

TYPO3 CMS 7 LTS – What's New

TSconfig & TypoScript

Created by:
Patrick Lobacher and Michael Schams

Inleiding

TYPO3 CMS 7 LTS - What's New

De volgende pagina's richten zich op een bepaald onderwerp. Afhankelijk van je rol kunnen de volgende onderwerpen interessant voor je zijn:

	<i>Interface backend</i>	<i>TypoScript</i>	<i>Systeemwijzigingen</i>	<i>Extbase/Fluid</i>	<i>Verouderd/verwijderd</i>	<i>Sys.Beheer</i>
Redacteuren	X					
Sitebouwers		X	X		X	
Ontwikkelaars			X	X	X	
SysAdmins						X

Download alle versies van de **What's New Slides** van typo3.org

TScnfig & TypoScript

TypoScript wordt gebruikt om informatie te definiëren in een hiërarchische structuur om een TYPO3 CMS installatie te *configureren*. Er zijn veel eigenschappen en opties toegevoegd, gewijzigd, als verouderd aangemerkt of verwijderd in TYPO3 CMS 7 LTS.

Open Graph attributen worden nu standaard ondersteund en de integriteit van externe JavaScript-bestanden kunnen worden geverifieerd worden met een SRI-hash, om maar twee functies te noemen.

De pagina's **Verouderde/verwijderde functies** geven een overzicht van de verwijderde TypoScript opties.

TScnfig & TypoScript

TScnfig beschikbaar voor linkvalidatoren

- TScnfig-configuratie wordt gelezen
 - of vanuit de backend (als Linkvalidator wordt gebruikt)
 - of vanuit de configuratie van de taakplanner
- Voorbeeld: TScnfig, die kan worden gelezen door Linkchecker:
`mod.linkvalidator.mychecker.myvar = 1`
- TScnfig is dan beschikbaar als `$this->tsConfig`

TScnfig & TypoScript

Linkcontrole: Raportage verwijderde records

- In TYPO3 CMS < 7.0 waarschuwt linkhandler alleen over links naar niet-bestaande of verwijderde records
- Sinds TYPO3 CMS >= 7.0 schakelt de volgende TScnfig-instelling een waarschuwing in als links wijzen naar uitgeschakelde records

```
mod.linkvalidator.linkhandler.reportHiddenRecords = 1
```

TScnfig & TypoScript

RTE: meerdere CSS-klassen per stijl

- Moderne frameworks zoals Twitter Bootstrap vereisen meerdere CSS-klassen per HTML tag
Bijvoorbeeld: `Pas op`
- Meerdere CSS-klassen worden nu ondersteund waardoor redacteuren maar één stijl hoeven te kiezen

```
RTE.classes.[ *classname* ] {  
    .requires = lijst met CSS-klassen  
}
```

RTE: CSS-klasse als niet-selecteerbaar instellen

- CSS-lassen kunnen nu als "niet-selecteerbaar" ingesteld worden

```
// waarde "1" betekent: klas is selecteerbaar  
// waarde "0" maakt het niet-selecteerbaar  
RTE.classes.[ *classname* ] {  
    .selectable = 1  
}
```

TScnfig & TypoScript

RTE: meerdere CSS-bestanden insluiten

- Er kunnen meerdere CSS-bestanden ingesloten worden

```
RTE.default.contentCSS {  
    file1 = fileadmin/rte_stylesheet1.css  
    file2 = fileadmin/rte_stylesheet2.css  
}
```

- Als niets ingesteld wordt is de standaard:
`typo3/sysextr/rtehtmlarea/res/contentcss/default.css`

Tsconfig & TypoScript

Afhandeling exceptie wanneer cObject-en worden gerenderd (1)

- In TYPO3 CMS < 7.0 breekt de foutmelding als die ontstaat bij het renderen van inhoudsobjecten (bijv. USER) de hele frontend
- Sinds TYPO3 CMS >= 7.0 is afhandeling van excepties toegevoegd waarmee een bericht afgebeeld kan worden in plaats van het foute cObject

Tsconfig & TypoScript

Afhandeling exceptie wanneer cObject-en worden gerenderd (2)

```
# standaard exceptie-afhandeling (actief in context "production")
config.contentObjectExceptionHandler = 1

# configuratie van een klasse voor exceptie-afhandeling
config.contentObjectExceptionHandler =
    TYPO3\CMS\Frontend\ContentObject\Exception\ProductionExceptionHandler

# eigen foutmelding (toon willekeurige foutcode)
config.contentObjectExceptionHandler.errorMessage = Oeps, dat ging fout. Code: %s

# configuratie van exceptiecodes die niet afgehandeld worden
tt_content.login.20.exceptionHandler.ignoreCodes.10 = 1414512813

# deactivatie van exceptie-afhandeling voor specifieke plug-ins of inhoudobjecten
tt_content.login.20.exceptionHandler = 0

# ignoreCodes en errorMessage kunnen globaal ingesteld worden...
config.contentObjectExceptionHandler.errorMessage = Oeps, er ging iets fout. Code: %s
config.contentObjectExceptionHandler.ignoreCodes.10 = 1414512813

# ...of lokaal voor specifieke inhoudsobjecten
tt_content.login.20.exceptionHandler.errorMessage = Oeps, er ging iets fout. Code: %s
tt_content.login.20.exceptionHandler.ignoreCodes.10 = 1414512813
```

