TYPO3 Version 11 LTS – What's New

Summary of the new features, changes and improvements

Created by:
Michael Schams

05/November/2021

TYPO3

TYPO3 Version 11 LTS – What’s New

Chapter Overview

Introduction

Backend User Interface

Multi-Factor Authentication (MFA)

Deep Links in the Backend

Changes for Integrators and Developers

Extbase and Fluid

Deprecated/Removed Functions

Miscellaneous

Installation and Upgrade

Sources and Authors

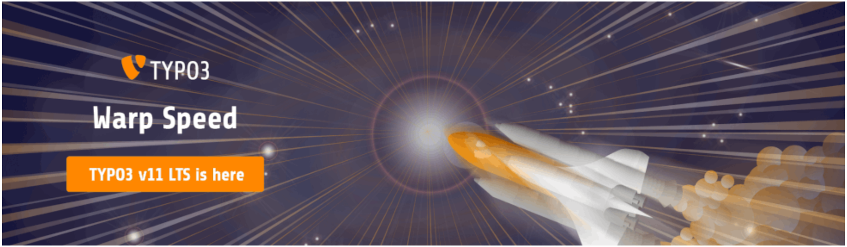
Introduction

Key facts and figures about TYPO3 v11 LTS

Introduction

TYPO3 Version 11 LTS - The Facts

- Release date: 05 October 2021
- Release type: Long-term Support Release (LTS)



Introduction

Executive Summary

No matter your industry, role, or what your daily tasks are. TYPO3 v11 LTS is the CMS for everyone. Editors and integrators will be delighted with many

improvements to backend usability and performance. TYPO3 now also ships with a secure, easy-to-use, and flexible multi-factor authentication (MFA) solution out-of-the-box. We kept our promise to continuously improve the TYPO3

ecosystem and migrated more components to use well-known PSR standards in TYPO3 v11 LTS, which now also supports PHP v7.4 and PHP v8.0. As a long-term

support release (LTS), this version will be maintained by the community for at least three years. Extended long-term support (ELTS) will be available from the TYPO3 GmbH to add another three years of support, planning time, and peace of mind.

Introduction

System Requirements

- Webserver such as Apache, Nginx, IIS, etc.
- PHP version 7.4 or 8.0
- PHP settings:
  - `memory_limit` >= 256M
  - `max_execution_time` >= 240s
  - `max_input_vars` >= 1500
  - compilation option `--disable-ipv6` must not be used

Introduction

System Requirements

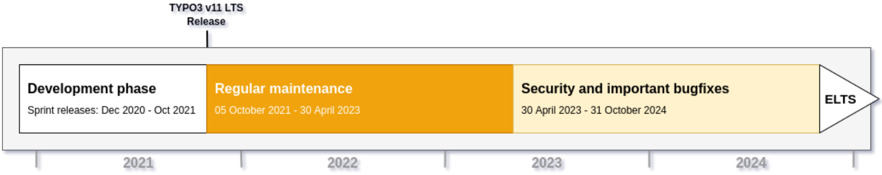
- All databases supported by **Doctrine DBAL** are also supported by TYPO3. For example:

MySQL MariaDB PostgreSQL Microsoft SQL Server SQLite

- Minimum disk space required: 200 MB
- The backend supports all modern browsers such as Microsoft Edge, Google Chrome, Firefox, Safari, or any other compatible browser

Introduction

Long-term Support


TYPO3 v11 LTS
Release
Development phase
Sprint releases: Dec 2020 - Oct 2021
Regular maintenance
05 October 2021 - 30 April 2023
Security and important bugfixes
30 April 2023 - 31 October 2024
ELTS
2021
2022
2023
2024

- Regular maintenance and bugfixes until April 2023
- Security and critical bugfixes until October 2024

Extended Support

The TYPO3 GmbH offers extended long-term support (ELTS) for TYPO3 v11 LTS for another three years

Introduction

TYPO3 v11 Releases

Published v11 sprint releases:

v11.0	22/Dec/2020	New system requirements and breaking changes
v11.1	23/Feb/2021	Multi-factor authentication
v11.2	04/May/2021	Link sharing for TYPO3 backend
v11.3	13/Jul/2021	UX improvements and PHP v8 support
v11.4	07/Sep/2021	Feature freeze

Long-term support relase:

v11.5	05/Oct/2021	LTS Release (long-term support)

Backend User Interface

The TYPO3 administration interface is now better than ever

Backend User Interface

Bootstrap Version 5

The TYPO3 backend now uses Bootstrap version 5.

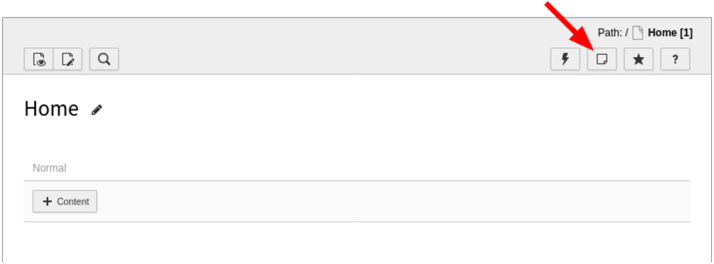
The decision to use version 5 (which is still in development at the time of writing) puts TYPO3 at the forefront of modern content management systems from a UI perspective.

Backend users won't need to learn how to use a new user interface with Bootstrap 5. They might, however, notice a few minor usability changes in TYPO3 v11.0

Backend User Interface

System Notes

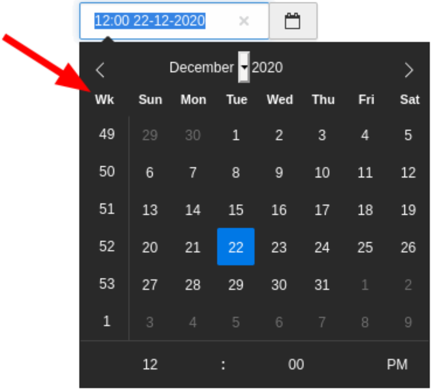
A new button has been added to the top right of the Page, List, and Info module to add system notes directly.



Backend User Interface

System Notes

The DateTimePicker now also shows week numbers in the calendar.



12:00 22-12-2020

December 2020

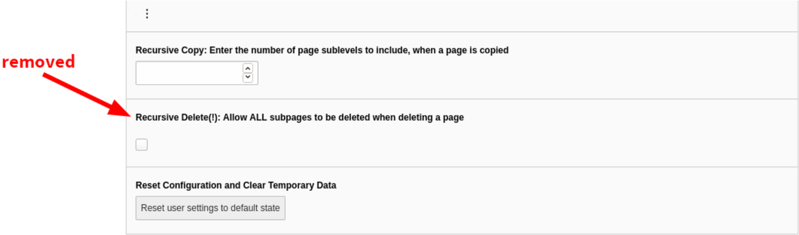
Wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat
49	29	30	1	2	3	4	5
50	6	7	8	9	10	11	12
51	13	14	15	16	17	18	19
52	20	21	22	23	24	25	26
53	27	28	29	30	31	1	2
1	3	4	5	6	7	8	9

12 : 00 PM

Backend User Interface

Recursive Page Deletion

Backend users can now delete pages with sub-pages in the page tree (recursive page deletion). In previous TYPO3 versions, users had to explicitly enable this function which is now always enabled.



Recursive Copy: Enter the number of page sublevels to include, when a page is copied

Recursive Delete(!): Allow ALL subpages to be deleted when deleting a page

□

Reset Configuration and Clear Temporary Data

Backend User Interface

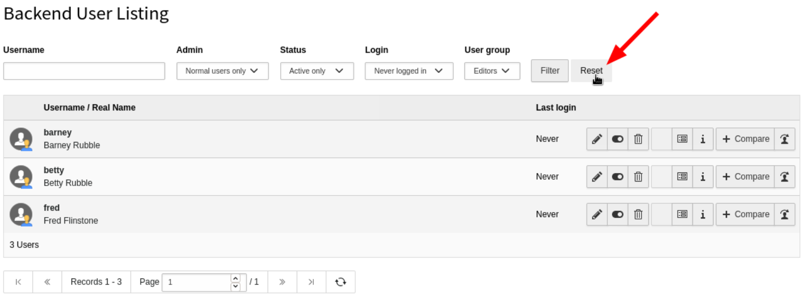
Context Menu

- The context menus are now accessible via keyboard.
- For example:
  - **Shift+F10** opens the context menu.
  - **Arrow keys**, **Home**, and **End** to navigate through the menu.
  - **Enter**, **Space** actives items or opens submenus.
- This change follows the best practices as described in WAI-ARIA Authoring Practices 1.1 by the W3C.

