous-dossiers dans le dossier `Command`
- Exemple :

Controller in file:

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

...can be called via CLI:

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

# Changements en profondeur

---

## Divers (1)

- Les boutons de déplacement du type TCA group peuvent être désactivés explicitement en utilisant l'option `hideMoveIcons = TRUE`
- La méthode `makeCategorizable` est étendue avec un nouveau paramètre `override` pour définir une nouvelle configuration de catégorie pour une combinaison de table et champ existante
- Exemple :

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

Le dernier paramètre (ici : `TRUE`) force la surcharge (par défaut `FALSE`).

# Changements en profondeur

---

## Divers (2)

- Nouvelle fonction pour générer un identifiant unique

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

- Le type de fichier `.typoscript` est ajouté à la liste des types de fichier texte valides
- Une nouvelle option de configuration définit les extensions des fichiers multimédia :

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

**BREAKING CHANGE!**

## Chapitre 4 : Extbase & Fluid

## Filtrage par sévérité dans FlashMessageQueue

- Avant TYPO3 CMS 7.5, tous les messages de la file (FlashMessageQueue) peuvent seulement être tous récupérés ou retirés
- À partir de TYPO3 CMS 7.5, les opérations peuvent être effectuées en filtrant par sévérité :

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



# Extbase & Fluid

---

## Ajout du support du filtre "between"

- Le support de `between` a été ajouté à l'objet Query d'Extbase
- Il n'y a pas différence de performance car le SGBD converti "between" en interne de toute manière : `min <= expr AND expr <= max`
- La nouvelle fonctionnalité Extbase réplique le comportement du SGBD en construisant une condition AND, supportant ainsi tous les SGBD

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

## Files d'attente de FlashMessage multiples

- Le support de multiples files d'attentes FlashMessageQueues est ajouté :

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

- L'accès avec Fluid fonctionne comme indiqué :

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

## ViewHelper Media (1)

- Afin de faciliter le rendu vidéo, audio ou d'autres fichiers avec une classe `Renderer` enregistrée en frontend, le `MediaViewHelper` a été implémenté
- `MediaViewHelper` vérifie d'abord si un `Renderer` est présent pour le fichier donné - dans le cas contraire, il retourne à afficher une balise d'image
- Exemples :

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

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

## ViewHelper Media (2)

### ■ Exemples (suite) :

```
<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

---

## Extension système `form` (1)

- L'extension système `form` (incluant les modèles personnalisés, les contrôleurs, la validation, les vues et l'apparence) a été transformée pour supporter la pile MVC Extbase/Fluid
- Ceci permet une meilleure personnalisation et le contrôle des comportements et de la sortie en modifiant des modèles Fluid ou en utilisant des `view helper` personnalisés
- Chaque élément de formulaire utilise sa propre `Partial`, pouvant être configuré par l'option `TypoScript partialPath = ...`

# Extbase & Fluid

---

## Extension système form (2)

- Les trois ViewHelpers suivants existent :
  - AggregateSelectOptionsViewHelper (for optgroup tags)
  - SelectViewHelper (for optgroup tags)
  - PlainMailViewHelper (to render plain text mails)
- Ainsi que ces trois vues :
  - show (the form itself)
  - confirmation (the confirmation page)
  - postProcessor/mail (the email)
- Le chemin des modèles et la visibilité des champs est personnalisable pour chaque vue individuellement

# Extbase & Fluid

---

## Annotation `@cli`

- En utilisant la nouvelle annotation `@cli`, les commandes d'un `CommandController` Extbase peuvent être marquées pour l'usage en ligne de commande uniquement
- Ces commandes sont exclues de la liste proposée dans le planificateur de tâche
- Les cas d'usage typiques sont les commandes comme `extbase:help:help`, par exemple

## Chapitre 5 : Fonctions dépréciées et retirées



# Fonctions dépréciées et retirées

---

## Méthodes d'échappement dans GeneralUtility

- Les méthodes suivantes de la classe GeneralUtility sont marquées **dépréciées** :
  - `GeneralUtility::addSlashesOnArray()`
  - `GeneralUtility::stripSlashesOnArray()`
  - `GeneralUtility::slashArray()`

# Fonctions dépréciées et retirées

---

## Méthodes et constantes liées à CLI

- Les opérations concernant l'exécution CLI usuelle avec l'option CLIkey ont été déplacées dans le `cliRequestHandler` et donc, les méthodes suivantes sont marquées **dépréciées** :

