

TYPO3 CMS 8.0 – Τι Νέο Υπάρχει

Σύνοψη των νέων λειτουργιών, αλλαγών και βελτιώσεων

Δημιουργήθηκε από:

Patrick Lobacher και Michael Schams

Μετάφραση στα Ελληνικά από:

Angeliki Plati

25/April/2016

Creative Commons BY-NC-SA 3.0



TYPO3 CMS 8.0 - Τι Νέο Υπάρχει

Σύνοψη κεφαλαίου

Εισαγωγή

Διεπαφή Χρήστη Backend

TSconfig & TypoScript

Ριζικές Αλλαγές

Extbase & Fluid

Κατηρηγμένες/Αφηρημένες Μέθοδοι

Πηγές και Συγγραφείς

Εισαγωγή Τα Γεγονότα

TYPO3 CMS 8.0 - Τα Γεγονότα

- Ημερομηνία Έκδοσης: 22 Μαρτίου 2016
- Τύπος Δημοσίευσης: **Sprint Release**
- Σλόγκαν: Ανάψτε τις μηχανές σας



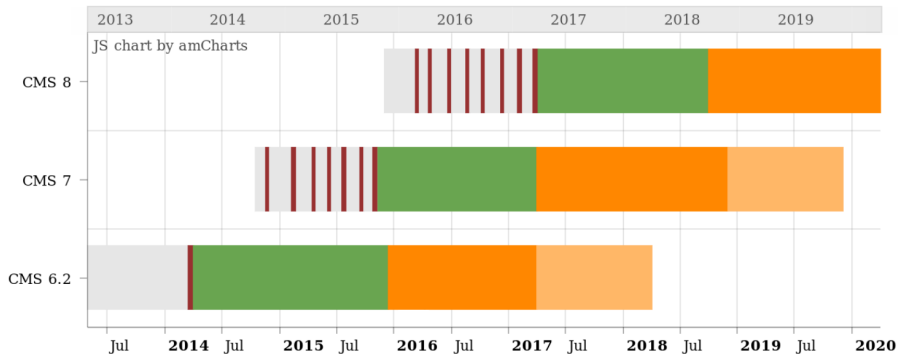
Εισαγωγή

Προδιαγραφές Συστήματος

- **PHP:** έκδοση 7
- **MySQL:** έκδοση 5.5 με 5.7
- **Χωρητ. Δίσκου:**
τουλάχιστον 200 MB
- **Ρυθμίσεις PHP:**
 - `memory_limit >= 128M`
 - `max_execution_time >= 240s`
 - `max_input_vars >= 1500`
 - η επιλογή `-disable-ipv6` δεν πρέπει να χρησιμοποιείται
- Για το **backend** απαιτείται η χρήση **Microsoft Internet Explorer 11** ή νεότερου, **Microsoft Edge**, **Google Chrome**, **Firefox**, **Safari** ή οποιουδήποτε άλλου μοντέρνου, συμβατού **browser**.

Εισαγωγή

Χρονοδιάγραμμα Ανάπτυξης και Δημοσίευσης



TYPO3 CMS - Πλάνο

Ημερομηνίες δημοσίευσης και ο πρωταρχικός τους στόχος:

- 8.0 22/Μαρ/2016 Προσθέτοντας τις τελευταίες λεπτομέρειες
- 8.1 03/Μαΐ/2016 Ενσωμάτωση Cloud
- v8.2 05/Ιουλ/2016 Επεξεργαστής Εμπλουτισμένου Κειμένου
- 8.3 30/Αυγ/2016 Frontend Επεξεργασία σε Steroids
- v8.4 18/Οκτ/2016 δεν έχει ακόμα οριστεί
- v8.5 20/Δεκ/2016 Υποστήριξη Integrator
- v8.6 14/Φεβ/2017 δεν έχει ακόμα οριστεί
- v8.7 04/Απρ/2017 Προετοιμασία του LTS

