TYPO3 CMS 8.2 and 8.3 – What’s New
Summary of the new features, changes and improvements

Created by:
Patrick Lobacher and Michael Schams

05/September/2016
Creative Commons BY-NC-SA 3.0
Chapter Overview

Introduction

Backend User Interface

TSconfig & TypoScript

In-Depth Changes

Extbase & Fluid

 Deprecated/Removed Functions

Sources and Authors
Introduction

The Facts
Introduction

TYPO3 CMS 8.2 and 8.3 – The Facts

TYPO3 CMS 8.2

- Release date: 05 July 2016
- Release type: Sprint Release
- Slogan: Upgrades

TYPO3 CMS 8.3

- Release date: 30 August 2016
- Release type: Sprint Release
- Slogan: Frontend Editing on Steroids
Introduction

System Requirements

- PHP: version 7
- MySQL: version 5.5 to 5.7
- Disk space: min 200 MB
- PHP settings:
  - memory_limit >= 128M
  - max_execution_time >= 240s
  - max_input_vars >= 1500
  - compilation option --disable-ipv6 must not be used

- The backend requires Microsoft Internet Explorer 11 or later, Microsoft Edge, Google Chrome, Firefox, Safari or any other modern, compatible browser
Introduction

Development and Release Timeline
Introduction

TYPO3 CMS Roadmap

Release dates and their primary focus:

- v8.0  22/Mar/2016  Adding last minute things
- v8.1  03/May/2016  Cloud Integration
- v8.2  05/Jul/2016  Upgrades
- v8.3  30/Aug/2016  Frontend Editing on Steroids
- v8.4  18/Oct/2016  to be determined
- v8.5  20/Dec/2016  Integrator Support
- v8.6  14/Feb/2017  to be determined
- v8.7  04/Apr/2017  LTS Preparation

https://typo3.org/typo3-cms/roadmap/
https://typo3.org/news/article/kicking-off-typo3-v8-development/
Introduction

Installation

- Official installation procedure under Linux/Mac OS X
  (DocumentRoot for example /var/www/site/htdocs):

  $ cd /var/www/site
  $ wget --content-disposition get.typo3.org/8.3
  $ tar xzf typo3_src-8.3.0.tar.gz
  $ cd htdocs
  $ ln -s ../typo3_src-8.3.0 typo3_src
  $ ln -s typo3_src/index.php
  $ ln -s typo3_src/typo3
  $ touch FIRST_INSTALL

- Symbolic links under Microsoft Windows:

  - Use junction under Windows XP/2000
  - Use mklink under Windows Vista and Windows 7
Introduction

Upgrade to TYPO3 CMS 8.x

- Upgrades only possible from TYPO3 CMS 7.6 LTS
- TYPO3 CMS < 7.6 LTS should be updated to TYPO3 CMS 7.6 LTS first

- Upgrade instructions:
  
  http://wiki.typo3.org/Upgrade#Upgrading_to_8.3

- Official TYPO3 guide "TYPO3 Installation and Upgrading":
  
  http://docs.typo3.org/typo3cms/InstallationGuide

- General approach:
  
  - Check minimum system requirements (PHP, MySQL, etc.)
  - Review `deprecation_*.log` in old TYPO3 instance
  - Update all extensions to the latest version
  - Deploy new sources and run Install Tool -> Upgrade Wizard
  - Review startup module for backend users (optionally)
Introduction

PHP Version 7

- PHP 7.0 is the minimum requirement for TYPO3 CMS 8.x
- TYPO3 will support subsequent PHP 7 releases as they come out
- This version raise gives a significant performance boost to the overall system
- Not only backend editors will notice a more fluent interface, but the new all-time record for a full cached page call in the frontend is below 7 milliseconds now, which is approximately 40% faster compared to running the very same website with PHP version 5.5
- We also started using new features from this PHP version, for instance the cryptographically secure pseudo-random generators are in active use already
Chapter 1:

Backend User Interface
"Clear Cache" Entry in Context Menu

A new entry has been added to the context menu of the page tree. The item is located inside "Page Actions" and allows to clear the cache of the selected page.
Backend User Interface

Ogg, Flac and Opus Support

Support for the following open formats has been added to the media field: ogg, flac and opus
Chapter 2:

TSconfig & TypoScript
Now it is possible to change the width and height of the popup, which is used in the edit panel of EXT:feedit by using User TSconfig:

```php
options.feedit.popupHeight = 700
options.feedit.popupWidth = 900
```
Chapter 3: In-Depth Changes
In-Depth Changes

Add Linkservice for Unified Referencing Syntax (1)