Tsconfig & TypeScript

StdWrap voor page.headTag

- TypeScript-instelling `page.headTag` heeft nu `stdWrap`-functionaliteit

```
page = PAGE
page.headTag = <head>
page.headTag.override = <head class="special">
page.headTag.override.if {
    isInList.field = uid
    value = 24
}
```

TScnfig & TypoScript

Asynchroon insluiten van JavaScript bestanden

- JavaScript-bestanden kunnen asynchroon worden geladen

```
page {  
    includeJS {  
        jsFile = /path/to/file.js  
        jsFile.async = 1  
    }  
}
```

- Dit heeft betrekking op:
 - includeJSlibs / includeJSLibs
 - includeJSFooterlibs
 - includeJS
 - includeJSFooter

TSconfig & TypoScript

HMENU-item selectie via additionalWhere

- TypoScript-cObject HMENU bevat nu een nieuwe eigenschap `additionalWhere`
- Dit laat een meer specifieke databasequery toe (zoals filtering)
- Voorbeeld:

```
lib.authormenu = HMENU
lib.authormenu.1 = TMENU
lib.authormenu.1.additionalWhere = AND author!=""
```

TScnfig & TypoScript

Aanvullende eigenschappen voor HMENU bladermenu's

- Twee nieuwe eigenschappen voor cObject HMENU (optie "special=browse") om menu-items nauwkeuriger te selecteren:
 - excludeNoSearchPages
 - includeNotInMenu
- Voorbeeld:

```
lib.browsemenu = HMENU
lib.browsemenu.special = browse
lib.browsemenu.special.excludeNoSearchPages = 1
lib.browsemenu.includeNotInMenu = 1
```

TScnfig & TypoScript

Meerdere HTTP headers

- HTTP headers kunnen nu worden ingesteld als een array (`config.additionalHeaders`)
- Dit maakt configuratie van meerdere headers tegelijk mogelijk

```
config.additionalHeaders {
    10 {
        # header string
        header = WWW-Authenticate: Negotiate
        # (optional) replace previous headers with the same name (default: 1)
        replace = 0
        # (optional) force HTTP response code
        httpResponseCode = 401
    }
    # set second additional HTTP header
    20.header = Cache-control: Private
}
```

TSconfig & TypeScript

Optie "auto" toegevoegd aan config.absRefPrefix

- TypeScript-instelling `config.absRefPrefix` kan gebruikt worden voor het herschrijven van URL's. Als een alternatief voor `config.baseUrl` (om een specifiek domein te configureren), kan `absRefPrefix` de siteroot automatisch detecteren:

```
config.absRefPrefix = auto
```

```
# ...instead of:
```

```
[ApplicationContext = Production]  
config.absRefPrefix = /
```

```
[ApplicationContext = Testing]  
config.absRefPrefix = /my_site_root/
```

Opmerking: de nieuwe optie kan gebruikt worden op omgevingen met meerdere domeinen om het meerdere malen cachen van dezelfde gegevens te voorkomen.

TSconfig & TypoScript

Tweeletterige ISO-code voor `sys_language` (1)

- Taalafhandeling maakt gebruik van databaserecords in `sys_language`, waarnaar gewoonlijk verwezen wordt in `sys_language_uid`
- In TYPO3 CMS 7.1 is de ISO 639-1 tweeletterige ISO-code geïntroduceerd:
 - Nieuw databaseveld: `sys_language.language_isocode`
 - Nieuwe TypoScript-optie: `sys_language_isocode`

Opmerking: **ISO 639** is een set normen door de Internationale Organisatie voor Standaardisatie. Een lijst van ISO 639-1-codes is te vinden op:

http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

Tsconfig & TypoScript

Tweeletterige ISO-code voor sys_language (2)

■ Voorbeeld:

```
# Danish by default
config.sys_language_uid = 0
config.sys_language_isocode_default = da

[globalVar = GP:L = 1]
  # ISO code stored in table sys_language (uid 1)
  config.sys_language_uid = 1
  # overwrite ISO code as required
  config.sys_language_isocode = fr
[GLOBAL]

page.10 = TEXT
page.10.data = TSFE:sys_language_isocode
page.10.wrap = <div class="main" data-language="|">
```

