

TYP03 CMS 7.5 – Qué hay Nuevo

Resumen de las nuevas características, cambios y mejoras

Creado por:

Patrick Lobacher y Michael Schams

Traducción en Español por:

Michel Mix y Sergio Catalá

23/October/2015

Creative Commons BY-NC-SA 3.0



TYPO3 CMS 7.5 - Qué hay Nuevo

Resumen de Capítulos

Introducción

Interfaz de Usuario de Backend

TSConfig & TypoScript

Cambios en Profundidad

Extbase & Fluid

Funciones Obsoletas/Eliminadas

Fuentes y Autores

Introducción

Los Hechos

Introducción

TYPO3 CMS 7.5 – Los Hechos

- Fecha de lanzamiento: 29 Septiembre 2015
- Tipo de lanzamiento: "Lanzamiento Sprint"
- Visión: Adoptar, Innovar, Lanzar
- Foco principal: Finalización



Introducción

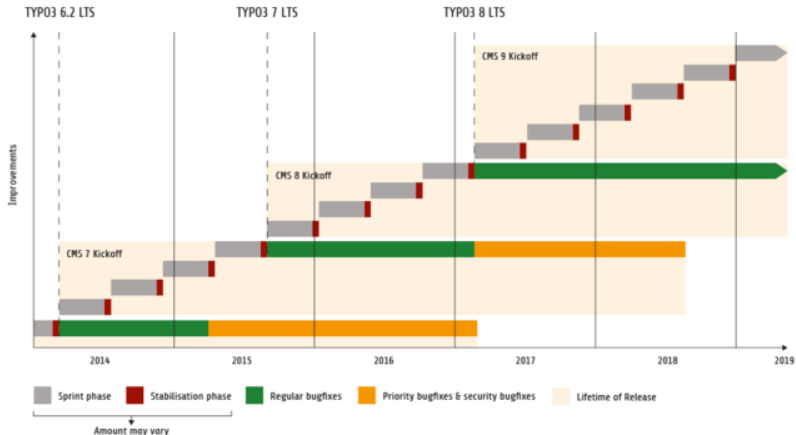
Requisitos del Sistema

- PHP*: v5.5.0 - v5.6.x
- MySQL: v5.5.x - v5.6.x (modo no estricto)
- Espacio de disco: mín 200 MB
- Ajustes de PHP:
 - `memory_limit` \geq 128M
 - `max_execution_time` \geq 240s
 - opción de compilación `-disable-ipv6` no debe ser usada
- Backend requiere IE \geq 9 o cualquier otro navegador moderno

*) Detalles adicionales: [Requisitos Mínimos de PHP para TYPO3 CMS 7](#)

Introducción

Línea de tiempo de Desarrollo y Lanzamiento



Introducción

Línea de lanzamiento de TYPO3 CMS

Fechas de lanzamiento estimadas y sus enfoques principales:

- v7.0 02/Dic/2014 Revisión de Backend Vol 1
- v7.1 24/Feb/2015 Optimización & Limpieza del núcleo
- v7.2 28/Apr/2015 Frontend
- v7.3 16/Jun/2015 Ecosistema de Paquetes, Composer y Manejo de Extensiones
- v7.4 04/Ago/2015 Revisión de Backend Vol 2
- **v7.5 29/Sep/2015 Finalización**
- v7 LTS Oct/Nov/2015 **TYPO3 CMS 7 LTS** (Lanzamiento a Largo Plazo)