- Resources within TYPO3 have been referenced using multiple, different forms of syntax in the past.
- TYPO3 now supports a modern and future-proof way of referencing resources using an extensible and expressive syntax which is easy to understand.
- The next slides explain the syntax using the following simple page link:

  t3://page?uid=13&campaignCode=ABC123
In-Depth Changes

Add Linkservice for Unified Referencing Syntax (2)

- The syntax consists of three parts:
  - Namespace (t3://)
    The namespace is fixed to t3:// to ensure the "LinkService" is executed to parse the URN.
  - Resource handler key (page)
    The resource handler key is a list of handlers available in TYPO3. At the time of writing the following handlers exist: page, file and folder. More keys can be configured in an associative array, where the key is the handler key and the value is a class implementing the LinkHandlerInterface:
    $TYPO3_CONF_VARS['SYS']['linkHandler']
In-Depth Changes

Add Linkservice for Unified Referencing Syntax (3)

■ ...and the 3rd part:

■ Resource parameters (?uid=13&campaignCode=ABC123)
  These are the specific identification parameters that are used by any handler. Note that these may carry additional parameters in order to configure the behavior of any handler.
In-Depth Changes

DebuggerUtility::var_dump(1)

- The property visibility information has been added to DebuggerUtility::var_dump() for each object property in the dump.
- If a closure is part of the debugging object, the source code of the closure is rendered, too.

See example on the next slide.
In-Depth Changes

DebuggerUtility::var_dump (2)

Extbase Variable Dump

TYPO3\CMS\Core\Database\DatabaseConnection [prototype object]

dbdebugOutput => public 0 (integer)
dbdebug_lastBuiltQuery => public '' (0 chars)
dbstore_lastBuiltQuery => public FALSE
dbexplainOutput => public 0 (integer)
dbdatabaseHost => protected 'localhost' (9 chars)
dbdatabasePort => protected 3306 (integer)
dbdatabaseSocket => protected '' (0 chars)
dbdatabaseName => protected 'typo383' (7 chars)
dbdatabaseUsername => protected 'root' (4 chars)
dbdatabasePassword => protected 'secure7bl' (9 chars)
dbpersistentDatabaseConnection => protected FALSE
dbcconnectionCompression => protected FALSE
dbcconnectionCharSet => protected 'utf8' (4 chars)
dbcinitializeCommandsAfterConnect => protected array (empty)
dbismConnected => protected FALSE
dblink => protected NULL
dbdefault_charset => public 'utf8' (4 chars)
dbpreProcessHookObjects => protected array (empty)
dbpostProcessHookObjects => protected array (empty)
In-Depth Changes

Import Module Disabled for Non Admin Users

- The import module of EXT:impexp is now disabled for non-admin users by default.

- For non-admin users, who need that functionality, the following User TSconfig option can be set:

  options.impexp.enableImportForNonAdminUser = 1

Warning: this can become a security problem in TYPO3 versions 6.2 and 7.6 and should only be enabled for trustworthy backend users.
In-Depth Changes

Hooks and Signals (1)

- A new hook allows registering custom result browser implementations.
- This approach allows to override the default implementation of AbstractPlugin::pi_list_browsresults() for either all extensions or only for specific ones.
- The hook can be registered in ext_localconf.php:

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS'][\TYPO3\CMS\Frontend\Plugin\AbstractPlugin::class]['pi_list_browsresults'][1463475262] = \Vendor\ExtensionKey\Hook\ResultBrowserHook::class
```
In-Depth Changes

Hooks and Signals (2)

- With the migration to Doctrine, hook buildQueryParameters has been introduced in class DatabaseRecordList.
- This hook replaces the hook makeQueryArray from the deprecated method AbstractDatabaseRecordList::makeQueryArray.
- Using the new hook allows modifying the parameters used to query the database for records to be shown in the record list view.
- The hook can be registered in ext_localconf.php:

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS'][Typo3\CMS\Recordlist\RecordList\DatabaseRecordList::class]['buildQueryParameters'][]
```

- ...and implements the public method buildQueryParametersPostProcess
In-Depth Changes

Miscellaneous

- SVGs and D3 rendering
  - As part of the ExtJS removal from the TYPO3 core, the tree within the form editing has been re-worked
  - Rendering is based on SVGs and D3 now, which comes with a significant performance boost
  - Re-working the page tree the same way is planned for the near future

- Extension icons can be stored in the following directory now: Resources/Public/Icons/<filename> (where <filename> can be: Extension.png, Extension.svg or Extension.gif)

- The new option backendFavicon in the Extension Manager configuration makes it possible to change the favicon of the backend.
Chapter 4:
Extbase & Fluid
Extbase & Fluid

Add IconForRecordViewHelper

A new ViewHelper to render icons for records has been added

```xml
<core:iconForRecord table="sys_template" row="{templateRecord}"/>

// output:
<span class="t3js-icon icon icon-size-small icon-state-default icon-mimetypes-x-content-template" data-identifier="mimetypes-x-content-template">
    <span class="icon-markup">
        <img src="/typo3/sysext/core/Resources/Public/Icons/T3Icons/mimetypes/mimetypes-x-content-template.svg" width="16" height="16">
    </span>
</span>
```

TYPO3 CMS 8.2 and 8.3 - What’s New
Interceptors in *Fluid Standalone* were introduced to be able to change the template output.