TSconfig & TypeScript

Custom TypeScript-conditions in de backend

- Ondersteuning voor speciaal gemaakte condities voor de **frontend** is reeds met TYPO3 CMS 7.0 geïntroduceerd
- Sinds TYPO3 CMS 7.1 is het ook mogelijk speciaal gemaakte condities in de **backend** te gebruiken
- De conditie moet afgeleid zijn van `AbstractCondition` en de implementatiemethode `matchCondition()`
- Voorbeeld van gebruik in TypeScript:

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition]
```

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = 7]
```

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = 7, != 6]
```

```
[BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = {$mysite.myconstant}]
```

TScnfig & TypoScript

Aanpassen iconen via PageTScnfig

- Waarde/label-paren van geselecteerde velden kunnen reeds geconfigureerd worden met de PageTScnfig-optie `addItems`
- Het is nu ook mogelijk om het **icon** van deze velden te beïnvloeden
 - Optie 1: door gebruik te maken van `addItems` en sub-property `.icon`
 - Optie 2: door gebruik te maken van `altIcons` (in het algemeen alle items)
- Voorbeeld:

```
TCEFORM.pages.doktype.addItems {
    10 = My Label
    10.icon = EXT:t3skin/icons/gfx/i/pages.gif
}
TCEFORM.pages.doktype.altIcons {
    10 = EXT:myext/icon.gif
}
```

TScnfig & TypoScript

Elementbrowser uitbreiden met koppelpunten

- Nieuwe UserTScnfig-optie `.append` stelt beheerders in staat om koppelpunten toe te voegen, in plaats van de geconfigureerde database koppelpunten van de gebruiker te vervangen
- Voorbeeld:

```
options.pageTree.altElementBrowserMountPoints = 20,31  
options.pageTree.altElementBrowserMountPoints.append = 1
```

Labels van selectievakjes en keuzerondjes overschrijven

- Labels van selectievakjes en keuzerondjes kunnen nu overschreven worden
- Voorbeeld:

```
// field with a single checkbox (use ".default")
TCEFORM.pages.hidden.altLabels.default = new label
TCEFORM.pages.hidden.altLabels.default = LLL:path/to/languagefile.xlf:individualLabel

// field with multiple checkboxes (0, 1, 2, 3...)
TCEFORM.pages.l18n_cfg.altLabels.0 = new label of first checkbox
TCEFORM.pages.l18n_cfg.altLabels.1 = new label of second checkbox
TCEFORM.pages.l18n_cfg.altLabels.2 = new label of third checkbox
...
```

TScnfig & TypoScript

Diversen (1)

- Breedte en hoogte van de elementbrowser kunnen geconfigureerd worden met UserTScnfig:

```
options.popupWindowSize = 400x900  
options.RTE.popupWindowSize = 200x200
```

- PageTScnfig: nieuwe RTE configuratie-eigenschap kan gebruikt worden om een standaard doel voor links van een bepaald type te configureren:

```
buttons.link.[type].properties.target.default
```

Het [type] kan bijv. een page, file, url, mail of spec zijn (extensies kunnen andere types beschikbaar maken)

TSconfig & TypoScript

Diversen (2)

- Standaard zijn sectiekoppen van zoekresultaten links. Het is nu mogelijk deze links uit te schakelen en secties als eenvoudige teksten te tonen

```
plugin.tx_indexedsearch.linkSectionTitles = 0
```

- `getData` heeft nu toegang tot `field`-gegevens (niet alleen arrays zoals `GPVar` en `TSFE`):

```
10 = TEXT  
10.data = field:fieldname|level1|level2
```

- TypoScript-instelling `config.pageTitle` heeft nu `stdWrap`-functionaliteit

```
# make value of <title> upper case  
page = PAGE  
page.config.pageTitle.case = upper
```


TScnfig & TypoScript

Flexibele URL voor voorvertoning (1)

- De voorvertoningslink van de knop "opslaan & bekijken" in de backend kan nu geconfigureerd worden.
- Dit wordt vaak gebruikt om blog- of nieuwsrecords vooraf te bekijken maar het is ook mogelijk om andere pagina's te definiëren om normale contentelementen vooraf te bekijken.