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

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

Introducción

Instalación

- Procedimiento de instalación oficial bajo Linux/Mac OS X (DocumentRoot por ejemplo /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
```

- Enlaces simbólicos bajo Microsoft Windows:
 - Use junction en Windows XP/2000
 - Use mklink en Windows Vista y Windows 7

Introducción

Actualización a TYPO3 CMS 7.x

- Actualizaciones sólo posibles desde TYPO3 CMS 6.2 LTS
- TYPO3 CMS < 6.2 debe ser actualizado a TYPO3 CMS 6.2 LTS primero
- Instrucciones de actualización:
http://wiki.typo3.org/Upgrade#Upgrading_to_7.5
- Guía oficial de TYPO3 "Instalación y Actualización de TYPO3":
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Enfoque general:
 - Comprobar requisitos mínimos del sistema (PHP, MySQL, etc.)
 - Revisar **deprecation_*.log** en instancia antigua de TYPO3
 - Actualizar todas las extensiones a la última versión
 - Desplegar fuentes nuevas y ejecutar Herramienta de Instalación
→Asistente de Actualización
 - Revisar el módulo de inicio para usuarios backend (opcionalmente)

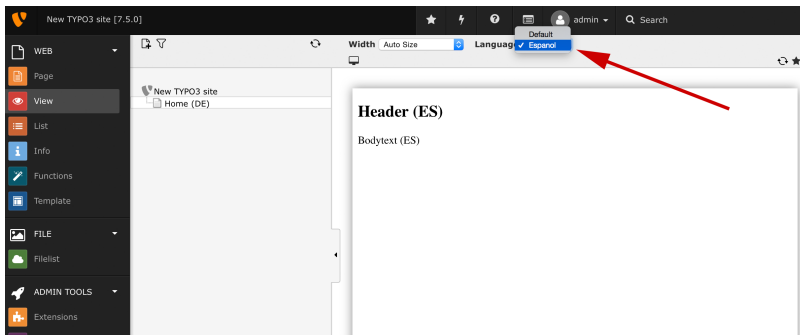
Capítulo 1: Interfaz de Usuario de Backend

Interfaz de Usuario de Backend

Selector de Lenguaje en Módulo "Vista de página"

WEB→View añade un desplegable para seleccionar un lenguaje para la previsualización de la página.

(puede desactivarse con `mod.SHARED.view.disableLanguageSelector = 1`)









Interfaz de Usuario de Backend

Elemento de Contenido `textmedia`

Un nuevo Elemento de Contenido **"Text & Media"** combina los elementos `text`, `image` y `textpic`.

New content element

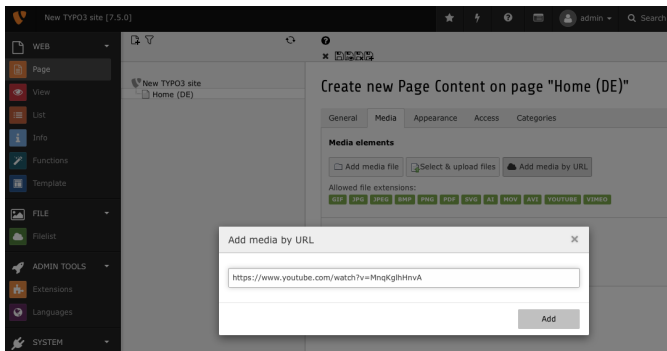
Please select the type of page content you wish to create:

Typical page content	Special elements	Form elements	Plugins
 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.			
 File Links			

Interfaz de Usuario de Backend

Ficheros YouTube y Vimeo

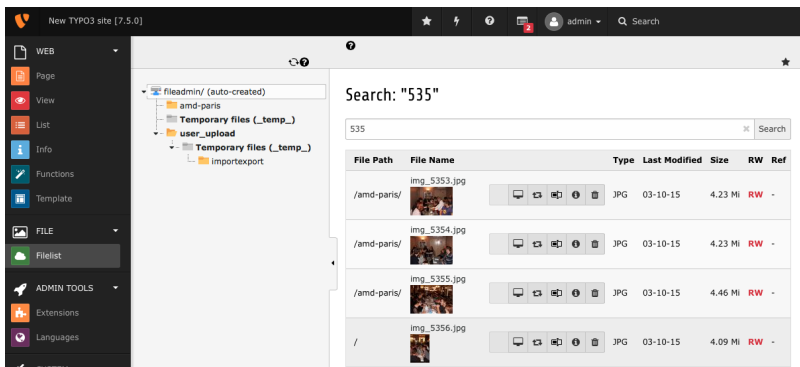
Elemento de Contenido **"Text & Media"** permite a los editores incluir ficheros externos YouTube y Vimeo, así como ficheros locales.



Interfaz de Usuario de Backend

Búsqueda en Módulo Filelist

Módulo "Filelist" añade una búsqueda por nombre de fichero ahora (recursivamente desde la carpeta actual).



The screenshot shows the TYPO3 Filelist module interface. The left sidebar contains navigation options: WEB (Page, View, List, Info, Functions, Template), FILE (Filelist), and ADMIN TOOLS (Extensions, Languages). The main content area displays a file tree for 'fileadmin/ (auto-created)' with subfolders 'amd-paris', 'Temporary files (_temp_)', and 'user_upload'. A search bar at the top right contains the text 'Search: "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	-

Capítulo 2: TSconfig & TypoScript

Tsconfig & TypeScript

Condiciones para Includes TypeScript

- INCLUDE_TYPOSCRIPT tiene una propiedad extra (opcional) "condition" ahora, que incluye el fichero/directorio sólo, si la condición se cumple

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

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


TScnfig & TypoScript

Opción TCA: Mostrar Offset de Fecha

- Opción TCA `disableAgeDisplay` deshabilita la visualización de la edad

(por ejemplo: "2015-08-30 (-27 días)")

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

- Como una precondición, tipo del campo tiene que ser `input` y `eval` tiene que configurarse a `date`

TSconfig & TypeScript

Ficheros de Etiquetas de Lenguaje Inline con TypeScript (1)

- Ficheros de lenguaje XLF pueden leerse y convertirse en un vector inline
- Esto habilita el acceso a etiquetas de lenguaje en JavaScript por ejemplo
- Se soportan los siguientes tres parámetros opcionales:
 - `selectionPrefix`:
sólo se incluirán claves de etiquetas que empiecen con este prefijo
 - `stripFromSelectionName`:
cadena que será eliminada de cualquier clave de etiqueta incluida
 - `errorMode`:
modo de error si no puede encontrarse el fichero:
0: entrada syslog (por defecto), 1: ignorar, 3: lanzar una excepción

TSconfig & TypeScript

Ficheros de Etiquetas de Lenguaje Inline con TypeScript (2)

■ Ejemplo:

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

■ Salida:

```
<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="22 932 234 955" data-label="Page-Footer"><p>TYPO3 CMS 7.5 - Qué hay Nuevo</p></div><div data-bbox="806 884 970 946" data-label="Page-Footer"><img alt="TYPO3 logo" data-bbox="806 884 970 946"/>The logo for TYPO3, featuring a stylized orange and white shape to the left of the text "TYPO3".</div>
```

TScnfig & TypoScript

Previsualización de Workspace por TScnfig

- TYPO3 CMS genera enlaces de previsualización sólo para tablas `tt_content`, `pages` y `pages_language_overlay` por defecto
- Esto puede configurarse usando PageTScnfig ahora:

```
# 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 (or table tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = field:pid
```

Tsconfig & TypoScript

Calidad de Imagen de `sourceCollection`

- Puede configurarse ahora la calidad de imagen de cada entrada `sourceCollection`
- Este ajuste toma precedencia sobre la configuración en la Herramienta de Instalación (almacenado en el fichero `LocalConfiguration.php`)
- Ejemplo:

```
# 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

Contar Elementos en una Lista

- Ha sido añadida una nueva propiedad `returnCount` a la propiedad `stdWrap split`
- Esto permite contar el número de elementos en una lista separada por comas
- El siguiente código devuelve 9 por ejemplo:

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

TSconfig & TypoScript

Manejo de Backend Layouts (1)