Backend User Interface

Reset Backend User Filter

The module "Backend Users" features a button to reset the user filter now.



Backend User Listing

Username	Admin	Status	Login	User group		
	Normal users only	Active only	Never logged in	Editors	Filter	Reset

Username / Real Name	Last login	
barney Barney Rubble	Never	+ Compare
betty Betty Rubble	Never	+ Compare
fred Fred Flinstone	Never	+ Compare

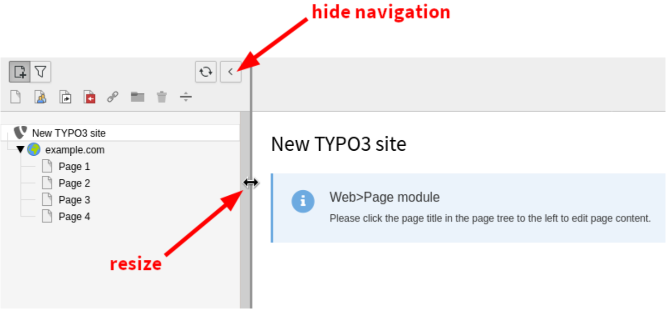
3 Users

Records 1 - 3 | Page 1 / 1

Backend User Interface

Resizable Navigation Component

Backend users can now resize the *navigation area* such as the pagetree component and file list. The icon to collapse/re-open has been moved inside the area.



New TYPO3 site

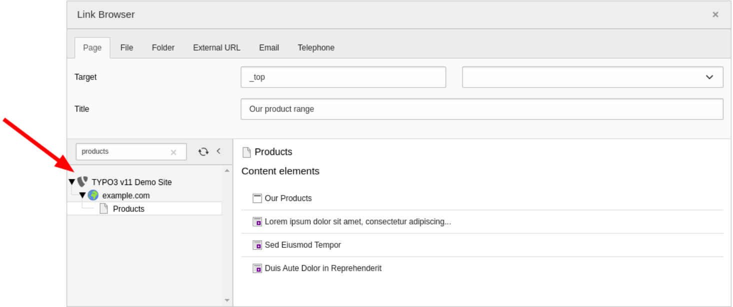
Web>Page module

Please click the page title in the page tree to the left to edit page content.

Backend User Interface

Page and File Tree in the Record Selector

The Record Selector (also known as *Element Browser* or *Link Browser*) now also features the SVG-based page tree and file list.



Target: _top

Title: Our product range

Products

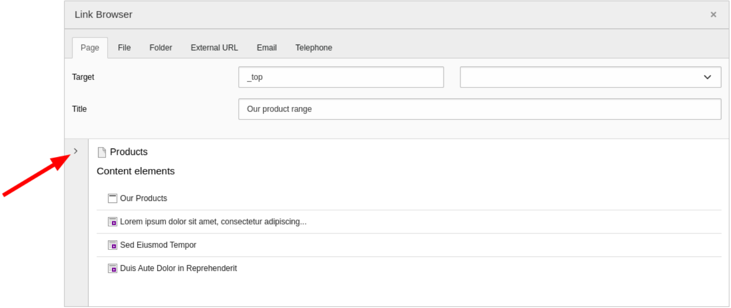
Content elements

- Our Products
- Lorem ipsum dolor sit amet, consectetur adipiscing...
- Sed Eiusmod Tempor
- Duis Aute Dolor in Reprehenderit

Backend User Interface

Resizable Navigation

Backend users benefit from all features such as filtering, collapsing, resizing, etc. which are available in all modals.


Link Browser ×

Page | File | Folder | External URL | Email | Telephone

Target: _top

Title: Our product range

Products

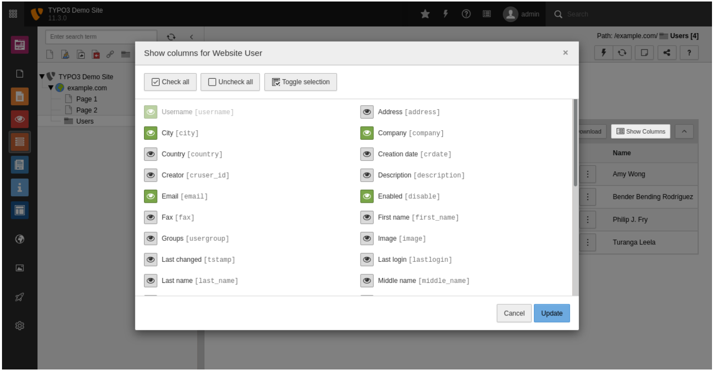
Content elements

- Our Products
- Lorem ipsum dolor sit amet, consectetur adipiscing...
- Sed Eiusmod Tempor
- Duis Aute Dolor in Reprehenderit

Backend User Interface

List Module: Selectable Columns

Backend users can now show/hide columns for each table under **Web -> List** without the need to switch to the single table view.



Show columns for Website User

☑ Check all · □ Uncheck all · Toggle selection

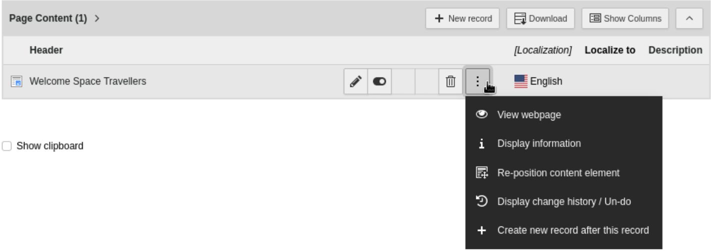
Username [username]	Address [address]
City [city]	Company [company]
Country [country]	Creation date [crdate]
Creator [cruser_id]	Description [description]
Email [email]	Enabled [disable]
Fax [fax]	First name [first_name]
Groups [usergroup]	Image [image]
Last changed [tstamp]	Last login [lastlogin]
Last name [last_name]	Middle name [middle_name]

Cancel · Update

Backend User Interface

List Module: Extended View

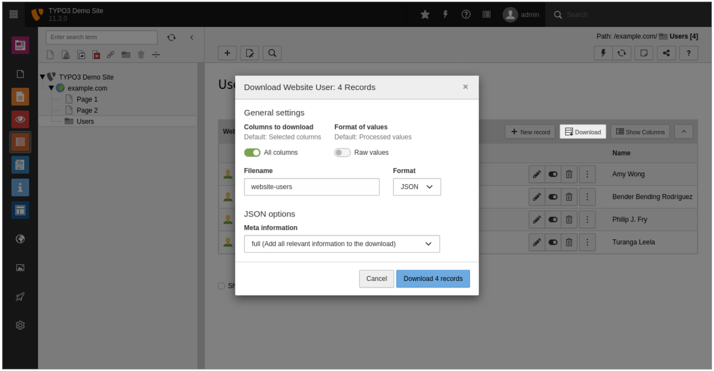
Secondary actions for each table under **Web -> List** are now available as a dropdown menu. This was previously known as the *extended table view*.



Backend User Interface

List Module: Download Data

Backend users can now trigger download in CSV and JSON format through a single button in the table header of each table in **Web -> List**.