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

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

Installation

- Επίσημη διαδικασία εγκατάστασης σε Linux/Mac OS X (DocumentRoot για παράδειγμα /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
```

- Συμβολικοί σύνδεσμοι για Microsoft Windows:
 - Χρησιμοποιήστε junction για Windows XP/2000
 - Χρησιμοποιήστε mlink για Windows Vista and Windows 7

Αναβάθμιση σε TYPO3 CMS 8.x (1)

- Η αναβάθμιση είναι δυνατή μόνο από TYPO3 CMS 7.6 LTS
- Η έκδοση TYPO3 CMS < 7.6 LTS πρέπει να αναβαθμιστεί πρώτα σε TYPO3 CMS 7.6 LTS
- Οδηγίες αναβάθμισης:
http://wiki.typo3.org/Upgrade#Upgrading_to_8.0
- Επίσημος οδηγός TYPO3 TYPO3 Installation and Upgrading:
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Γενική προσέγγιση:
 - Ελέγξτε τις ελάχιστες απαιτήσεις συστήματος (PHP, MySQL, etc.)
 - Κάνετε ανασκόπηση του **deprecation_*.log** στην παλιά υπόσταση της TYPO3
 - Ανανεώστε όλες τις **extensions** στην τελευταία τους έκδοση

Αναβάθμιση σε **TYPO3 CMS 8.x (2)**

- Γενική προσέγγιση - Συνέχεια:
 - Κάνετε **deploy** τις νέες πηγές και τρέξτε το **Install Tool -> Upgrade Wizard**
 - Κάνετε ανασκόπηση του **startup module** για τους χρήστες **backend** (προαιρετικά)

PHP Έκδοση 7 (1)

- Η PHP 7.0 είναι η στοιχειώδης προϋπόθεση για την TYPO3 CMS 8.x
- Η TYPO3 θα υποστηρίζει ακόλουθες PHP 7 releases όταν δημοσιοποιούνται
- Η νέα έκδοση δίνει μια σημαντική ώθηση στην απόδοση του συνολικού συστήματος
- Οι χρήστες **backend** δε θα παρατηρήσουν μόνο ένα πιο ομαλό **interface**, αλλά και το «νέο ρεκόρ όλων των εποχών» για την κλήση πλήρους **cached** σελίδας, το οποίο βρίσκεται τώρα κάτω από τα 7 ms. Ο χρόνος αυτός είναι περίπου 40% λιγότερος συγκριτικά με εκείνου της έκδοσης PHP 5.5

PHP Έκδοση 7 (2)

- Ξεκινήσαμε επίσης να χρησιμοποιούμε νέα **features** από αυτή την έκδοση **PHP**, για παράδειγμα οι κρυπτογραφικά ασφαλείς γεννήτριες ψευδο-τυχαίων αριθμών είναι ήδη σε χρήση

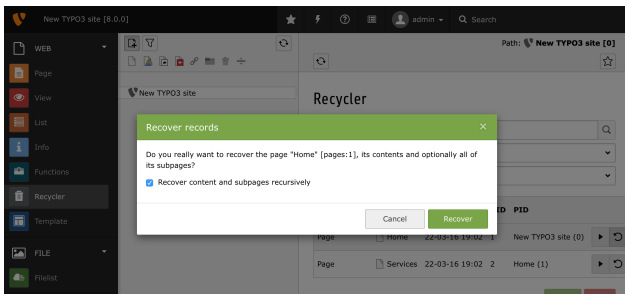
Κεφάλαιο 1:

Διεπαφή Χρήστη **Backend**

Διεπαφή Χρήστη **Backend**

Ανάκτηση σελίδων αναδρομικά προς την κορυφή της **rootline**

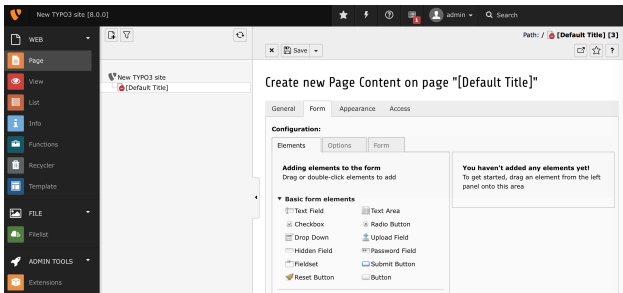
Ο **Recycler** υποστηρίζει τώρα την ανάκτηση διεγραμμένων σελίδων αναδρομικά προς την κορυφή της **rootline**. Αυτή η λειτουργία είναι διαθέσιμη μόνο για χρήστες **admin** εξαιτίας εσωτερικών περιορισμών άδειας (**permission**).



Διεπαφή Χρήστη Backend

Απευθείας φόρτωση του **form wizard** ως **inline wizard**

Ο **wizard** της **EXT:form** φορτώνεται απευθείας ως ενσωματωμένος (**inline wizard**). Δε χρειάζεται πλέον να γίνει **save** και **reload** του πρόσφατα δημιουργημένου στοιχείου περιεχομένου για να ανοίξει ο **wizard**. Αυτή είναι μία τεράστια βελτίωση της χρηστικότητας.



Διεπαφή Χρήστη **Backend**

Ορισμός ενός εναλλακτικού λογοτύπου **backend** μέσω του **Extension Manager**

Το λογότυπο του **backend** στην επάνω αριστερή γωνία μπορεί τώρα να διαμορφωθεί μέσα στην **configuration** της επέκτασης **EXT:backend** στον **Extension Manager**.

Οι επιλογές τις διαμόρφωσης είναι:

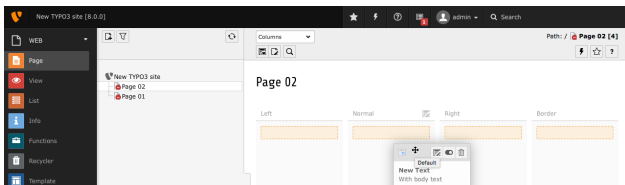
- πόρος ως σχετικό μονοπάτι της εγκατάστασης **TYPO3**
π.χ. `fileadmin/images/my-background.jpg`