- Se ha simplificado el manejo de backend layouts para el frontend
- Puede usarse una nueva opción `pagelayout` en TypoScript
- Ejemplo:

```
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
    }
}
```

(continúa en la siguiente página)

Manejo de Backend Layouts (2)

- ...donde `key.data = pagelayout` reemplaza el siguiente código:

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


Tsconfig & TypoScript

Miscelánea

- Ha sido introducida la función `stdWrap bytes` en TYPO3 CMS 7.4
- La habilidad para configurar la base ha sido añadida en TYPO3 CMS 7.5, que permite definir si usar una base de 1000 o 1024 con la que calcular

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

```
bytes.base = 1000
```

Capítulo 3: Cambios en Profundidad

Cambios en Profundidad

Elementos de Contenido basados en Fluid (1)

- Ha sido implementada una nueva extensión del sistema **"Elementos de Contenido basados en Fluid"**
- Se usan templates Fluid para el renderizado de elementos de contenido en lugar de TypoScript
- Podría ser una alternativa a *CSS Styled Content* en un punto en el futuro
- Incluye los siguientes templates estáticos para usar esta característica:
 - Elementos de Contenido (`fluid_styled_content`)
 - Elementos de Contenido CSS (opcional) (`fluid_styled_content`)

Cambios en Profundidad

Elementos de Contenido basados en Fluid (2)

- Además, el siguiente template PageTSconfig tiene que añadirse a las propiedades de la página:

```
Elementos de Contenido basados en Fluid  
(fluid_styled_content)
```

- Sobreescribe templates por defecto añadiendo rutas propias en la configuración 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/
```

Cambios en Profundidad

Elementos de Contenido basados en Fluid (3)

- Migre de *CSS Styled Content* a *Elementos de Contenido basados en Fluid*:
 - Desinstale extensión `css_styled_content`
 - Instale extensión `fluid_styled_content`
 - Use el Asistente de Actualización en la Herramienta de Instalación para migrar Elementos de Contenido `text`, `image` y `textpic` a `textmedia`

Nota: "*Elementos de Contenido basados en Fluid*" está todavía en un estado temprano y cambios de última hora son posibles hasta TYPO3 CMS 7 LTS. También algunos conflictos referentes a *CSS Styled Content* posiblemente todavía existan.

Cambios en Profundidad

Método `SELECTmmQuery`

- Ha sido añadido a la clase `DatabaseConnection` un nuevo método `SELECT_mm_query`
- Extraído de `exec_SELECT_mm_query` para separar la construcción y ejecución de consultas M:M.
- Esto habilita el uso de la construcción de consultas en la capa de abstracción de la base de datos

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

Cambios en Profundidad

Optimizar Tablas de Base de Datos en MySQL

- Nueva tarea del programador para ejecutar el comando MySQL `OPTIMIZE TABLE` en las tablas seleccionadas de la base de datos
- Este comando reorganiza el almacenaje físico de los datos de la tabla y los datos de índices asociados para reducir espacio de almacenaje y mejorar la eficiencia de E/S
- Se soportan los siguientes tipos de tablas:
MyISAM, InnoDB y ARCHIVE
- Usar esta tarea con DBAL y otros DBMS no está soportado debido al hecho de que los comandos usados son específicos de MySQL

Nota: optimizar tablas es un proceso intensivo de E/S. También en MySQL < 5.6.17 el proceso bloquea las tablas mientras se ejecuta, lo que puede impactar en la página web.

Cambios en Profundidad

Manejo de Medios Online (1)

- Ahora se soportan medios externos (medios online)
- Como ejemplos, se ha implementado el soporte de vídeos Youtube y Vimeo en el núcleo
- Pueden añadirse recursos como URLs usando el elemento de contenido **"Text & Media"** por ejemplo
- La clase auxiliar apropiada recoge los meta datos y suministra una imagen que será usada como la previsualización si está disponible

Cambios en Profundidad

Manejo de Medios Online (2)

Las siguientes sintaxis de URL son posibles:

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>`

Cambios en Profundidad

Manejo de Medios Online (3)

- Puede lograrse el acceso a los recursos usando Fluid como sigue:

```
<!-- 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>
```

- Opciones de configuración personalizadas para vídeos YouTube:
autoplay, controls, loop, enablejsapi, showinfo, no-cookie
- Opciones de configuración personalizadas para vídeos Vimeo:
autoplay, loop, showinfo

Cambios en Profundidad

Manejo de Medios Online (4)

- Para registrar tu propio servicio de medios online, necesitas una clase `OnlineMediaHelper` que implemente `OnlineMediaHelperInterface` y una clase `FileRenderer` que implemente `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';
```

Cambios en Profundidad

Enrutamiento Backend

- Se ha añadido un nuevo componente de enrutamiento al backend de TYPO3 que maneja dirigirse a diferentes módulos/llamadas dentro de TYPO3 CMS
- Pueden definirse rutas en la siguiente clase:

Configuration/Backend/Routes.php

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

- El método llamado contiene objetos de petición y respuesta que cumplen con PSR-7:

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

Cambios en Profundidad

Definición de Autocargado en `ext_emconf.php`

- Extensiones pueden proporcionar ahora una o más definiciones PSR-4 en el fichero `ext_emconf.php`
- Esto era ya posible en `composer.json`, pero con esta nueva característica, los desarrolladores de extensiones no necesitan proporcionar más un fichero `composer` sólo para esto

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

(ésta es la nueva manera recomendada de registrar clases en TYPO3)

Cambios en Profundidad

Nueva Fábrica de Iconos (1)

- Lógica para trabajar con iconos, tamaños de iconos y superposiciones de iconos está ahora agrupada en la nueva clase `IconFactory`
- La nueva fábrica de iconos reemplazará a la antigua API de skins de iconos paso por paso
- Se registrarán todos los iconos del núcleo directamente en la clase `IconRegistry`
- Extensiones deben usar `IconRegistry::registerIcon()` para sobrescribir iconos existentes o añadir iconos adicionales a la fábrica de iconos:

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

Cambios en Profundidad

Nueva Fábrica de Iconos (2)

- El núcleo de TYPO3 CMS implementa tres clases proveedoras de iconos: `BitmapIconProvider`, `FontawesomeIconProvider` y `SvgIconProvider`

- Uso de ejemplo:

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

- Valores válidos para `Icon::SIZE_...` son: `SIZE_SMALL`, `SIZE_DEFAULT` y `SIZE_LARGE`
- Valores válidos para `Icon::STATE_...` son: `STATE_DEFAULT` y `STATE_DISABLED`

Cambios en Profundidad

Nueva Fábrica de Iconos (3)

- El núcleo de TYPO3 CMS proporciona un ViewHelper Fluid que facilita el usar iconos dentro de una vista 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>
```


Cambios en Profundidad

Hooks y Señales

- Ha sido añadida una nueva señal al LinkValidator, que permite procesamiento adicional sobre la inicialización de un registro específico (p.e. obteniendo datos de contenido desde la configuración del plugin en registro)
- Registrando la señal en el fichero `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'  
);
```

Cambios en Profundidad

JumpUrl como Extensión del Sistema (1)

- Se ha movido la generación y manejo de JumpURLs a una nueva extensión del sistema `jumpurl`
- Fueron introducidos nuevos hooks que permiten la generación y manejo personalizado de URL (ver siguiente página)

BREAKING CHANGE!