Download Website User: 4 Records

General settings

Columns to download
Default: Selected columns
● All columns

Format of values
Default: Processed values
○ Raw values

Filename: website-users

Format: JSON

JSON options

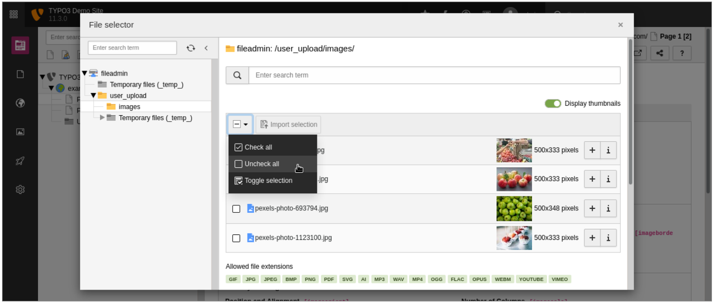
Meta information: full (Add all relevant information to the download)

Cancel | Download 4 records

Backend User Interface

File Selector

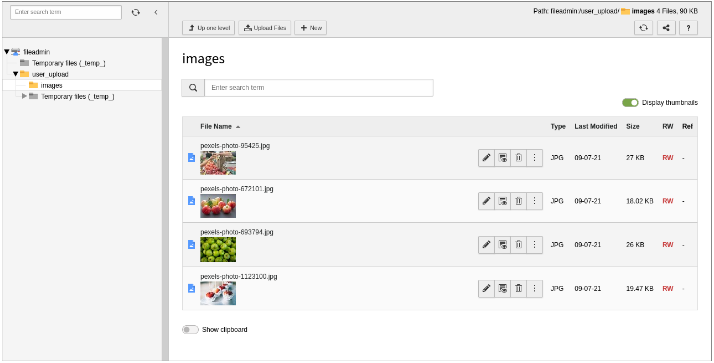
The File Selector (e.g. when selecting an image in the "Text and Media" content element) now features a checkbox on the left-hand side for each file. Backend users can check/uncheck/toggle elements easily.



Backend User Interface

Filelist Module

The Filelist backend module has an updated visual appearance that is more conistent with the List module.



Path: fileadmin:/user_upload/ images 4 Files, 90 KB

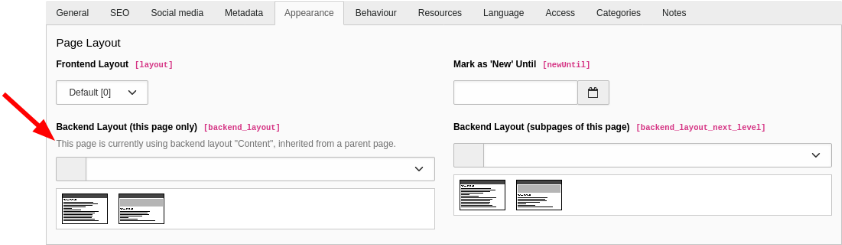
images

File Name	Type	Last Modified	Size	RW	Ref
pexels-photo-95425.jpg	JPG	09-07-21	27 KB	RW	-
pexels-photo-672101.jpg	JPG	09-07-21	18.02 KB	RW	-
pexels-photo-693794.jpg	JPG	09-07-21	26 KB	RW	-
pexels-photo-1123100.jpg	JPG	09-07-21	19.47 KB	RW	-

Backend User Interface

Backend Layouts

A hint is shown in the page properties, if a page inherits the backend layout from a parent page.



Page Layout

Frontend Layout [layout]

Default [0]

Mark as 'New' Until [newUntil]

Backend Layout (this page only) [backend_layout]

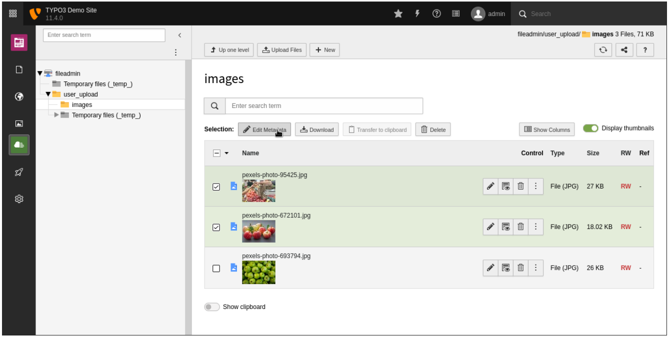
This page is currently using backend layout "Content", inherited from a parent page.

Backend Layout (subpages of this page) [backend_layout_next_level]

Backend User Interface

Filelist Module

It is now also possible to execute actions in the **File -> Filelist** module directly without transferring files to TYPO3's clipboard first.

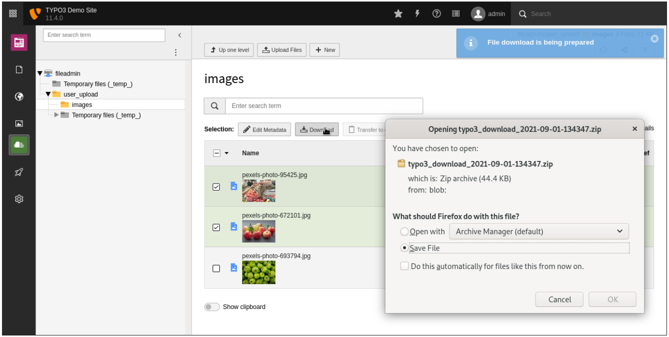


	Name	Control	Type	Size	RW	Ref
☑	pexels-photo-95425.jpg		File (JPG)	27 KB	RW	-
☑	pexels-photo-672101.jpg		File (JPG)	18.02 KB	RW	-
□	pexels-photo-693794.jpg		File (JPG)	26 KB	RW	-

Backend User Interface

Filelist Module

Backend users can now download multiple files or folders in one go by instructing TYPO3 to create a ZIP archive of them.



File download is being prepared

Opening typo3_download_2021-09-01-134347.zip

You have chosen to open:

typo3_download_2021-09-01-134347.zip

which is: Zip archive (44.4 KB)
from: blob:

What should Firefox do with this file?

○ Open with Archive Manager (default)

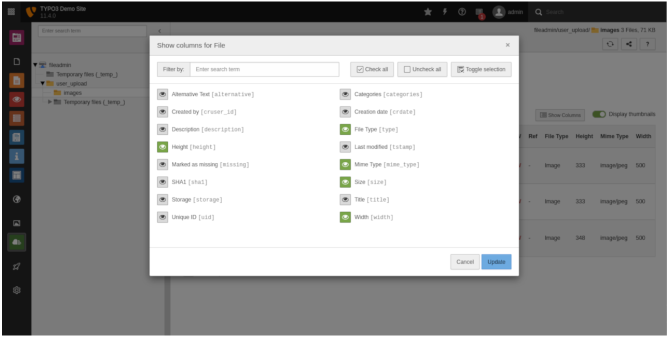
● Save File

□ Do this automatically for files like this from now on.

Backend User Interface

Filelist Module

The Column Selector is now also available in **File -> Filelist**. Backend users can select which fields for files/folders should be shown.



Show columns for File ×

Filter by: Enter search term

☑ Check all | ☐ Uncheck all | Toggle selection

- ☐ Alternative Text [alternative]
- ☐ Created by [cruser_id]
- ☐ Description [description]
- ☑ Height [height]
- ☐ Marked as missing [missing]
- ☐ SHA1 [sha1]
- ☐ Storage [storage]
- ☐ Unique ID [uid]
- ☐ Categories [categories]
- ☐ Creation date [crdate]
- ☑ File Type [type]
- ☐ Last modified [tstamp]
- ☑ Mime Type [mime_type]
- ☑ Size [size]
- ☐ Title [title]
- ☑ Width [width]

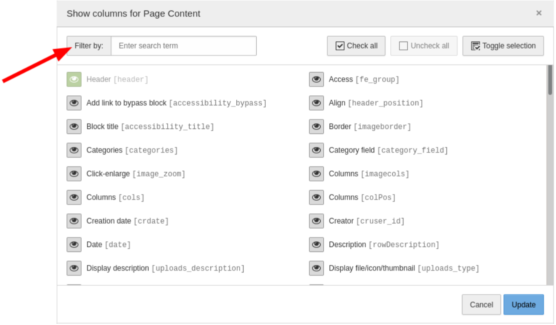
Cancel | Update

Ref	File Type	Height	Mime Type	Width
-	Image	333	image/jpeg	500
-	Image	333	image/jpeg	500
-	Image	348	image/jpeg	500

Backend User Interface

Columns Selector Filter

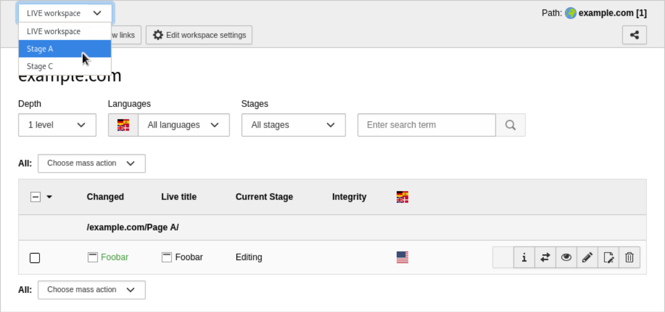
The column selector in the record list now features a filter at the top. This makes it easier for backend users to enable/disable the right items in particular for tables with many columns.