- μονοπάτι προς μία επέκταση
π.χ. `EXT:my_theme/Resources/Public/Images/my-background.jpg`
- ένας εξωτερικός πόρος
π.χ. `//example.com/my-background.png`



Διεπαφή Χρήστη **Backend**

Αντιγραφή σείδων με τη λειτουργία **drag & drop**

Επιπρόσθετα στη συνήθη λειτουργία **drag & drop** στο δομοστοιχείο της σελίδας (που μετακινούσε στοιχεία περιεχομένου), είναι τώρα δυνατή η δημιουργία αντιγράφων: πατήστε το πλήκτρο **CTRL** καθώς κάνετε **drop** για να δημιουργήσετε ένα αντίγραφο του στοιχείου που «σέρνετε». Αφού το **drop** έχει ολοκληρωθεί, το δομοστοιχείο της σελίδας θα επαναφορτωθεί για να είναι σίγουρο πως το νέο στοιχείο θα δημιουργηθεί μαζί με όλη την απαραίτητη πληροφορία.



Κεφάλαιο 2: TScnfig & TypoScript

TSconfig & TypoScript

Ταξινόμηση των καρτελών στον **wizard** νέων στοιχείων περιεχομένου

- Είναι δυνατή η διαμόρφωση της σειράς των καρτελών στον **wizard** νέων στοιχείων περιεχομένου θέτοντας τις τιμές **before** και **after** στο **Page TSconfig**:

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

TSconfig & TypoScript

HTMLparser.stripEmptyTags.keepTags

- Έχει προστεθεί νέα επιλογή για τη διαμόρφωση HTMLparser.stripEmptyTags, η οποία επιτρέπει τη διατήρηση διαμορφωμένων tags
- Πριν από αυτή την αλλαγή, μόνο μία λίστα από tags μπορούσε να παρασχεθεί, η οποία έπρεπε να αφαιρεθεί
- Το ακόλουθο παράδειγμα αφαιρεί όλα τα κενά tags εκτός των tr και td:

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

Σημαντικό: εάν χρησιμοποιείται αυτή η ρύθμιση, η διαμόρφωση stripEmptyTags.tags δεν έχει πλέον καμία επίδραση. Μπορείτε να χρησιμοποιήσετε μία επιλογή κάθε φορά.

TScnfig & TypoScript

EXT:form - ενσωμάτωση των προκαθορισμένων φορμών (1)

- Το στοιχείο περιεχομένου της **EXT:form** επιτρέπει τώρα την ενσωμάτωση προκαθορισμένων φορμών.
- Ένας **integrator** μπορεί να καθορίσει φόρμες (π.χ. μέσα σε ένα πακέτο σελίδας) χρησιμοποιώντας το `plugin.tx_form.predefinedForms`
- Ένας **editor** μπορεί τώρα να προσθέσει ένα νέο στοιχείο περιεχομένου `mailform` σε μία σελίδα και να διαλέξει μία φόρμα από μία λίστα με προκαθορισμένα στοιχεία

TSconfig & TypeScript

EXT:form - ενσωμάτωση των προκαθορισμένων φορμών (2)

- Οι **integrators** μπορούν τώρα να δημιουργήσουν τις δικές τους φόρμες με **TypoScript**, η οποία παρέχει πολύ περισσότερες επιλογές από το να το κάνουν μέσω του **form wizard** (π.χ. οι **integrators** μπορούν να χρησιμοποιήσουν την λειτουργία **stdWrap**, η οποία δεν είναι διαθέσιμη με τη χρήση του **form wizard** (για λόγους ασφαλείας)

Tsconfig & TypoScript

EXT:form - ενσωμάτωση των προκαθορισμένων φορμών (3)

- Οι **editors** δε χρειάζεται να χρησιμοποιούν πλέον τον **form wizard**. Οι **editors** μπορούν να διαλέξουν τις προκαθορισμένες φόρμες οι οποίες διαθέτουν βελτιστοποιημένα **layouts**.
- Οι φόρμες μπορούν να επαναχρησιμοποιηθούν οπουδήποτε
- Οι φόρμες μπορούν να αποθηκευτούν έξω από τη βάση δεδομένων και να γίνουν **versioned**
- Για να είναι δυνατή η επιλογή μίας προκαθορισμένης φόρμας στο **backend**, η φόρμα αυτή πρέπει να έχει καταχωρηθεί χρησιμοποιώντας το **PageTS**:

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

TSconfig & TypeScript

EXT:form - ενσωμάτωση των προκαθορισμένων φορμών (4)

■ Παράδειγμα φόρμας:

```
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.de = Kontakt Formular
            }
        }
    }
}
10 = TEXTLINE
10 {
    name = name
    ...
```


Κεφάλαιο 3: Ριζικές Αλλαγές

Υποστήριξη **PECL-memcached** στο **MemcachedBackend**

- Η υποστήριξη για το δομοστοιχείο **PECL memcached** έχει προστεθεί στο **MemcachedBackend** του **Caching Framework**
- Αν και τα δύο, **memcache** και **memcached** είναι εγκατεστημένα, χρησιμοποιείται το **memcache** για να αποφευχθεί μία τεράστια αλλαγή.
- Ένας **integrator** μπορεί να θέσει την επιλογή **peclModule** για να χρησιμοποιήσει το προκαθορισμένο δομοστοιχείο **PECL**:

```
$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'  
        ]  
    ]  
]
```

Εγγενής υποστήριξη για την **Symfony Console (1)**

- Το **TYPO3** υποστηρίζει τώρα **out-of-the-box** το στοιχείο **Symfony Console** με την παροχή ενός νέου **command line script** που βρίσκεται στο `typo3/sysext/core/bin/typo3`. Στα **TYPO3 instances** που έχουν εγκατασταθεί μέσω του **Composer**, το **binary** συνδέεται με το **bin-directory**, π.χ. `bin/typo3`.
- Το νέο **binary** υποστηρίζει ακόμα τις υπάρχουσες παραμέτρους **command line** σε περίπτωση που δε βρεθεί καμία κατάλληλη εντολή **Symfony Console** ως εναλλακτική.

Εγγενής υποστήριξη για την **Symfony Console (2)**

- Η καταχώριση μίας εντολής έτσι ώστε να είναι διαθέσιμη μέσω του εργαλείου **typos3 command line** λειτουργεί με την τοποθέτηση ενός αρχείου **Configuration/Commands.php** μέσα σε οποιαδήποτε εγκατεστημένη επέκταση. Αυτό καταγράφει τις κλάσεις της **Symfony/Console/Command** που θα εκτελεστούν από την **typos3** μέσα σε ένα **associative array**. Το κλειδί είναι το όνομα της εντολής που θα κληθεί ως η πρώτη παράμετρος στην **typos3**.

Εγγενής υποστήριξη για την **Symfony Console (3)**

- Μια απαιτούμενη παράμετρος κατά την εγγραφή μιας εντολής είναι η ιδιότητα `class`. Προαιρετικά, η παράμετρος `user` μπορεί να τεθεί έτσι ώστε ένας χρήστης του **backend** να είναι συνδεδεμένος όταν καλείται η εντολή.
- Ένα αρχείο `Configuration/Commands.php` μπορεί να μοιάζει κάπως έτσι:

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

Εγγενής υποστήριξη για την **Symfony Console (4)**

- Μία κλήση θα μπορούσε να μοιάζει κάπως έτσι:

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

- Για **non-Composer** εγκατάσταση:

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

Κρυπτογραφικά ασφαλής γεννήτρια ψευδο-τυχαίων αριθμών

- Μία νέα κρυπτογραφικά ασφαλής γεννήτρια ψευδο-τυχαίων αριθμών (**CSPRNG**) έχει υλοποιηθεί στον **TYPO3 core**.
Αξιοποιεί τις νέες λειτουργίες της **CSPRNG** στην **PHP 7**.
- Το **API** βρίσκεται στη κλάση `\TYPO3\CMS\Core\Crypto\Random`
- Παράδειγμα:

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

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

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

Το Στοιχείο **Wizard** (1)

- Ένας νέος **wizard** έχει προστεθεί. Το στοιχείο αυτό μπορεί να χρησιμοποιηθεί για καθοδηγούμενες αλληλεπιδράσεις χρήστη
- Το δομοστοιχείο **RequireJS** μπορεί να χρησιμοποιηθεί με την συμπερίληψη του `TYPO3\CMS\Backend\Wizard`
- Ο **wizard** υποστηρίζει μόνο απλές ενέργειες (οι διασταυρώσεις δεν είναι ακόμη δυνατές)
- Το **API** βρίσκεται στην κλάση `\TYPO3\CMS\Core\Crypto\Random`
- Το στοιχείο **wizard** έχει τις ακόλουθες **public** μεθόδους:

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


Το Στοιχείο **Wizard** (2)

- Το **event wizard-visible** πυροδοτείται όταν ολοκληρωθεί το **rendering** του **wizard**
- Οι **wizards** μπορούν να τερματιστούν με τη πυροδότηση του **event wizard-dismiss**
- Οι **wizards** πυροδοτούν το **event wizard-dismissed** αν ο **wizard** τερματιστεί
- Μπορείτε να ενσωματώσετε το δικό σας **listener** με τη χρήση του **Wizard.getComponent()**

Ριζικές Αλλαγές

Τα παραγόμενα **asset** αρχεία έχουν μετακινηθεί

- Η δομή των αρχείων μέσα στο `typo3temp` έχει αλλάξει για να διαχωρίσει τα **assets** που απαιτούν δημόσια πρόσβαση από προσωρινά δημιουργημένα αρχεία (π.χ. για σκοπούς **caching** ή **locking** και απαιτούμενης **server-side** πρόσβασης μόνο).
- Αυτά τα **assets** έχουν μετακινηθεί από τους φακέλους: `_processed_`, `compressor`, `GB`, `temp`, `Language`, `pics` και έχουν αναδιοργανωθεί μέσα στους:
 - `typo3temp/assets/js/`
 - `typo3temp/assets/css/`,
 - `typo3temp/assets/compressed/`
 - `typo3temp/assets/images/`

Αλλαγές στο **ImageMagick/GraphicsMagick** (1)

- Οι ρυθμίσεις του επεξεργαστή γραφικών για το **Image-GraphicsMagick** έχουν μετονομαστεί (αρχείο: `LocalConfiguration.php`).

ΠΑΛΑΙΟ:

```
im_
```

NEO:

```
processor_
```

- Η «αρνητική» ονομασία όπως `noScaleUp` έχει αλλάξει στην αντίστοιχη «θετική». Κατά τη μετατροπή αυτή, οι προηγούμενες τιμές των ρυθμίσεων αναιρούνται για να αντικατοπτρίσουν τις αλλαγές στη σημασιολογία αυτών των επιλογών.

Αλλαγές στο **ImageMagick/GraphicsMagick** (2)

- Επιπρόσθετα, οι αναφορές σε συγκεκριμένες εκδόσεις του **ImageMagick/GraphicsMagick** έχουν αφαιρεθεί από ονόματα ρυθμίσεων και από τιμές.
- Η αχρησιμοποίητη επιλογή ρυθμίσεων `image_processing` έχει αφαιρεθεί χωρίς αντικατάσταση
- Η επιλογή ρυθμίσεων επεξεργαστή `colorspace` βρίσκεται στο `namespace processor_`

Ριζικές Αλλαγές

Άγκιστρα και Σήματα (1)

- Ένα επιπλέον άγκιστρο έχει προστεθεί στη μέθοδο `BackendUtility::viewOnClick()` για την μετεπεξεργασία της προεπισκόπησης **URL**
- Εγγραφή μίας κλάσης άγκιστρου η οποία υλοποιεί τη μέθοδο `postProcess`:

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

Άγκιστρα και Σήματα (2)

- Προγενέστερα του **TYPO3 CMS 7.6**, ήταν δυνατό να γίνει **override** ενός **overlay** εγγραφής μέσα σε **Web -> List**. Ένα νέο άγκιστρο στο **TYPO3 CMS 8.0** παρέχει την παλιά λειτουργικότητα.
- Το άγκιστρο καλείται με την ακόλουθη υπογραφή:

```
/**
 * @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) { ... }
```

- Εγγραφή μίας κλάσης άγκιστρου η οποία υλοποιεί τη μέθοδο **postOverlayPriorityLookup**:

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

Άγκιστρα και Σήματα (3)

- Ένα νέο σήμα έχει υλοποιηθεί προτού μία αποθήκη πόρων αρχικοποιηθεί.
- Εγγραφή της κλάσης η οποία υλοποιεί τη δική σας λογική στο `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'  
);
```

- Η μέθοδος καλείται με τις ακόλουθες παραμέτρους:
 - `int $uid` το `uid` μιας εγγραφής
 - `array $recordData` όλα τα δεδομένα ως `array`
 - `string $fileIdentifier` το αναγνωριστικό αρχείου

Αλγόριθμος κατακερματισμού συνθηματικού: **PBKDF2**

- Ένας νέος αλγόριθμος κατακερματισμού συνθηματικού **PBKDF2** έχει προστεθεί στην επέκταση συστήματος **saltedpasswords**
- Το **PBKDF2** σημαίνει: **Password-Based Key Derivation Function 2**
- Ο αλγόριθμος έχει σχεδιαστεί έτσι ώστε να είναι υπολογιστικά δαπανηρός για να μπορεί να αντισταθεί στο «βίαιο σπάσιμο» των συνθηματικών

Κεφάλαιο 4: Extbase & Fluid

Extbase & Fluid

Αυτόνομο αναθεωρημένο **Fluid**

- Η μηχανή απόδοσης (**rendering**) του **Fluid** στο **TYPO3 CMS** έχει αντικατασταθεί από το αυτόνομο **Fluid** το οποίο τώρα συμπεριλαμβάνεται ως μία εξάρτηση του **composer**
- Η παλιά επέκταση **Fluid** έχει μεατραπεί, θα λέγαμε, σε έναν **Fluid adapter** ο οποίος επιτρέπει στο **TYPO3 CMS** να χρησιμοποιεί το αυτόνομο **Fluid**
- Νέα χαρακτηριστικά/δυνατότητες έχουν προστεθεί σε περίπου όλες τις ενότητες του **Fluid**
- Το πιο σημαντικό: πολλά από τα στοιχεία του **Fluid** τα οποία είναι εντελώς εσωτερικά και που ήταν αδύνατο να αντικατασταθούν στο παρελθόν, είναι τώρα εύκολο να αντικατασταθούν και έχουν εξοπλιστεί με ένα δημόσιο **API**

RenderingContext (1)

- Το πιο σημαντικό κομμάτι του δημόσιου **API** είναι το **RenderingContext**
- Το προηγούμενο -εσωτερικό μόνο- **RenderingContext** που χρησιμοποιείτο από το **Fluid** έχει αναπτυχθεί έτσι ώστε να είναι υπεύθυνο για ένα νέο ζωτικό χαρακτηριστικό του **Fluid**: την υλοποίηση του **provisioning**
- Αυτό επιτρέπει στους προγραμματιστές να αλλάξουν μια σειρά από κλάσεις, που χρησιμοποιεί το **Fluid** για **parsing, resolving, caching** κτλ.
- Αυτό μπορεί να επιτευχθεί με την συμπερίληψη ενός προσαρμοσμένου **RenderingContext** ή με το χειρισμό του προεπιλεγμένου **RenderingContext** από δημόσιες μεθόδους.

RenderingContext (2)

- Τα χαρακτηριστικά στις ακόλουθες διαφάνειες μπορούν να ελεγχθούν με το χειρισμό του **RenderingContext**. Ως εκ προεπιλογής, κάνανα από αυτά δεν είναι ενεργοποιημένο - αλλά η κλήση μίας απλής μεθόδου (μέσω του στιγμιοτύπου **View**) σας επιτρέπει να τα ενεργοποιήσετε:

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

ExpressionNodes (1)

- Ο τύπος **ExpressionNodes** είναι μια νέα δομή σύνταξης του **Fluid**, ο οποίος μπορεί να χρησιμοποιηθεί αποκλειστικά εντός αγκίστρων

```
$view->getRenderingContext()->setExpressionNodeTypes(array(
    'Class\Number\One',
    'Class\Number\Two'
));
```
- Οι προγραμματιστές μπορούν να προσθέσουν τους δικούς τους επιπρόσθετους τύπους **ExpressionNode**
- Ο καθένας αποτελείται από ένα μοτίβο «ταιριαστών» μεθόδων που υπαγορεύεται από ένα **interface** που επεξεργάζεται αυτά τα «ταιριάσματα»
- Κάθε υπάρχων τύπος **ExpressionNode** μπορεί να χρησιμοποιηθεί ως αναφορά