Cambios en Profundidad

JumpUrl como Extensión del Sistema (2)

■ Hook 1: manipulando **URLs** durante la generación del enlace

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

■ Hook 2: manejo de **enlaces**

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

Cambios en Profundidad

Interfaz de Línea de Comandos (CLI)

- Llamando a `typo3/cli_dispatch.phpsh` vía la línea de comandos ahora muestra un mensaje coloreado de error si una clave CLI inválida o ninguna clave es suministrada como primer parámetro
- Controladores de comando Extbase pueden ahora residir en subcarpetas arbitrarias dentro de la carpeta `Command`
- Ejemplo:

Controlador en fichero:

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

...puede ser llamado vía CLI:

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

Cambios en Profundidad

Miscelánea (1)

- Los botones de mover del tipo group TCA pueden ser ahora explícitamente deshabilitados usando la opción `hideMoveIcons = TRUE`
- Método `makeCategorizable` ha sido extendido con un nuevo parámetro `override` para ajustar una nueva configuración de categoría para una combinación ya registrada de table/field
- Ejemplo:

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

Último parámetro (aquí: TRUE) fuerza la sobrescritura (valor por defecto es FALSE).

Cambios en Profundidad

Miscelánea (2)

- Nueva función genera un ID único

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

- El tipo de fichero `.typoscript` ha sido añadido a la lista de tipos de ficheros de texto plano válidos

- Nueva opción de configuración define extensiones de ficheros media

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

BREAKING CHANGE!

Capítulo 4: Extbase & Fluid

Filtrado de Severidad para FlashMessageQueue

- En TYPO3 CMS < 7.5, todos los mensajes de la FlashMessageQueue pueden ser obtenidos y/o eliminados sólo
- En TYPO3 CMS >= 7.5, esto puede hacerse para una severidad específica:

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


Extbase & Fluid

Soporte Query añadido para "between"

- Se ha añadido soporte para `between` al objeto Extbase Query
- No hay beneficio de funcionamiento debido al hecho de que la DBMS convierte "between" internamente de todos modos: `min <= expr AND expr <= max`
- La nueva característica Extbase replica el comportamiento de la DBMS construyendo una condición lógica AND, de manera que ésta funcione a lo largo de todas las DBMS

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

Colas de FlashMessage Múltiples

- Ahora es posible implementar múltiples FlashMessageQueues:

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

- El acceso usando Fluid funciona como sigue:

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

Extbase & Fluid

ViewHelper Media (1)

- Para renderizar fácilmente vídeo, audio y todos los otros tipos de fichero con una clase registrada `Renderer` en el frontend, el `MediaViewHelper` ha sido implementado
- `MediaViewHelper` primero chequea si hay un `Renderer` presente para el fichero dado - si no, recurre y renderiza un tag `image`
- Ejemplos:

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

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

ViewHelper Media (2)