Backend User Interface

Workspaces Module

The usability of the **Workspaces module** has been improved. A dropdown box at the top allows backend users, for example, to switch between workspaces easily.



Path: example.com [1]

example.com

Depth: 1 level

Languages: All languages

Stages: All stages

Enter search term

All: Choose mass action

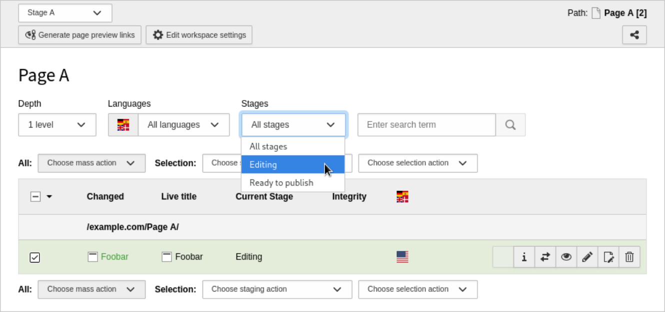
☐	Changed	Live title	Current Stage	Integrity	
	/example.com/Page A/				
☐	Foobar	Foobar	Editing		

All: Choose mass action

Backend User Interface

Workspaces Module

A new dropdown menu in the **Workspaces module** enables users to filter items for a specific stage, e.g. "Editing" or "Ready to publish".



Path: Page A [2]

Page A

Depth: 1 level

Languages: All languages

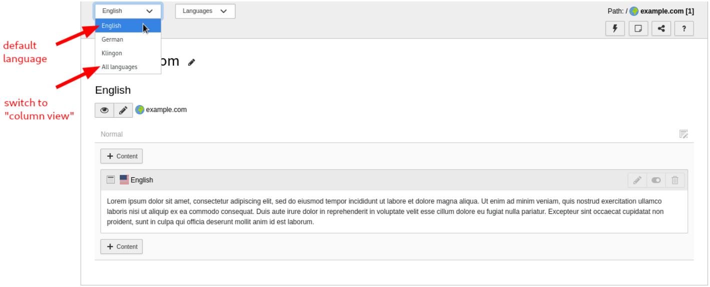
Stages: All stages

☐	Changed	Live title	Current Stage	Integrity
	/example.com/Page A/			
☑	Foobar	Foobar	Editing	

Backend User Interface

Languages/Translations

The "Languages" view under **Web -> Page** is now more consistent with the column view. The default language only shows content elements of this language. Backend users can use the "All languages" item to switch to the column view to show available languages.

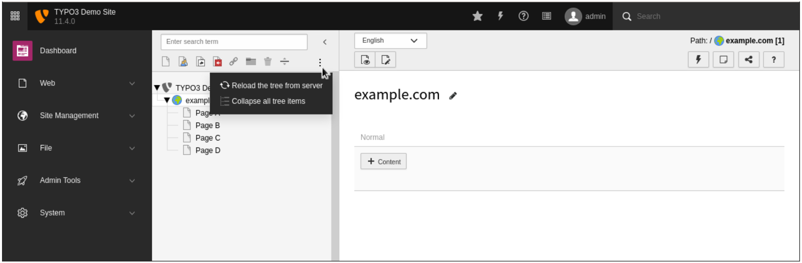


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Backend User Interface

Additional Actions

All SVG-based trees (e.g. the page tree, file/folder list, etc.) now feature additional actions. This menu contains the function to reload the tree and to collapse all items.

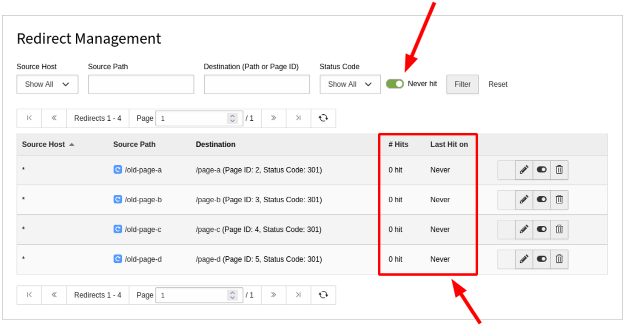


Backend User Interface

Redirects

Backend users can now enable a "never hit" option in **Site Management -> Redirects**. This filter shows only redirects that were never accessed.

Note: the function requires the "redirects hit count" feature activated.



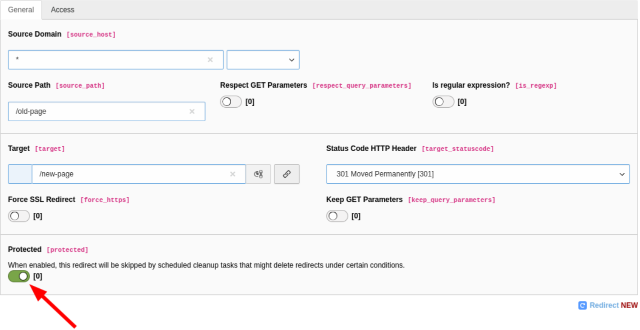
Redirect Management

Source Host	Source Path	Destination	# Hits	Last Hit on
*	/old-page-a	/page-a (Page ID: 2, Status Code: 301)	0 hit	Never
*	/old-page-b	/page-b (Page ID: 3, Status Code: 301)	0 hit	Never
*	/old-page-c	/page-c (Page ID: 4, Status Code: 301)	0 hit	Never
*	/old-page-d	/page-d (Page ID: 5, Status Code: 301)	0 hit	Never

Backend User Interface

Clean-up Redirects

If enabled and configured, TYPO3 removes redirects (for example redirects older than a specific number of days). To mark certain redirects and prevent their deletion, a "protected" switch has been added.



Source Domain [source_host]

*

Source Path [source_path]

/old-page

Respect GET Parameters [respect_query_parameters]

○ [0]

Is regular expression? [is_regexp]

○ [0]

Target [target]

/new-page

Status Code HTTP Header [target_statuscode]

301 Moved Permanently [301]

Force SSL Redirect [force_https]

○ [0]

Keep GET Parameters [keep_query_parameters]

○ [0]

Protected [protected]

When enabled, this redirect will be skipped by scheduled cleanup tasks that might delete redirects under certain conditions.

● [0]

Backend User Interface

Miscellaneous

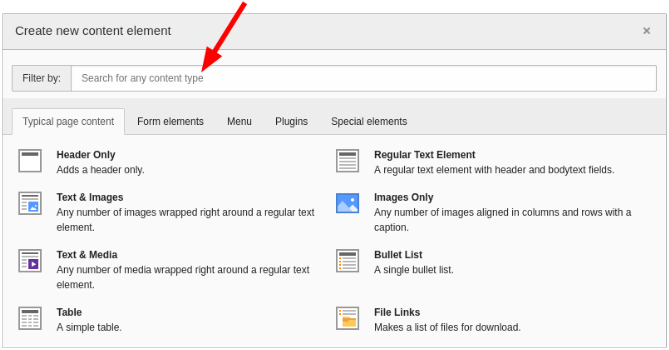
Further user interface improvements include:

- Backend users can select a row in the file list and list view by simply clicking on it (no need to click on the checkbox)
- Keyboard shortcuts can be used in multi record selections:
    - **Shift** key: check/uncheck a range
    - **Ctrl/Option** key: toggle current selection
- The "select all", "toggle selection", and "select range" functions are also available in List module, and no longer bound to the clipboard functionality (therefore, the "clipboard" column has been removed)

Backend User Interface

New Record Wizard

When selecting new content elements, an input field lets backend users filter for specific records.