ExpressionNodes (2)

Οι τύποι **ExpressionNodes** επιτρέπουν νέες συντάξεις, όπως:

- Ο τύπος **CastingExpressionNode**

επιτρέπει το **casting** μίας μεταβλητής σε συγκεκριμένους τύπους, για τη διασφάλιση ότι μία μεταβλητή είναι **integer** ή **boolean**. Χρησιμοποιείται απλά με τη λέξη κλειδί `as`: `{myStringVariable as boolean}` ή `{myBooleanVariable as integer}` κ.ο.κ. Η απόπειρα του να γίνει `cast` μίας μεταβλητής σε έναν μη συμβατό τύπο προκαλεί ένα τυπικό σφάλμα **Fluid**.

- Ο τύπος **MathExpressionNode**

επιτρέπει τις βασικές μαθηματικές πράξεις πάνω σε μεταβλητές, π.χ. `{myNumber + 1}`, `{myPercent / 100}` ή `{myNumber * 100}` κ.ο.κ. Μία αδύνατη μαθηματική παράσταση επιστρέφει ένα άδειο αποτέλεσμα.

ExpressionNodes (3)

Οι τύποι **ExpressionNodes** επιτρέπουν νέες συντάξεις, όπως:

- Ο τύπος **TernaryExpressionNode**

επιτρέπει μία **inline ternary** συνθήκη η οποία λειτουργεί μόνο πάνω σε μεταβλητές. Τυπική περίπτωση χρήσης είναι: «αν αυτή η μεταβλητή ... τότε χρησιμοποίησε αυτή τη μεταβλητή αλλιώς χρησιμοποίησε μία άλλη μεταβλητή». Χρησιμοποιείται ως:

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

Σημείωση: δεν υποστηρίζει εμφωλευμένες εκφράσεις, συντάξεις **inline ViewHelper** ή ο,τιδήποτε παρόμοιο. Πρέπει να χρησιμοποιείται μόνο με τυπικές μεταβλητές ως είσοδο.

Extbase & Fluid

Τα **namespaces** είναι επεκτάσιμα (1)

- Το **Fluid** επιτρέπει σε κάθε ψευδώνυμο **namespace** (π.χ. `f:`) να επεκταθεί με την προσθήκη ενός επιπρόσθετου **PHP namespaces** σε αυτό
- Τα **PHP namespaces** ελέγχονται επίσης για την παρουσία κλάσεων **ViewHelper**
- Αυτό σημαίνει επίσης πως οι προγραμματιστές μπορούν να κάνουν **override** μεμονωμένους **ViewHelpers** με προσαρμοσμένες εκδόσεις και να τους καλούν όποτε το **namespace f:** χρησιμοποιείται.

Extbase & Fluid

Τα **namespaces** είναι επεκτάσιμα (2)

- Αυτή η αλλαγή επίσης σημαίνει πως τα **namespaces** δεν είναι πλέον μοναδικά.

Όταν χρησιμοποιείτε `{namespace f=My\Extension\ViewHelpers\}`

δε θα λαμβάνετε πλέον το σφάλμα **namespace already registered**.

Αντάυτου, το **Fluid** θα προσθέσει αυτό το **PHP namespace** και θα κοιτάξει για **ViewHelpers** και εκεί.

- Τα επιπρόσθετα **namespaces** ελέγχονται από πάνω προς τα κάτω, επιτρέποντας στα επιπλέον **namespaces** να κάνουν **override** κλάσεις **ViewHelper** με την τοποθέτησή τους στο ίδιο **scope**

Extbase & Fluid

Τα **namespaces** είναι επεκτάσιμα (3)

- Για παράδειγμα: `f:format.nl2br` μπορεί να γίνει **override** από `My\Extension\ViewHelpers\Format\Nl2brViewHelper`, με δεδομένη την εγγραφή **namespace** στις προηγούμενες διαφάνειες

Extbase & Fluid

Το **rendering** χρησιμοποιεί `f:render` (1)

Επιτρέπει προεπιλεγμένο περιεχόμενο στο προαιρετικό `f:render`:

- Όταν το `f:render` χρησιμοποιείται και το **flag optional = TRUE** έχει τεθεί, η απόδοση (**rendering**) ενός απόντος τμήματος έχει ως αποτέλεσμα ένα άδειο **output**.
- Αντί να αποδίδεται αυτό το άδειο **output**, μία νέα ιδιότητα **default (mixed)** έχει προστεθεί και μπορεί να συμπληρωθεί από μία προεπιλεγμένη τιμή τύπου **fallback**.
- Εναλλακτικά, το περιεχόμενο του **tag** μπορεί να χρησιμοποιηθεί για να καθορίσει αυτή την **default** τιμή, όπως πολλοί άλλοι ευέλικτοι **ViewHelpers** ως προς το περιεχόμενο και τις ιδιότητες.