```
TCEMAIN.preview {
  <tabelnaam> {
    previewPageId = 123
    useDefaultLanguageRecord = 0
    fieldToParameterMap {
      uid = tx_myext_pi1[showUid]
    }
    additionalGetParameters {
      tx_myext_pi1[special] = HELLO
    }
  }
}
```

TSconfig & TypoScript

Flexibele URL voor voorvertoning (2)

- `previewPageId`:
UID van de pagina om het item vooraf te bekijken (bij het ontbreken hiervan wordt de huidige pagina gebruikt)
- `useDefaultLanguageRecord`:
of vertaalde records de UID van het standaard record moeten gebruiken (standaard geactiveerd, waarde: 1)
- `fieldToParameterMap`:
mapping om velden te selecteren van het record dat in de GET-parameters opgenomen moet worden
- `additionalGetParameters`:
toevoegen of overschrijven van alle mogelijke GET-parameters

Tsconfig & TypoScript

RTE Configuratie: Standaard doel

- eigenschap van RTE-configuratie voor PageTsconfig om een standaard doel in te stellen voor links van een bepaald type

```
buttons.link.[ type ].properties.target.default = ...
```

- Mogelijke linktypes zijn:
(extra types kunnen door extensies aangevuld worden)
 - page
 - file
 - url
 - mail
 - spec

Lege HTML tags verwijderen in HTMLparser

- Nieuwe functionaliteit in de HTMLparser om lege tags te verwijderen

```
stdWrap {
    // alle lege HTML tags verwijderen
    HTMLparser.stripEmptyTags = 1
    // alleen lege h2 en h3 tags verwijderen
    HTMLparser.stripEmptyTags.tags = h2, h3
}

RTE.default.proc.entryHTMLparser_db {
    stripEmptyTags = 1
    stripEmptyTags.tags = p
    stripEmptyTags.treatNonBreakingSpaceAsEmpty = 1
}
```

Note: Let op: HTMLparser verwijdert standaard alle onbekende tags.
Daarom is het misschien nuttig om deze te bewaren:

```
HTMLparser.keepNonMatchedTags = 1
```

Tsconfig & TypoScript

Allerlei

- Nieuwe eigenschap `buttons.abbreviation.removeFieldsets` in `PageTsconfig` om het schermpje voor afkortingen te configureren

Mogelijke waarden zijn:

`acronym`, `definedAcronym`, `abbreviation`, `definedAbbreviation`

```
buttons.abbreviation.removeFieldsets = acronym,definedAcronym
```

- Eigenschap `inlineLanguageLabel` van het object `PAGE` kan nu `LLL`: verwijzingen afhandelen

Tsconfig & TypoScript

Nieuwe stdWrap functie strtotime

- De nieuwe stdWrap eigenschap strtotime zet geformatteerde datums om naar Unix timestamps, bijv. voor datum berekeningen
- Geldige waarden zijn 1 of elke tekst die gebruikt kan worden als eerste argument voor de PHP functie strtotime()

```
date_as_timestamp = TEXT
date_as_timestamp {
    value = 2015-04-15
    strtotime = 1
}
```

```
next_weekday = TEXT
next_weekday {
    data = GP:selected_date
    strtotime = + 2 weekdays
    strftime = %Y-%m-%d
}
```

TSconfig & TypeScript

GPmerged in Conditions

- GP in TypeScript conditions geeft alleen de POST variabele, als POST en GET variabelen aanwezig zijn
- De nieuwe optie GPmerged combineert GET en POST

```
[globalVar = GPmerged:tx_demo|foo = 1]
  page.90 = TEXT
  page.90.value = DEMO
[global]
```

TScnfig & TypoScript

Nieuwe opties voor stdWrap.case

- Opties uppercamelcase en lowercamelcase zijn toegevoegd aan stdWrap.case
- Example:

```
tt_content = CASE
tt_content {
    key.field = CType
    my_custom_ctype =< lib.userContent
    my_custom_ctype {
        file = EXT:site_base/Resources/Private/Templates/SomeOtherTemplate.html
        settings.extraParam = 1
    }
    default =< lib.userContent
    default {
        file = TEXT
        file.field = CType
        file.stdWrap.case = uppercamelcase
        file.wrap = EXT:site_base/Resources/Private/Templates/|.html
    }
}
```


Tsconfig & TypeScript

Eigenschap `integrity` toegevoegd voor JavaScript bestanden(1)

- Nieuwe eigenschap `integrity` bij het invoegen van JS bestanden om een SRI hash in te stellen waarmee de bron geverifieerd wordt (SRI: Sub-Resource Integrity, zie volgende pagina)
- Dit beïnvloedt de TypeScript PAGE eigenschappen `page.includeJSLibs`, `page.includeJSFooterlibs`, `includeJS` en `includeJSFooter`
- Voorbeeld:

```
page {  
  includeJS {  
    jquery = https://code.jquery.com/jquery-1.11.3.min.js  
    jquery.external = 1  
    jquery.disableCompression = 1  
    jquery.excludeFromConcatenation = 1  
    jquery.integrity = sha256-7LkWEzqTdpEfELxcZZlS6wAx5Ff13zZ83lY02/ujj7g=  
  }  
}
```

TSconfig & TypeScript

Eigenschap `integrity` toegevoegd voor JavaScript bestanden(2)

- SRI is een W3C specificatie waarmee vastgesteld kan worden dat bestanden op een server van derden niet gecompromiteerd zijn
- Aanmaken van integrity hashes:
 - Optie 1: <https://srihash.org>
 - Optie 2: gebruik het volgende shell commando