Create new content element ×

Filter by: Search for any content type

Typical page content | Form elements | Menu | Plugins | Special elements

Header Only
Adds a header only.

Regular Text Element
A regular text element with header and bodytext fields.

Text & Images
Any number of images wrapped right around a regular text element.

Images Only
Any number of images aligned in columns and rows with a caption.

Text & Media
Any number of media wrapped right around a regular text element.

Bullet List
A single bullet list.

Table
A simple table.

File Links
Makes a list of files for download.

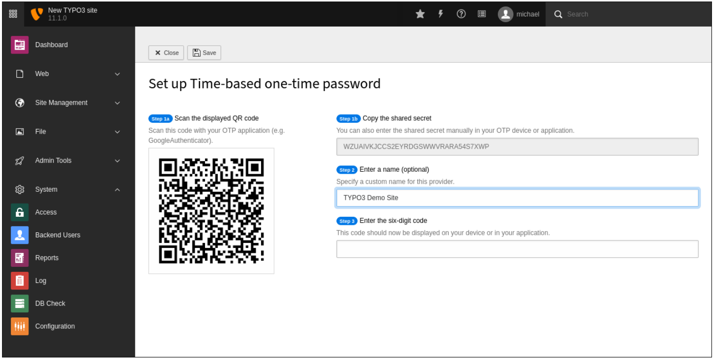
Multi-Factor Authentication (MFA)

Flexible and secure authentication for the TYPO3 backend

Multi-Factor Authentication (MFA)

Multi-factor Authentication

If activated, backend users can now add a second authentication factor to their login process (for example a time-based one-time password).



Set up Time-based one-time password

Step 1a Scan the displayed QR code

Scan this code with your OTP application (e.g. GoogleAuthenticator).

Step 1b Copy the shared secret

You can also enter the shared secret manually in your OTP device or application.

WZUAIVKJCCS2EYRDGSWWVRARA54S7XWP

Step 2 Enter a name (optional)

Specify a custom name for this provider.

TYPO3 Demo Site

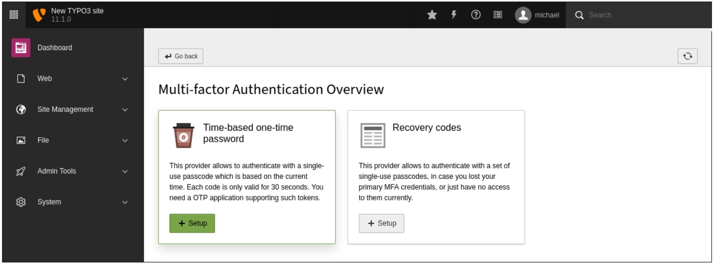
Step 3 Enter the six-digit code

This code should now be displayed on your device or in your application.

Multi-Factor Authentication (MFA)

Multi-factor Authentication

- The TYPO3 Core includes two MFA providers by default:
    - Time-based one-time password (TOTP)
    - Recovery codes (*fallback provider*)



Multi-factor Authentication Overview

Time-based one-time password

This provider allows to authenticate with a single-use passcode which is based on the current time. Each code is only valid for 30 seconds. You need a OTP application supporting such tokens.

Recovery codes

This provider allows to authenticate with a set of single-use passcodes, in case you lost your primary MFA credentials, or just have no access to them currently.

Multi-Factor Authentication (MFA)

Multi-factor Authentication

- Additional providers can be added as TYPO3 extensions.
- Providers also feature a locking functionality (e.g. if OTP was entered wrongly multiple times).
- The following global configuration controls if MFA is required:

  `$GLOBALS['TYPO3_CONF_VARS']['BE']['requireMFA']`

  - 0: MFA is optional (default)
  - 1: MFA is required for all users
  - 2: MFA is required for non-admin users only
  - 3: MFA is required for admin users only

Multi-Factor Authentication (MFA)

Multi-factor Authentication

- TSconfig can be used to disable the MFA field for backend users:

  ```
  setup.fields.mfaProviders.disabled = 1
  ```

- It is possible to restrict the available providers for some users or groups (see "Access Rights/List"), or by TSconfig:

  ```
  auth.mfa.disableProviders := addToList(totp)
  ```

- The following global configuration can be used to recommend a provider:

  ```
  $GLOBALS['TYPO3_CONF_VARS']['BE']['recommendedMfaProvider']
  ```

- Integrators can overwrite the global configuration by TSconfig:

  ```
  auth.mfa.recommendedProvider = totp
  ```

Multi-Factor Authentication (MFA)

Multi-factor Authentication

- Extension developers are encouraged to review the PHP classes in: `typo3/sysext/core/Classes/Authentication/Mfa/`
- Some initial demo extensions for example:
  - mfa_yubikey
  - mfa_hotp
  - mfa_webauthn (planned to be implemented in the TYPO3 Core in v11 LTS)
- Read more about the current API in the changelog.

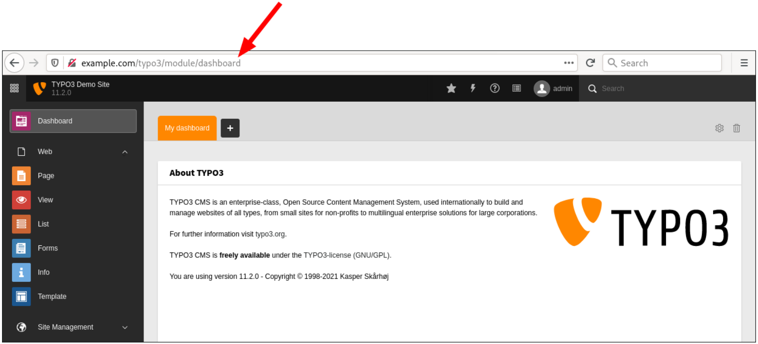
Deep Links in the Backend

Direct links to modules, pages, content elements, etc.

Deep Links in the Backend

Deep Links in the Backend

Deep linking is now possible in the TYPO3 backend. This lets users share links with other users and it makes working with the backend much more efficient.



About TYPO3

TYPO3 CMS is an enterprise-class, Open Source Content Management System, used internationally to build and manage websites of all types, from small sites for non-profits to multilingual enterprise solutions for large corporations.

For further information visit typo3.org.

TYPO3 CMS is **freely available** under the TYPO3-license (GNU/GPL).

You are using version 11.2.0 - Copyright © 1998-2021 Kasper Skårhøj

Deep Links in the Backend

Deep Links in the Backend

Some examples of deep links:

- (Pre)view page (ID=123):

  `https://example.com/typo3/module/web/ViewpageView?id=123`

- Edit the Site Configuration:

  `https://example.com/typo3/module/site/configuration?action=edit`

- Edit a content element (ID=42):

  `https://example.com/typo3/record/edit?edit[tt_content][42]=edit`

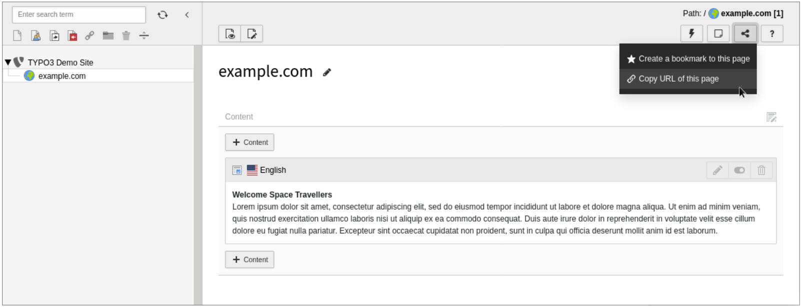
- Open the module "Backend Users":

  `https://example.com/typo3/module/system/BeuserTxBeuser`

Deep Links in the Backend

Sharing Backend Links

Sharing TYPO3 backend deeplinks is even easier now. Backend users can copy the URL of the page to their clipboard with two mouse clicks.