Extbase & Fluid

Το **rendering** χρησιμοποιεί `f:render` (2)

Περνώντας το περιεχόμενο ενός **tag** με το `f:render` σε ένα **partial/section**:

- Παρέχει μία νέα προσέγγιση για τη διάρθρωση του **Fluid template rendering**
- Τα **partials** και τα **sections** μπορούν να χρησιμοποιηθούν ως **wrappers** για ένα αυθαίρετο κομμάτι κώδικα **template**.
- Παράδειγμα:

```
<f:section name="MyWrap">
  <div>
    <!-- more HTML, using variables if desired -->
    <!-- tag content of f:render output: -->
    {contentVariable -> f:format.raw()}
  </div>
</f:section>
<f:render section="MyWrap" contentAs="contentVariable">
  This content will be wrapped. Any Fluid code can go here.
</f:render>
```

Περίπλοκες εκφράσεις συνθηκών (1)

- Το **Fluid** υποστηρίζει τώρα κάθε βαθμό πολυπλοκότητας εκφράσεων συνθηκών με εμφωλευση και ομαδοποίηση:

```
<f:if condition="{variableOne} && {variableTwo} || {variableThree} || {variableFour}">
  // Done if both variable one and two evaluate to true, or if either variable three or four do.
</f:if>
```

- Επιπρόσθετα, το **f:else** έχει εξοπλιστεί με μία «συμπεριφορά» παρόμοια του **elseif**:

```
<f:if condition="{variableOne}">
  <f:then>Do this</f:then>
  <f:else if="{variableTwo}">
    Do this instead if variable two evals true
  </f:else>
  <f:else if="{variableThree}">
    Or do this if variable three evals true
  </f:else>
  . . .
```

Περίπλοκες εκφράσεις συνθηκών (2)

- Επιπρόσθετα, το `f:else` έχει εξοπλιστεί με μία «συμπεριφορά» παρόμοια του `elseif` - Sun'eqeia:

```
. . .  
  <f:else>  
    Or do this if nothing above is true  
  </f:else>  
</f:if>
```

Δυναμικά μέρη ονόματος μεταβλητής (1)

- Ένα άλλο αναγκαίο χαρακτηριστικό, επίσης συμβατό «προς τα πίσω», είναι η προστετημένη δυνατότητα της χρήσης αναφορών υπο-μεταβλητής κατά την προσπέλαση των μεταβλητών σας. Σκεφτείτε τον ακόλουθο πίνακα μεταβλητών **Fluid template**:

```
$mykey = 'foo'; // or 'bar', set by any source
$view->assign('data', ['foo' => 1, 'bar' => 2]);
$view->assign('key', $mykey);
```

- Με το ακόλουθο **Fluid template**:

```
You chose: {data.{key}}.
(output: "1" if key is "foo" or "2" if key is "bar")
```

Δυναμικά μέρη ονόματος μεταβλητής (2)

- Η ίδια προσέγγιση μπορεί επίσης να χρησιμοποιηθεί για την παραγωγή δυναμικών μερών μίας γραμματοσειράς ονόματος μεταβλητής:

```
$mydynamicpart = 'First'; // or 'Second', set by any source
$view->assign('myFirstVariable', 1);
$view->assign('mySecondVariable', 2);
$view->assign('which', $mydynamicpart);
```

- Με το ακόλουθο Fluid template:

```
You chose: {my{which}Variable}.
(output: "1" if which is "First" or "2" if which is "Second")
```


Νέοι ViewHelpers

- Μερικοί νέοι **ViewHelpers** έχουν επίσης προστεθεί ως μέρος του αυτοδύναμου **Fluid** και έτσι είναι διαθέσιμοι στο **TYPO3** από εδώ και στο εξής:

- **f:or**

Αυτός είναι ένας συντομότερος τρόπος γραφής (αλληλένδετων) συνθηκών. Υποστηρίζει την ακόλουθη σύνταξη, η οποία ελέγχει κάθε μεταβλητή και δίνει ως έξοδο την πρώτη μη κενή:

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

- **f:spaceless**

Αυτό μπορεί να χρησιμοποιηθεί σε μορφή **tag** γύρω από κώδικα **template** για να αποφεύγονται τα περιττά **whitespaces** και οι κενές γραμμές που προκαλούνται για παράδειγμα όταν δημιουργούμε εσοχές στις χρήσεις **ViewHelper**

Extbase & Fluid

Τα **ViewHelper namespaces** μπορούν να επεκταθούν και από την **PHP (1)**

- Με την προσπέλαση του **ViewHelperResolver** του **RenderingContext**, οι προγραμματιστές μπορούν να αλλάξουν τις συμπεριλήψεις των **ViewHelper namespaces** σε καθολικό επίπεδο (βλ: στιγμιότυπο **View**):

```
$resolver = $view->getRenderingContext()->getViewHelperResolver();  
// equivalent of registering namespace in template(s):  
$resolver->registerNamespace('news', 'GeorgRinger\News\ViewHelpers');  
// adding additional PHP namespaces to check when resolving ViewHelpers:  
$resolver->extendNamespace('f', 'My\Extension\ViewHelpers');  
// setting all namespaces in advance, globally, before template parsing:  
$resolver->setNamespaces(array(  
    'f' => array(  
        'TYPO3Fluid\Fluid\ViewHelpers', 'TYPO3\CMS\Fluid\ViewHelpers',  
        'My\Extension\ViewHelpers'  
    ),  
    . . .  
);
```