■ Ejemplos (continuación):

```
<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

Extensión del Sistema `form` (1)

- Extensión del sistema `form` (incluyendo el modelo de datos personalizado, lógica de controlador, validación de propiedades, vistas y templating) ha sido adaptado para soportar el stack MVC Extbase/Fluid
- Esto permite mejor personalización y control del comportamiento generado y marcado simplemente modificando los templates Fluid o utilizando una lógica propia de view helper personalizada
- Cada elemento del `form` usa su propio `Partial`, que puede ser también configurado a través de la opción `TypoScript partialPath = ...`

Extbase & Fluid

Extensión del Sistema `form` (2)

- Los siguientes tres nuevos ViewHelpers existen:
 - `AggregateSelectOptionsViewHelper` (para tags `optgroup`)
 - `SelectViewHelper` (para tags `optgroup`)
 - `PlainMailViewHelper` (para visualizar mails de texto plano)
- Además, hay tres Vistas:
 - `show` (el form en sí mismo)
 - `confirmation` (la página de confirmación)
 - `postProcessor/mail` (el email)
- Rutas de template y visibilidad de los campos puede ser personalizada para cada Vista individualmente

Extbase & Fluid

Anotación `@cli`

- Al usar la nueva anotación `@cli`, comandos en un Extbase `CommandController` pueden ser marcados como comandos CLI solamente
- Estos comandos son excluidos de la selección de comandos del programador
- Típicos casos de uso son comandos tales como `extbase:help:help` por ejemplo

Capítulo 5: Funciones Obsoletas/Eliminadas

Funciones Obsoletas/Eliminadas

Métodos relacionados con Slash en GeneralUtility

- Los siguientes métodos de la clase GeneralUtility han sido marcados como **obsoletos**:
 - `GeneralUtility::addSlashesOnArray()`
 - `GeneralUtility::stripSlashesOnArray()`
 - `GeneralUtility::slashArray()`

Funciones Obsoletas/Eliminadas

CLI-related Constants and Methods

- Lógica relativa a scripts basados en regular CLI con la opción CLIkey ha sido movida a la `CliRequestHandler` y entonces, el siguiente método marcado como **obsoleto**:

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

- Adicionalmente, las siguientes constantes y parámetros globales han sido marcados **obsoletos**:

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

Funciones Obsoletas/Eliminadas

IconUtility

- Interfaz `IconUtilityOverrideResourceIconHookInterface` así como la clase `IconUtility` será eliminada en TYPO3 CMS 8
- Todos los métodos públicos de esta clase han sido marcados como **obsoletos**:

<code>IconUtility::skinImg()</code>	<code>IconUtility::getIcon()</code>
<code>IconUtility::getSpriteIcon()</code>	<code>IconUtility::getSpriteIconForFile()</code>
<code>IconUtility::getSpriteIconForRecord()</code>	<code>IconUtility::getSpriteIconForResource()</code>
<code>IconUtility::getSpriteIconClasses()</code>	

(algunos métodos han sido movidos a `IconFactory`)

- Además, el siguiente ajuste `PageTSConfig` ha sido marcado como **obsoleto**:

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

Funciones Obsoletas/Eliminadas

Métodos `HtmlParser` Obsoletos

- Funcionalidad de sustitución de marcador ha sido movida desde `core/Classes/Html/HtmlParser.php` a su propia clase `core/Classes/Service/MarkerBasedTemplateService.php`
- Los siguientes métodos dentro de `HtmlParser` serán eliminados en TYPO3 CMS 8 y han sido marcados como **obsoletos** ahora:

```
HtmlParser::getSubpart()
```

```
HtmlParser::substituteSubpart()
```

```
HtmlParser::substituteSubpartArray()
```

```
HtmlParser::substituteMarker()
```

```
HtmlParser::substituteMarkerArray()
```

```
HtmlParser::substituteMarkerAndSubpartArrayRecursive()
```

Funciones Obsoletas/Eliminadas

Extensión del sistema `form`

- Extensión del sistema `form` está basada en Extbase/Fluid ahora
- De este modo todos los ajustes del layout basados en TypoScript han sido marcados como **obsoletos** y usar `layout` no está recomendado más:

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

Funciones Obsoletas/Eliminadas

ViewHelpers y Métodos Obsoletos

- Los siguientes ViewHelpers han sido marcados como **obsoletos**:

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

- Los siguientes métodos han sido marcados como **obsoletos**:

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

Capítulo 6: Fuentes y Autores

Fuentes y Autores

Fuentes

Noticias de TYPO3:

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

Informaciones de Lanzamiento:

- http://wiki.typo3.org/TYPO3\CMS_7.5.0
- [INSTALL.md](#) and [Changelog](#)
- [typo3/sysex/core/Documentation/Changelog/7.5/*](http://typo3.org/typo3/core/Documentation/Changelog/7.5/)

Sistema de seguimiento de errores de TYPO3:

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

Repositorios Git de TYPO3:

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

Diapositivas TYPO3 CMS Qué hay Nuevo:

Patrick Lobacher

(Investigación, Recogida de Información y Versión en Alemán)

Michael Schams

(Líder del Proyecto y Versión en Inglés)

Traducciones por:

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>

Licencia bajo Creative Commons BY-NC-SA 3.0