Welcome Space Travellers

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Changes for Integrators and Developers

In-depth TYPO3 Core improvements and new features

Changes for Integrators and Developers

Extension Manager

- The Extension Manager now features an *all versions* view that shows detailed information about extensions.
- The following database table is not used anymore and was removed:
  - `tx_extensionmanager_domain_model_repository`
- The (undocumented) support for extension-in-extension installations has been removed. TYPO3 now ignores the following directory:
  - `EXT:my_extension/Initialisation/Extensions/`
- As we have phased out the proprietary format `.t3x` in favor of `.zip` files, the option to upload `.t3x` files has been removed.

Changes for Integrators and Developers

Resizable Text Area

- As the user setting "**Make text areas flexible**" has been removed, the following User TSconfig settings are obsolete:
  - `setup.default.resizeTextareas_Flexible`
  - `setup.override.resizeTextareas_Flexible`
  - `setup.fields.resizeTextareas_Flexible.disabled`
- TYPO3 integrators should update the TypoScript code and remove these settings, if they exist.

Changes for Integrators and Developers

Miscellaneous

- As TYPO3 now always uses the Fluid-based page module, the feature toggle to switch between different implementations has been removed.
- All functionality of the system extension `EXT:about` have been migrated into the main backend extension, and the extension removed.
- The following system-wide setting has been removed to harden the security (it is now deemed as activated):

  `$GLOBALS['TYPO3_CONF_VARS']['BE']['lockBeUserToDBmounts']`
- The following TSconfig setting has been removed:

  `mod.web_layout.disableAdvanced`

Changes for Integrators and Developers

Clean-up Redirects

- A new CLI command can delete redirects that match certain criteria:
  - Age of records in days
  - Domain(s), comma-separated
  - Hit count
  - Status code(s), comma-separated
  - Path pattern
- The command can also run as a Scheduler task.
- Redirects can be marked as "protected" to prevent their deletion.
- For example:

```
vendor/bin/typo3 redirects:cleanup --age 90 --path "/foobar%"
```

(this deletes all redirects that start with `/foobar` and are older than 90 days)

Changes for Integrators and Developers

TypoScript Condition: `pagelayout`

- New TypoScript condition to check for the backend layout of a page is now available in the frontend and backend.
- For example:

```
# Using backend_layout records
[tree.pagelayout == 2]
  page.1 = TEXT
  page.1.value = Layout 2
[END]

# Using TSconfig provider of backend layouts
[tree.pagelayout == "pagets__Home"]
  page.1 = TEXT
  page.1.value = Layout Home
[END]
```

Changes for Integrators and Developers

IP-Lock for Backend Users

- TYPO3 v11 still supports to lock a backend user session to an IP address or IP address range by using the following option:
  `$TYPO3_CONF_VARS[BE][lockIP]`
- In previous TYPO3 versions, administrators were able to overwrite this configuration for *specific* backend users if required.
- This user-specific configuration, along with the following database field and the TCA definition, have been removed:
  `be_users.disableIPlock`

Changes for Integrators and Developers

Admin Panel

- The Admin Panel offers a list of frontend user groups to choose from to render the page under this context.
- This list now only contains **enabled** user groups (hidden/disabled groups are not listed anymore).
- This change was made to ensure consistency for the authentication process.

Changes for Integrators and Developers

Content Object `TEMPLATE`

- The cObject `TEMPLATE`has been removed.
- The following related the PHP class has been removed:
  `TYPO3\CMS\Frontend\ContentObject\TemplateContentObject`.
- The content object was used to render marker-based templates.
- The successor `FLUIDTEMPLATE` acts as a replacement for marker-based templates today.

Changes for Integrators and Developers

Backend URLs

- Backend URLs have changed in TYPO3 v11:
  OLD: `/typo3/index.php?route=%2Frecord%2Fedit&token=...`
  NEW: `/typo3/record/edit?token=...`
- To access the TYPO3 backend, the web server configuration now requires a rewrite rule.
- In previous TYPO3 versions, the backend was explicitly excluded from any URL rewriting logic.
- When accessing the Install Tool, a silent update takes care of adjusting the configuration file for Apache and Microsoft IIS, if possible.

Changes for Integrators and Developers

Backend URLs

- If a manual adjustment of the web server configuration is required (e.g. in customized setups):

  Previous configuration:

  ```
  RewriteRule ^(?:typo3/|fileadmin/|typo3conf/|typo3temp/|uploads/) - [L]
  ```

  New configuration:

  ```
  RewriteRule ^(?:fileadmin/|typo3conf/|typo3temp/|uploads/) - [L]
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !-l
  RewriteCond %{REQUEST_URI} ^/typo3/.*$
  RewriteRule ^typo3/(.*)$ %{ENV:CWD}typo3/index.php [QSA,L]
  ```

- See change log for more details.

Changes for Integrators and Developers

Email Validation

- Developers can use the following method to validate email addresses:
  `TYPO3\CMS\Core\Utility\GeneralUtility::validEmail()`
- Since TYPO3 v11.0, it is possible to configure the validators and to implement custom solutions.
- For example:

```
$GLOBALS['TYPO3_CONF_VARS']['MAIL']['validators'] = [
  \Egulias\EmailValidator\Validation\RFCValidation::class,
  \Egulias\EmailValidator\Validation\DNSCheckValidation::class
];
```

- The following default validators are available:
  - `\Egulias\EmailValidator\Validation\RFCValidation` (used by default)
  - `\Egulias\EmailValidator\Validation\DNSCheckValidation`
  - `\Egulias\EmailValidator\Validation\SpoofCheckValidation`
  - `\Egulias\EmailValidator\Validation\NoRFCWarningsValidation`

Changes for Integrators and Developers

DateTimeAspect

- The DateTimeAspect returns the timestamp as an integer now.
- For example:

```
use TYPO3\CMS\Core\Context\Context;
use TYPO3\CMS\Core\Utility\GeneralUtility;
...
$context = GeneralUtility::makeInstance(Context::class);
$timestamp = $context->getPropertyFromAspect('date', 'timestamp');
```

Changes for Integrators and Developers

Page Language Detection

- Translated page records (based on the fallback handling in the Site Configuration) are now available at a much earlier stage.
- This change was implemented to address a conceptual issue when accessing translated page records.
- As a consequence, TypoScript conditions based on the page record possibly return different results now.
- See change log for further details and migration options.

Changes for Integrators and Developers

Frontend User Groups

- The frontend user groups are now resolved in an earlier state of the frontend request.
- This allows PSR-15 middlewares to properly work with these user groups.
- The functionality has been moved from TSFE into middleware, which decouples the user authentication from the TSFE object.
- Developers of custom authentication services are encouraged to re-test their code in TYPO3 v11.

Changes for Integrators and Developers

JavaScript and CSS Inclusion

- The following methods and properties of the class `TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController` have been removed:
  - `setJS()`
  - `additionalJavaScript`
  - `additionalCSS`
  - `JSCode`
  - `inlineJS`
- A better approach to add JavaScript and CSS snippets is to use the AssetCollector API for example.

Changes for Integrators and Developers

Backend ViewHelpers

- A new optional attribute `defaultValues` was added to the following backend ViewHelpers:
  - `be:uri.newRecord`
  - `be:link.newRecord`
- The argument can be used to prefill fields with default values.
- For example:

```
<be:link.newRecord table="tt_content" pid="17" returnUrl="foo/bar"
  defaultValues="{tt_content: {header: 'value'}}">
  New record
</be:link.newRecord>
```

Changes for Integrators and Developers

Session Storage Wrapper

- A new wrapper class has been introduced to store data temporarily during a session life time: `TYPO3\CMS\Core\Storage\BrowserSession`
- The following API methods are available:
  - `get(key)`
    To fetch the data based on a given key.
  - `set(key, value)`
    To set/override a key with arbitrary content.
  - `isset(key)`
    To check if a key is in use.
  - `unset(key)`
    To remove a key from the storage.
  - `clear()`
    To empty the storage and clear all data.
  - `unsetByPrefix(prefix)`
    To clear all data that have keys that start with a specific prefix.

Changes for Integrators and Developers

"Redirect" Middleware Position

- The "Redirect" middleware has a new default position in the middleware stack.
- TYPO3 now executes `cms-redirects/redirecthandler` **before** `cms-frontend/base-redirect-resolver`

```
...
typo3/cms-redirects/redirecthandler
typo3/cms-frontend/base-redirect-resolver
...
```

- The change of the order was a feature toggle in TYPO3 v10.
- Developers are adviced to check the position of their custom developed middlewares.