```
cat FILENAME.js | openssl dgst -sha256 -binary | openssl enc -base64 -A
```

- Meer informatie:
 - <http://www.w3.org/TR/SRI/>

Data-Provider voor Backend Layouts (1)

- Backend layouts kunnen nu via page TSconfig gedefinieerd worden en ook opgeslagen worden in bestanden. Voorbeeld:

```
mod {
  web_layout {
    BackendLayouts {
      exampleKey {
        title = Voorbeeld
        config {
          backend_layout {
            colCount = 1
            rowCount = 2
            rows {
              1 {
                columns {
                  1 {
                    name = LLL:EXT:frontend/ ... /locallang_ttc.xlf:colPos.I.3
                    colPos = 3
                    colspan = 1
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  [...]
}
```


TScnfig & TypoScript

Meta-tags uitgebreid

- Optie `page.meta` kent nu ook [Open Graph](#) attributen

```
page {
  meta {
    X-UA-Compatible = IE=edge,chrome=1
    X-UA-Compatible.attribute = http-equiv
    keywords = TYPO3
    # <meta property="og:site_name" content="TYPO3" />
    og:site_name = TYPO3
    og:site_name.attribute = property
    description = Inspiring people to share
    og:description = Inspiring people to share
    og:description.attribute = property
    og:locale = en_GB
    og:locale.attribute = property
    og:locale:alternate {
      attribute = property
      value.1 = fr_FR
      value.2 = de_DE
    }
    refresh = 5; url=http://example.com/
    refresh.attribute = http-equiv
  }
}
```

TSconfig & TypoScript

languageField standaard aanwezig

- Bij TypoScript optie `select` (gebruikt in bijvoorbeeld cObject CONTENT) moest `languageField` expliciet ingesteld worden
- Dit is niet meer nodig; de instelling wordt automatisch opgehaald uit TCA informatie

```
config.sys_language_uid = 2
page.10 = CONTENT
page.10 {
    table = tt_content
    select.where = colPos=0

    # volgende regel is niet meer nodig:
    #select.languageField = sys_language_uid

    renderObj = TEXT
    renderObj.field = header
    renderObj.htmlSpecialChars = 1
}
```

TScnfig & TypoScript

Caching van losse content

- Vanaf TYPO3 CMS 7.4 is er caching voor losse inhoud die - net zoals `stdWrap.cache` - ook werkt met COA objecten (zoals "Magento Block Caching")

```
page = PAGE
page.10 = COA
page.10 {
    cache.key = coaout
    cache.lifetime = 60
    #stdWrap.cache.key = coastdWrap
    #stdWrap.cache.lifetime = 60
    10 = TEXT
    10 {
        cache.key = mycurrenttimestamp
        cache.lifetime = 60
        data = date : U
        strftime = %H:%M:%S
        noTrimWrap = |10: | |
    }
}
[...]
```

```
[...]
20 = TEXT
20 {
    data = date : U
    strftime = %H:%M:%S
    noTrimWrap = |20: | |
}
}
```

TScnfig & TypoScript

Aantal elementen in lijst

- De `stdWrap` optie `split` heeft een nieuwe eigenschap `returnCount`
- Deze telt bijvoorbeeld het aantal elementen in een kommagescheiden lijst
- De volgende code geeft bijvoorbeeld 9:

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


TScnfig & TypoScript

Sorteervolgorde van tabellen in Lijstmodule

- Een nieuwe TScnfig optie `mod.web_list.tableDisplayOrder` voor de "Lijst" module
- Met deze optie is de volgorde waarin tabellen worden weergegeven in te stellen
- Termen `before` en `after` stellen volgorde relatief aan andere tabellen in

Syntax:

```
mod.web_list.tableDisplayOrder {
    <tableName> {
        before = <tableA>, <tableB>, ...
        after = <tableA>, <tableB>, ...
    }
}
```

Voorbeeld:

```
mod.web_list.tableDisplayOrder {
    be_users.after = be_groups
    sys_filemounts.after = be_users
    pages_language_overlay.before = pages
    fe_users.after = fe_groups
    fe_users.before = pages
}
```

TSconfig & TypeScript

Content-Language in HTTP Header

- De HTTP header Content-language: XX wordt automatisch verstuurd met als "XX" de ISO code van de `sys_language_content` configuratie
- Met `config.disableLanguageHeader = 1` kan dit uitgeschakeld worden (de Content-language header wordt helemaal niet verstuurd)

Tsconfig & TypeScript

Optie Recursive voor bestandsverzameling

- Bestandsverzamelingen gebaseerd op een map hebben nu een optie om alle bestanden recursief op te halen
- De optie is ook beschikbaar in het TypeScript Object FILES