```
BackendUserAuthentication->checkCLIuser()
```

- Les constantes et paramètres globaux suivants ont aussi été marqués **dépréciés** :

```
const TYPO3_cliKey  
const TYPO3_cliInclude  
$GLOBALS['MCONF']['name']  
$GLOBALS['temp_cliScriptPath']  
$GLOBALS['temp_cliKey']
```

# Fonctions dépréciées et retirées

---

## IconUtility

- L'interface `IconUtilityOverrideResourceIconHookInterface` comme la classe `IconUtility` seront retirés de TYPO3 CMS 8
- Toutes les méthodes publiques de la classe sont marquées **dépréciées** :

```
IconUtility::skinImg()
IconUtility::getSpriteIcon()
IconUtility::getSpriteIconForRecord()
IconUtility::getSpriteIconClasses()
IconUtility::getIcon()
IconUtility::getSpriteIconForFile()
IconUtility::getSpriteIconForResource()
```

(certaines méthodes sont déplacées dans `IconFactory`)

- L'option `TSconfig` de page suivante a aussi été marquée **dépréciée** :

```
mod.wizards.newContentElement.wizardItems.*.elements.*.icon
```

# Fonctions dépréciées et retirées

---

## Méthodes de `HtmlParser` dépréciées

- La fonctionnalité de substitution de marqueurs a été déplacée de `core/Classes/Html/HtmlParser.php` vers sa propre classe `core/Classes/Service/MarkerBasedTemplateService.php`
- Les méthodes suivantes de `HtmlParser` seront retirées de TYPO3 CMS 8 et sont marquées **dépréciées** :

`HtmlParser::getSubpart()`

`HtmlParser::substituteSubpart()`

`HtmlParser::substituteSubpartArray()`

`HtmlParser::substituteMarker()`

`HtmlParser::substituteMarkerArray()`

`HtmlParser::substituteMarkerAndSubpartArrayRecursive()`

# Fonctions dépréciées et retirées

---

## Extension système `form`

- L'extension système `form` se base maintenant sur Extbase/Fluid
- Ainsi, toutes les configurations TypoScript de l'apparence sont marquées **dépréciées** et l'utilisation de `layout` n'est plus recommandée :

```
10 = FORM
10 {
    layout {
        containerWrap = <div><elements ></elements></div>
        elementWrap = <div><element ></element></div>
    }
}
```

# Fonctions dépréciées et retirées

---

## Méthodes et ViewHelpers dépréciés

- Les ViewHelpers suivants sont **dépréciés** :

```
\TYPO3\CMS\Fluid\ViewHelpers\Be\Buttons\IconViewHelper  
\TYPO3\CMS\Backend\ViewHelpers\SpriteManagerIconViewHelper
```

- Les méthodes suivantes sont marquées **dépréciées** :

```
BackendUtility::getExcludeFields()  
BackendUtility::getExplicitAuthFieldValues()  
BackendUtility::getSystemLanguages()  
BackendUtility::getRegisteredFlexForms()  
BackendUtility::exec_foreign_table_where_query()  
BackendUtility::replaceMarkersInWhereClause()
```

# Chapitre 6 : Sources et Auteurs

# Sources et Auteurs

---

## Sources

### Actualités TYPO3 :

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

### Informations des sorties :

- [http://wiki.typo3.org/TYPO3\CMS\\_7.5.0](http://wiki.typo3.org/TYPO3\CMS_7.5.0)
- [INSTALL.md](#) et [ChangeLog](#)
- `typo3/sysexst/core/Documentation/Changelog/7.5/*`

### Suivi des anomalies TYPO3 :

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

### Dépôts Git de TYPO3 :

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



## Diapositives TYPO3 CMS What's New :

Patrick Lobacher

(Recherche et collecte d'informations)

Michael Schams

(Version anglaise/allemande et chef de projet)

### Traduction par :

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

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

Sous licence Creative Commons BY-NC-SA 3.0