Changes for Integrators and Developers

Session Handling

- The session handling in the TYPO3 Core has been significantly restructured.
- It is not part of the user authentication objects anymore, but has been moved into dedicated classes:
  - `TYPO3\CMS\Core\Session\UserSession`
  - `TYPO3\CMS\Core\Session\UserSessionManager`
- This change leads to a centralized handling of sessions.
- The `UserSessionManager` should be used as a factory to create new sessions for various use-cases.
- See TYPO3 documentation for further details.

Changes for Integrators and Developers

Cookies set in the PSR-7 Response

- TYPO3 now attaches cookie information for the user session information in a PSR-15 middleware by default.
- Previously, cookies were added using PHP's native functions `header()` and `setcookie()`.
- The new logic allows developers to attach the cookies to a PSR-7 response object by using `$GLOBALS[BE_USER]->appendCookieToResponse()`.

Changes for Integrators and Developers

PSR-7 Responses in ContentObjects

- PSR-7 standardized request objects have been added to frontend ContentObjects.
- Any kind of custom cObject using PHP code can now access the PSR-7 Request object to fetch information about the current request.
- This method should be used instead of superglobal variables such as `$_GET`, `$_POST`, `$_SERVER`,or TYPO3's API methods `GeneralUtility::_GP()`and `GeneralUtility::getIndpEnv()`.

Changes for Integrators and Developers

PSR-7 Responses in ContentObjects

- Example how to access the PSR-7 Request object in a custom cObject:

TypoScript

```
page.10 = USER
page.10.userFunc = Vendor\FooBar\Frontend\MyClass->test
```

PHP code

```
<?php
namespace Vendor\FooBar\Frontend;
class MyClass
{
  public function test(string $content, array $config, ServerRequestInterface $request)
  {
    $myValue = $request->getQueryParams()[’myGetParameter’];
    $normalizedParams = $request->getAttribute(’normalizedParams’);
    // ...
  }
}
```

Changes for Integrators and Developers

Class PageRenderer

- TYPO3's `PageRenderer`class now always renders the full HTML page, instead of header/footer separately.
- As a consequence, the following constants are now protected:
  - `PageRenderer::PART_COMPLETE`
  - `PageRenderer::PART_HEADER`
  - `PageRenderer::PART_FOOTER`
- `PageRenderer->render()` does not accept any method arguments anymore.
- Extensions should use custom hooks to process or manipulate header or footer parts.

Changes for Integrators and Developers

Shortcut API

- The Shortcut API has been reworked to support Symfony routing to resolve internal endpoints, e.g. backend modules.
- This is in preparation for "speaking URLs" in the TYPO3 backend (BE users will be able to share links to specific functions/pages, for example)
- Developers are adviced to review the following change logs if they use the Shortcut API:
  - 93093: Rework Shortcut PHP API
  - 92132: Deprecate Shortcut PHP API
  - 93060: Shortcut title must be set by controllers
  - 93093: Deprecate method name in Shortcut PHP API

Changes for Integrators and Developers

Form Framework

- Since TYPO3 v10, integrators and developers can use Fluid-based emails in the *EmailFinisher*.
- Therefore, the StandaloneView has now been marked as deprecated, along with the configuration option `templatePathAndFilename`.

Changes for Integrators and Developers

Remote Extension Repositories

- The Extension Manager now uses an **Extension Remotes API**.
- Previously, alternatives to the TYPO3 Extension Repository (TER) were stored in a database table.
- It possible to define new remotes, disable registered remotes, or change the default remote.
- A custom remote configuration can be configured in file `Configuration/Services.yaml` for example:

```
extension.remote.myremote:
  class: 'TYPO3\CMS\Extensionmanager\Remote\TerExtensionRemote'
  arguments:
    $identifier: 'myremote'
    $options:
       remoteBase: 'https://example.com/'
  tags:
    - name: 'extension.remote'
      default: true
```

Changes for Integrators and Developers

SelectCheckBox Groups

- The new TCA setting `expandAll` can be used to expand grouped checkboxes by default. The setting needs to be placed in `[’config’][’appearance’]`
- For example:

```
’select_checkbox’ => [
  ’label’ => ’Select your dinner’,
  ’config’ => [
    ’type’ => ’select’,
    ’renderType’ => ’selectCheckBox’,
    ’appearance’ => [ ’expandAll’ => true ],
    ’items’ => [
      [’group 1’, ’--div--’],
      [’Pizza’, 1],
      [’Pasta’, 2],
      [’group 2’, ’--div--’],
      [’Salad’, 3],
      [’Ice cream’, 4]
    ]
  ]
]
```

Changes for Integrators and Developers

CKEditor Plugin "Placeholder"

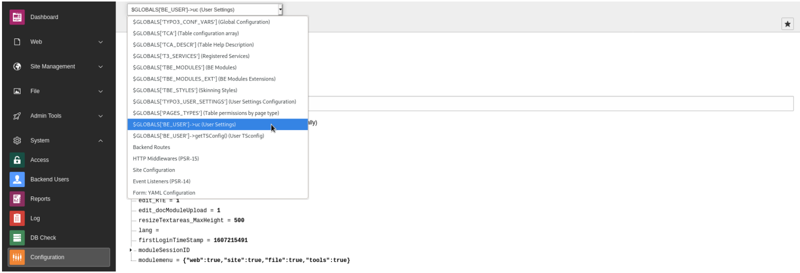
- A new CKEditor plugin is now available: "Editor Placeholder"
- It displays a placeholder text when the editor content is empty
- TCA configuration example:

```
'bodytext' => [
  'label' => 'LLL:EXT:core/Resources/Private/Language/ ...',
  'config' => [
    'placeholder' => 'This is a placeholder',
    'type' => 'text',
    'enableRichtext' => true
  ]
]
```

Changes for Integrators and Developers

Configuration Backend Module (`EXT:lowlevel`)

- The backend module "**Configuration**" can now be customized
- Developers can add their own configurations and disable existing configurations


Dashboard
Web
Site Management
File
Admin Tools
System
Access
Backend Users
Reports
Log
DB Check
Configuration
$GLOBALS['BE_USER']->uc (User Settings)
$GLOBALS['TYPO3_CONF_VARS'] (Global Configuration)
$GLOBALS['TCA'] (Table configuration array)
$GLOBALS['TCA_DESCR'] (Table Help Description)
$GLOBALS['T3_SERVICES'] (Registered Services)
$GLOBALS['TBE_MODULES'] (BE Modules)
$GLOBALS['TBE_MODULES_EXT'] (BE Modules Extensions)
$GLOBALS['TBE_STYLES'] (Skinning Styles)
$GLOBALS['TYPO3_USER_SETTINGS'] (User Settings Configuration)
$GLOBALS['PAGES_TYPES'] (Table permissions by page type)
$GLOBALS['BE_USER']->uc (User Settings)
$GLOBALS['BE_USER']->getTSConfig() (User TSconfig)
Backend Routes
HTTP Middlewares (PSR-15)
Site Configuration
Event Listeners (PSR-14)
Form: YAML Configuration
ally)

```
edit_RTE = 1
edit_docModuleUpload = 1
resizeTextareas_MaxHeight = 500
lang =
firstLoginTimeStamp = 1607215491
moduleSessionID
modulemenu = {"web":true,"site":true,"file":true,"tools":true}
```

Changes for Integrators and Developers

New Datetime-Picker: "flatpickr"

- New datetime picker has been introduced to the TYPO3 backend: "flatpickr" is "lean", lightweight, and doesn't depend on any libraries.
- This change has several implications, most of which affect developers:
  - Initializing all datetime pickers at once triggers a deprecation warning: `DateTimePicker.initialize()`.
  - Option to add/subtract a date or time range in the FormEngine has been removed.