```
filecollection = FILES
filecollection {
  folders = 1:images/
  folders.recursive = 1
  renderObj = IMAGE
  renderObj {
    file.import.data = file:current:uid
  }
}
```

TSconfig & TypeScript

Extensie `.ts` voor statische sjablonen

- In TYPO3 CMS < 7.4 de enige toegestane bestandsnamen voor statische TypeScript-sjablonen waren:
 - `constants.txt`
 - `setup.txt`
 - `include_static.txt`
 - `include_static_files.txt`
- Voor `constants` en `setup` kan nu ook de bestandsextensie `.ts` gebruikt worden
- In dit verband heeft `.ts` prioriteit boven `.txt`

TSconfig & TypoScript

Knop opslaan & tonen

- De knop "opslaan & tonen" kan nu met TSconfig uitgeschakeld worden
- TSconfig `TCEMAIN.preview.disableButtonForDokType` bevat een kommagescheiden lijst met "doktypes"
- Standaard is "254, 255, 199" (Opslagmap, Prullenbak en Menuscheiding)
- Het gevolg is dat de knop "opslaan & tonen" niet meer in mappen en prullenbakken getoond wordt

TSconfig & TypoScript

stdWrap voor treatIdAsReference

- Het object `getImgResource` kent de optie `treatIdAsReference`, die definieert dat UID's worden gezien als UID's van een `sys_file_reference` in plaats van een `sys_file`.
- Optie `treatIdAsReference` kent nu `stdWrap` functionaliteit

TSconfig & TypeScript: Data Processors

Introductie Data Processors

- De volgende Data Processors zijn nieuw. Ze bieden een flexibele afhandeling van kommagescheiden lijsten, array's, bestanden, enz.:
 - SplitProcessor
 - CommaSeparatedValueProcessor
 - FilesProcessor
 - GalleryProcessor
 - DatabaseQueryProcessor
- Zie: `TYPO3\CMS\Frontend\DataProcessing`

TSconfig & TypeScript: Data Processors

De SplitProcessor

- De "SplitProcessor" zorgt voor het splitsen op een scheidingsteken naar een array:

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:site_default/Resources/Private/Template/Default.html
page.10.dataProcessing.2 = TYPO3\CMS\Frontend\DataProcessing\SplitProcessor
page.10.dataProcessing.2 {
    if.isTrue.field = bodytext
    delimiter = ,
    fieldName = bodytext
    removeEmptyEntries = 1
    filterIntegers = 1
    filterUnique = 1
    as = keywords
}
```

- Mogelijke toepassing in Fluid:

```
<f:for each="{keywords}" as="keyword">
  <li>Keyword: {keyword}</li>
</f:for>
```


TSconfig & TypeScript: Data Processors

De CommaSeparatedValueProcessor (1)

- De "CommaSeparatedValueProcessor" splitst waarden op een scheidingsteken naar een tweedimensionale array:

```
page.10 = FLUIDTEMPLATE
page.10.file = EXT:site_default/Resources/Private/Template/Default.html
page.10.dataProcessing.4 = TYPO3\CMS\Frontend\DataProcessing\CommaSeparatedValueProcessor
page.10.dataProcessing.4 {
    if.isTrue.field = bodytext
    fieldName = bodytext
    fieldDelimiter = |
    fieldEnclosure =
    maximumColumns = 2
    as = table
}
```

- Handig om CSV-bestanden te gebruiken of bijvoorbeeld tt_content data van het CType "table"

Voorbeeld volgt op volgende pagina

TSconfig & TypeScript: Data Processors

De CommaSeparatedValueProcessor (2)

■ Mogelijke toepassing in Fluid:

```
<table>
  <f:for each="{table}" as="columns">
    <tr>
      <f:for each="{columns}" as="column">
        <td>
          {column}
        </td>
      </f:for>
    <tr>
  </f:for>
</table>
```

TSconfig & TypeScript: Data Processors

De FilesProcessor (1)

- De "FilesProcessor" behandelt referenties naar bestanden, bestanden of bestanden in een map of verzameling om ze in de frontend te kunnen gebruiken

```
tt_content.image.20 = FLUIDTEMPLATE
tt_content.image.20 {
    file = EXT:myextension/Resources/Private/Templates/ContentObjects/Image.html
    dataProcessing.10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
    dataProcessing.10 {
        references.fieldName = image
        references.table = tt_content
        files = 21,42
        collections = 13,14
        folders = 1:introduction/images/,1:introduction/posters/
        folders.recursive = 1
        sorting = description
        sorting.direction = descending
        as = myfiles
    }
}
```

Voorbeeld volgt op volgende pagina

TSconfig & TypoScript: Data Processors

De FilesProcessor (2)