Extbase & Fluid

Τα **ViewHelper namespaces** μπορούν να επεκταθούν και από την **PHP (2)**

- Με την προσπέλαση του **ViewHelperResolver** του **RenderingContext**, οι προγραμματιστές μπορούν να αλλάξουν τις συμπεριλήψεις των **ViewHelper namespaces** σε καθολικό επίπεδο (βλ: στιγμιότυπο **View**)
- Συνέχεια:

```
. . .  
'vhs' => array(  
    'FluidTYPO3\Vhs\ViewHelpers', 'My\Extension\ViewHelpers'  
),  
'news' => array(  
    'GeorgRinger\News\ViewHelpers',  
);  
));
```

Extbase & Fluid

Οι **ViewHelpers** μπορούν να δεχτούν αυθαίρετες παραμέτρους (1)

- Αυτό το χαρακτηριστικό επιτρέπει στη δική σας κλάση **ViewHelper** να λάβει οποιονδήποτε αριθμό από επιπλέον παραμέτρους με τη χρήση οποιωνδήποτε ονομάτων επιθυμείτε
- Λειτουργεί με τον διαχωρισμό των παραμέτρων που «περνιούνται» σε κάθε **ViewHelper** σε 2 ομάδες: σε αυτές που είναι δηλωμένες χρησιμοποιώντας το `registerArgument` και στις υπόλοιπες
- Αυτές που δεν είναι δηλωμένες, «περνιούνται» σε μία ειδική συνάρτηση `handleAdditionalArguments` στην κλάση **ViewHelper**, η οποία «πετάει» σφάλμα στην προεπιλεγμένη υλοποίηση αν υπάρχουν επιπρόσθετες παράμετροι

Extbase & Fluid

Οι **ViewHelpers** μπορούν να δεχτούν αυθαίρετες παραμέτρους (2)

- Με το **overriding** αυτής της μεθόδου στον δικό σας **ViewHelper**, μπορείτε να αλλάξετε το αν και τότε ο **ViewHelper** πρέπει να «πετάξει» σφάλμα αν λάβει αδήλωτες παραμέτρους
- Αυτό το χαρακτηριστικό επιτρέπει επίσης στους **TagBasedViewHelpers** να δέχονται ελεύθερα αυθαίρετες παραμέτρους με πρόθεμα `data-` χωρίς να αποτυγχάνουν
- στους **TagBasedViewHelpers**, η μέθοδος **handleAdditionalArguments** απ'α *prosj'etei n'ees param'etrous sto tag* που παράγεται και «πετάει» σφάλμα αν δοθεί κάποια επιπρόσθετη παράμετρος που δεν είναι ούτε δηλωμένη αλλά ούτε και έχει το πρόθεμα `data-`.

Extbase & Fluid

Η παράμετρος **allowedTags** για το **f:format.stripTags**

- Η παράμετρος **allowedTags** που περιέχει μία λίστα από **HTML tags** τα οποία δε θα αφαιρεθούν μπορεί τώρα να χρησιμοποιηθεί στο **f:format.stripTags**
- Η σύνταξη της λίστας με τα **tags** είναι πανομοιότυπη με τη δεύτερη παράμετρο της συνάρτησης **PHP strip_tags** (βλ: http://php.net/strip_tags)

Πρόσβαση της **ObjectStorage** ως πίνακα στο **Fluid**

- Δημιουργεί ένα ψευδώνυμο του `toArray()` επιτρέποντας στη μέθοδο να καλείται ως `toArray()` το οποίο με τη σειρά του επιτρέπει στη μέθοδο να καλείται με διαφάνεια από το `ObjectAccess::getPropertyPath`, ενεργοποιώντας την πρόσβαση στο **Fluid** και σε άλλα μέρη
- Παράδειγμα: πάρε το 4ο στοιχείο

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

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

Κεφάλαιο 5:

Κατηρηγμένες/Αφηρημένες Μέθοδοι

Κατηρηγμένες/Αφηρημένες Μέθοδοι

Διάφορα

- Οι ακόλουθες επιλογές διαμόρφωσης έχουν αφαιρεθεί:
 - `$TYPO3_CONF_VARS['SYS']['t3lib_cs_utils']`
 - `$TYPO3_CONF_VARS['SYS']['t3lib_cs_convMethod']`

(η λειτουργικότητα ανιχνεύεται τώρα αυτόματα και το `mbstring` χρησιμοποιείται ως εκ προεπιλογής αν είναι διαθέσιμο)
- Η κατηρηγμένη ιδιότητα `TypoScript page.includeJSlibs` έχει αφαιρεθεί. Χρησιμοποιήστε την ιδιότητα `TypoScript page.includeJSlibs` (κεφαλαίο L) αντ' αυτής
- Η επιλογή `TypoScript config.renderCharset`, η οποία χρησιμοποιείτο ως σύνολο χαρακτήρων για εσωτερική μετατροπή με ένα `frontend request`, έχει αφαιρεθεί

Κεφάλαιο 6: Πηγές και Συγγραφείς

Πηγές και Συγγραφείς

Πηγές

Νέα της **TYPO3**:

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

Πληροφορίες Δημοσίευσης:

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

Αναφορές Προβλημάτων στην **TYPO3**:

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

TYPO3 Git Αποθήκες (**Repositories**):

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

Ομάδα **TYPO3 CMS** «Τι Νέο Υπάρχει»:

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

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

Υπό την άδεια της Creative Commons BY-NC-SA 3.0