The Fluid API already allows for registration of custom interceptors. Now it is possible to define custom interceptors by using the following option:

```php
$GLOBALS[‘TYPO3_CONF_VARS’][‘fluid’][‘interceptors’]
```

Interceptors registered here are added to the Fluid parser configuration.
Add Fluid Interceptor Registration (2)

- Register own interceptor to fluid parser configuration

```php
$GLOBALS['TYPO3_CONF_VARS']['SYS']['fluid']['interceptors']['\TYPO3\CMS\Fluid\Core\Parser\Interceptor\DebugInterceptor::class'] = \TYPO3\CMS\Fluid\Core\Parser\Interceptor\DebugInterceptor::class;
```

- Class code:

```php
use TYPO3Fluid\Fluid\Core\Parser\InterceptorInterface;
use TYPO3Fluid\Fluid\Core\Parser\ParsingState;
use TYPO3Fluid\Fluid\Core\Parser\SyntaxTree\NodeInterface;

class DebugInterceptor implements InterceptorInterface
{
    public function process(NodeInterface $node, $interceptorPosition, ParsingState $parsingState)
    {
        return $node;
    }

    public function getInterceptionPoints()
    {
        return [];
    }
}
```
Chapter 4: Deprecated/Removed Functions
Deprecated/Removed Functions

Extbase Query Cache Removed

- The PHP-based query cache functionality within the Extbase persistence layer has been removed.
- The following public methods within the Extbase persistence layer have been removed:
  - Typo3DbBackend->quoteTextValueCallback()
  - Typo3DbBackend->injectCacheManager()
  - Interface definition in QuerySettingsInterface->getUseQueryCache
- The according cache configuration has no effect anymore:
  - $TYPO3_CONF_VARS[SYS][cache][cacheConfigurations][extbase_typo3dbbackend_queries]
Deprecation/Removed Functions

**Extbase: Prepared Statement Query Option**

- The option to use prepared statements within the Extbase persistence has been removed.

- The following methods have been removed from the QuerySettingsInterface, as the database abstraction layer will take care of prepared statements automatically:
  - `getUsePreparedStatement()`
  - `usePreparedStatement()`
Deprecated/Removed Functions

**Miscellaneous (1)**

- Two RTE-related TSconfig options, which do not allow align and class attributes when converting content from the RTE to the database, have been removed:
  RTE.default.proc.skipAlign and RTE.default.proc.skipClass

- The "xcache" core cache backend has been dropped because PHP version 7 does no support xcache anymore

- The following TypoScript `stdWrap`-function has been declared **deprecated**:
  `removeBadHTML`
Deprecated/Removed Functions

Miscellaneous (2)

- One-line comments in TypoScript must start with two forward slashes as the first non-blank character and should be followed by a whitespace. A single slash as a comment is deprecated now.

- The Extbase table column cache (which was used to store all database fields of all database tables) has been removed: `extbase_typo3dbbackend_tablecolumns`

- The associated configuration variable can be removed: `TYPO3_CONF_VARS[SYS][caching][cacheConfigurations][extbase_typo3dbbackend_tablecolumns]`
Chapter 5:
Sources and Authors
Sources and Authors

Sources

**TYPO3 News:**

- [http://typo3.org/news](http://typo3.org/news)

**Release Infos:**

- [http://wiki.typo3.org/TYP03_CMS_8.2.0](http://wiki.typo3.org/TYP03_CMS_8.2.0) and [TYP03_CMS_8.3.0](http://wiki.typo3.org/TYP03_CMS_8.3.0)
- INSTALL.md and ChangeLog
- typo3/sysext/core/Documentation/Changelog/8.2/* and /8.3/*

**TYPO3 Bug-/Issuetracker:**


**TYPO3 and Fluid Git Repositories:**

- [https://git.typo3.org/Packages/TYP03.CMS.git](https://git.typo3.org/Packages/TYP03.CMS.git)
- [https://github.com/TYPO3Fluid/Fluid](https://github.com/TYPO3Fluid/Fluid)
Sources and Authors

TYPO3 CMS What’s New Team:

Pierrick Caillon, Sergio Catala, Richard Haeser, Jigal van Hemert, Patrick Lobacher, Michel Mix, Sinisa Mitrovic, Nena Jelena Radovic, Michael Schams and Roberto Torresani

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

Licensed under Creative Commons BY-NC-SA 3.0