■ Mogelijke toepassing in Fluid:

```
<ul>
  <f:for each="{myfiles}" as="file">
    <li>
      <a href="{file.publicUrl}">{file.name}</a>
    </li>
  </f:for>
</ul>
```

TSconfig & TypeScript: Data Processors

De GalleryProcessor

- De "GalleryProcessor" geeft de max grootte van een reeks bestanden

```
tt_content.text_media.20 = FLUIDTEMPLATE
tt_content.image.20 {
    file = EXT:myextension/Resources/Private/Templates/ContentObjects/Image.html
    dataProcessing {
        10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
        20 = TYPO3\CMS\Frontend\DataProcessing\GalleryProcessor
        20 {
            filesProcessedDataKey = files
            mediaOrientation.field = imageorient
            numberOfColumns.field = imagecols
            equalMediaHeight.field = imageheight
            equalMediaWidth.field = imagewidth
            maxGalleryWidth = 1000
            maxGalleryWidthInText = 1000
            columnSpacing = 0
            borderEnabled.field = imageborder
            borderWidth = 0
            borderPadding = 10
            as = gallery
        }
    }
}
```

TSconfig & TypeScript: Data Processors

De DatabaseQueryProcessor (1)

- De "DatabaseQueryProcessor" haalt gegevens uit de database op

```
tt_content.mycontent.20 = FLUIDTEMPLATE
tt_content.mycontent.20 {
    file = EXT:myextension/Resources/Private/Templates/ContentObjects/MyContent.html
    dataProcessing.10 = TYPO3\CMS\Frontend\DataProcessing\DatabaseQueryProcessor
    dataProcessing.10 {
        if.isTrue.field = records
        table = tt_address
        colPos = 1
        pidInList = 13,14
        as = myrecords
        dataProcessing {
            10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
            10 {
                references.fieldName = image
            }
        }
    }
}
```

Voorbeeld volgt op volgende pagina

TSconfig & TypeScript: Data Processors

De DatabaseQueryProcessor (2)

■ Mogelijke toepassing in Fluid:

```
<ul>
  <f:for each="{myrecords}" as="record">
    <li>
      <f:image image="{record.files.0}" ></f:image>
      <a href="{record.data.www}">{record.data.first_name} {record.data.last_name}</a>
    </li>
  </f:for>
</ul>
```

TScnfig & TypoScript

Voorwaardelijke TypoScript-cludes

- INCLUDE_TYPOSCRIPT heeft nu een extra (optionele) eigenschap "condition". Alleen als aan de voorwaarde is voldaan, wordt het bestand (of de map) gebruikt.

```
// TypoScript alleen laden als de gebruiker is ingelogd:  
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypoScript/feuser.ts"  
  condition="[loginUser = *]">
```

```
// TypoScript alleen laden wanneer de applicationContext overeenkomt:  
<INCLUDE_TYPOSCRIPT: source="FILE:EXT:my_extension/Configuration/TypoScript/staging.ts"  
  condition="applicationContext = /^Production\\/Staging\\/Server\\d+$/">
```


Tsconfig & TypoScript

TCA-optie om de ouderdom te verbergen

- Met TCA-optie `disableAgeDisplay` kan de ouderdom verbergen worden (bijvoorbeeld: "30-08-2015 (-27 dagen)")

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

- Als voorwaarde moet het type een input-veld zijn en `eval` moet `date` hebben

TSconfig & TypeScript

Inline taallabels met TypeScript (1)

- XLF-taalbestanden kunnen worden ingelezen en geconverteerd in een array
- Dat maakt het bijvoorbeeld mogelijk om taallabels in JavaScript te gebruiken
- De volgende 3 optionele parameters worden ondersteund:
 - `selectionPrefix`:
gebruik alleen labels waarvan de index met deze tekst begint
 - `stripFromSelectionName`:
tekst die van elke index wordt verwijderd
 - `errorMode`:
foutafhandeling wanneer het bestand niet wordt gevonden:
0: log (standaard), 1: negeren, 3: gooi een exception

TSconfig & TypeScript

Inline taallabels met TypeScript (2)

■ Voorbeeld:

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

■ Uitvoer:

```
<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="22 936 223 958" data-label="Page-Footer"><p>TYPO3 CMS 7 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>
```

TScnfig & TypoScript

Workspacevoorbeelden via TScnfig

- Standaard genereert TYPO3 CMS in de workspace alleen voorbeeldpagina's voor de tabellen tt_content, pages en pages_language_overlay
- Dat kan nu worden aangepast met PageTScnfig:

```
# gebruik pagina 123 voor het voorbeeld (alle tabellen)
options.workspaces.previewPageId = 123
```

```
# gebruik het veld pid voor het voorbeeld (alle tabellen)
options.workspaces.previewPageId = field:pid
```

```
# gebruik pagina 123 voor het voorbeeld (tabel tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = 123
```

```
# gebruik het veld pid voor het voorbeeld (tabel tx_myext_table)
options.workspaces.previewPageId.tx_myext_table = field:pid
```

Afbeeldingskwaliteit per sourceCollection

- De afbeeldingskwaliteit van elke sourceCollection kan worden aangepast
- Deze instelling heeft voorrang op de configuratie in de Installatie-module (die wordt opgeslagen in bestand LocalConfiguration.php)
- Voorbeeld:

```
# voor kleine retina-afbeeldingen
tt_content.image.20.1.sourceCollection.smallRetina.quality = 80

# voor grote retina-afbeeldingen
tt_content.image.20.1.sourceCollection.largeRetina.quality = 65
```

TSconfig & TypeScript

Aantal elementen in een lijst

- Een nieuwe eigenschap `returnCount` is toegevoegd aan `stdWrap`-functie `split`
- Dit maakt het mogelijk om het aantal elementen in een kommagescheiden lijst te tellen
- De volgende code heeft bijvoorbeeld 9 als uitvoer:

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

TSconfig & TypoScript

Afhandeling van Backend Layouts (1)

- De afhandeling van backend layouts voor de frontend is vereenvoudigd
- In TypoScript kan de nieuwe optie `pagelayout` worden gebruikt
- Voorbeeld:

```
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
    }
}
```

(vervolg op volgende pagina)

Afhandeling van Backend Layouts (2)

- Daarbij vervangt `key.data = pagelayout` de volgende code:

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


TSconfig & TypoScript

Diversen

- In TYPO3 CMS 7.4 werd de `stdWrap`-functie `bytes` geïntroduceerd
- In TYPO3 CMS 7.5 is aan `bytes` de eigenschap `base` toegevoegd, die het mogelijk maakt om bij de bestandsgrootte te rekenen met 1000 of 1024

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

```
bytes.base = 1000
```

indexed_search: Parameters

- De volgende TypoScript-eigenschappen kunnen nu ingesteld worden voor indexed_search:

```
titleCropAfter = 50
titleCropSignifier = ...
summaryCropAfter = 180
summaryCropSignifier =
hrefInSummaryCropAfter = 60
hrefInSummaryCropSignifier = ...
markupSW_summaryMax = 300
markupSW_postPreLgd = 60
markupSW_postPreLgd_offset = 5
markupSW_divider = ...
```

- Mogelijke sleutels:
 - plugin.tx_indexedsearch.results.
 - plugin.tx_indexedsearch.settings.results.
- Elke eigenschap heeft stdWrap-functionaliteit

TSconfig & TypoScript

`indexed_search`: Instelbaar pad-scheidingsteken

- Nieuwe TypoScript-optie `breadcrumbWrap` is toegevoegd
- Hiermee kan het scheidingsteken voor het broodkruimelpad in `indexed_search`-resultaten ingesteld worden
- Deze optie ondersteunt de **option split**-syntax uit TypoScript. Standaardconfiguratie is `"/`:

```
plugin.tx_indexedsearch.settings.breadcrumbWrap = / || /
```

TSconfig & TypoScript

indexed_search: Instelbare no_cache-parameter

- Nieuwe TypoScript-optie is toegevoegd:
forwardSearchWordsInResultLink.no_cache
- Dit bepaalt of de no_cache-parameter wordt toegevoegd aan paginalinks voor indexed_search

```
// voor Indexed Search Extbase plugins  
plugin.tx_indexedsearch.settings.forwardSearchWordsInResultLink.no_cache = 1
```

```
// voor plugins gebaseerd op AbstractPlugin  
plugin.tx_indexedsearch.forwardSearchWordsInResultLink.no_cache = 1
```

Bronnen en auteurs

Bronnen en auteurs

Bronnen

TYPO3 Nieuws:

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

Publicatie-informatie:

- https://wiki.typo3.org/Category:ReleaseNotes/TYPO3_7.x
- [INSTALL.md](#) en [Changelog](#)
- `typo3/sysexst/core/Documentation/Changelog/*`

TYPO3 Bug-/Issuetracker:

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

TYPO3 Git Repositories:

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

TYPO3 CMS What's New Slides:

Patrick Lobacher

(Onderzoek, Informatieverzamelen en Duitse versie)

Michael Schams

(Projectleider en Engelse versie)

Vertalingen en bijdragen door:

Andrey Aksenov, Paul Blondiaux, Pierrick Caillon, Sergio Catalá,
Ben van't Ende, Jigal van Hemert, Sinisa Mitrovic, Michel Mix, Angeliki Plati,
Nena Jelena Radovic en Roberto Torresani

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

Licensed under Creative Commons BY-NC-SA 3.0