Changes for Integrators and Developers

Recursive Page Deletion

- As recursive page deletion is now always enabled and can't be disabled, the following implications affect integrators and developers:
- The following TSconfig settings have been removed:
  - `setup.default.recursiveDelete`
  - `setup.override.recursiveDelete`
  - `setup.fields.recursiveDelete.disabled`
- On a PHP level, the following property has been removed:
  - `DataHandler->deleteTree`
- The following backend user setting has been removed:
  - `uc['recursiveDelete']`

Changes for Integrators and Developers

TypoScript Option addQueryString.method

- The TypoScript option `addQueryString.method` has been removed.
- This has the following implications:
  - `UriBuilder->setAddQueryStringMethod()` now triggers a deprecation notice.
  - `ContentObjectRenderer->getQueryArguments()` with more than one argument has no effect anymore.
- Additionally, using the argument `addQueryStringMethod` in the following ViewHelpers triggers a deprecation notice:
  - `<f:form>`
  - `<f:link.action>`
  - `<f:link.page>`
  - `<f:link.typolink>`
  - `<f:uri.action>`
  - `<f:uri.page>`
  - `<f:uri.typolink>`

Changes for Integrators and Developers

Console Command List

- Two new properties have been added to the `console.command` dependency injection tag (to be used in the `Services.yaml` file):
  - `description`
  - `hidden`
- The description must be set next to the command name:

```
services:
  My\Namespace\Command\ExampleCommand:
    tags:
      - name: 'console.command'
        command: 'my:example'
        description: 'An example command that demonstrates some stuff'
        hidden: false
```

- Extension authors should use this method instead of setting description through `$this->setDescription()`.

Changes for Integrators and Developers

Console Command List

- The TYPO3 command line utility does not instantiate all available console commands anymore when a user executes `typo3 list`.
- This prevents the CLI tool from slowing down or breaking the command list (e.g. if a command is not instantiable).
- TYPO3 integrators benefit from a faster and more robust command list.

Changes for Integrators and Developers

Backend Routes

- Backend routes can now be restricted to HTTP methods. (e.g. GET, POST, PUT, DELETE)
- The optional property "`methods`" can be set in the files:
  - `Configuration/Backend/Routes.php`
  - `Configuration/Backend/Ajax.php`
- For example:

```
return [
  ’my_route’ => [
    ’path’ => ’/benni/my-route’,
    ’methods’ => [’POST’],
    ’target’ => MyVendor\MyPackage\Controller\MyRouteController::class . ’::submitAction’
  ]
];
```

- No restriction applies if no property is set.

Changes for Integrators and Developers

Content Element Wizard

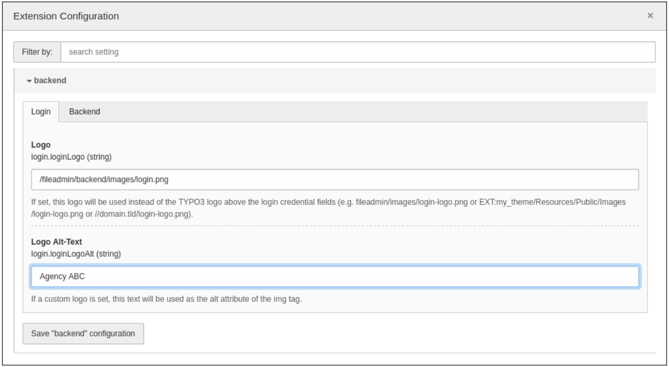
- New "new content element" wizard now supports an icon overlay for each wizard element.
- Useful for custom content elements that use the same `iconIdentifier` multiple times.
- For example (TSconfig):

```
mod.wizards.newContentElement.wizardItems {
  common.elements {
    my_element {
      iconIdentifier = content-my-icon
      iconOverlay = content-my-icon-overlay
      title = LLL:EXT:my_extension/Resources/Private/Language/ContentTypes.xlf:title
      description = LLL:EXT:my_extension/Resources/Private/Language/ContentTypes.xlf:description
      tt_content_defValues {
        CType = my_element
      }
    }
  }
}
```

Changes for Integrators and Developers

Backend Login Image

- The backend login image now supports an `alt`-tag.
- Without a value set, custom images produce a deprecation warning.



Extension Configuration

Filter by: search setting

backend

Login | Backend

Logo
login.loginLogo (string)

/fileadmin/backend/images/login.png

If set, this logo will be used instead of the TYPO3 logo above the login credential fields (e.g. fileadmin/images/login-logo.png or EXT:my_theme/Resources/Public/Images/login-logo.png or //domain.tld/login-logo.png).

Logo Alt-Text
login.loginLogoAlt (string)

Agency ABC

If a custom logo is set, this text will be used as the alt attribute of the img tag.

Changes for Integrators and Developers

FlexForm Data Processor

- A new data processor has been introduced: `TYPO3\CMS\Frontend\DataProcessing\FlexFormProcessor`
- This makes FlexForms available in Fluid templates.
- For example (TypoScript):

```
10 = TYPO3\CMS\Frontend\DataProcessing\FlexFormProcessor
10 {
  fieldName = my_flexform_field
  as = myOutputVariable
}
```

Changes for Integrators and Developers

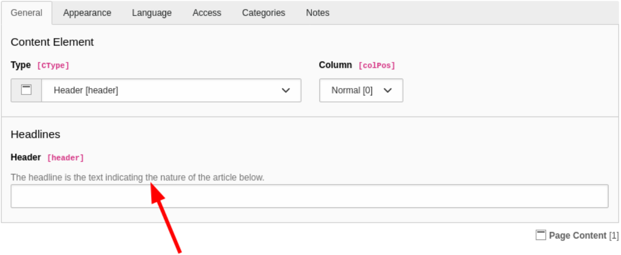
Error Page Caching

- The content of 404 error pages ("page not found") is now cached in the TYPO3 page cache.
- Integrators and administrators are adviced not to output any sensitive information such as the user name of an authenticated frontend user.
- If the 404 error page shows dynamic content, we recommend to implement a custom PHP-based error handler.

Changes for Integrators and Developers

TCA Description

- Since TYPO3 v9, every TCA field can have a description next to its label
  For example:



Content Element

Type [CType]: Header [header]

Column [colPos]: Normal [0]

Headlines

Header [header]

The headline is the text indicating the nature of the article below.

- Integrators can now overwrite the description by using Page TSconfig

Changes for Integrators and Developers

TCA Description

- This option can also be used to set a description for a field that does not feature a TCA description yet
- Example 1 – set/overwrite a description:

```
TCEFORM.<table>.<field>.description = <text>
TCEFORM.tt_content.header.description = The headline is the text...
```

- Example 2 – on a per record type basis:

```
TCEFORM.<table>.<field>.types.<type>.description = Description for <type>
```

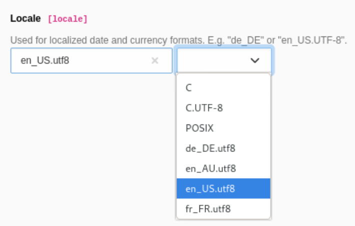
- Example 3 – referring to a language file:

```
TCEFORM.tt_content.header.description = LLL:EXT:my_ext/Resources/Private/
    Language/locallang.xlf:description
```

Changes for Integrators and Developers

System Locales

- System locales are used to localize date and currency formats across the system
- TYPO3 now lists all available locales in the Site Configuration:



Locale [locale]

Used for localized date and currency formats. E.g. "de_DE" or "en_US.UTF-8".

en_US.utf8

- Integrators can easily select the right option for each language

Changes for Integrators and Developers

Disable `hreflang` and/or `canonical` tags

- Under some rare circumstances, integrators want to disable the `hreflang` and/or `canonical` tags used in links:

```
<link rel="alternate" hreflang="de" href="..." />
<link rel="canonical" href="..." />
```

- This is now possible on a page-by-page basis by using TypoScript
- Examples:

```
# Disable hreflang attributes:
config.disableHrefLang = 1

# Disable canonical URLs:
config.disableCanonical = 1
```

Changes for Integrators and Developers

User Group IDs in Conditions

- The following user group IDs are now available in TypoScript:
  - Currently logged-in **backend user**
    (as `backend.user.userGroupIds` in the frontend and backend)
  - Currently logged-in **frontend user**
    (as `frontend.user.userGroupIds` in the frontend)
- This allows integrators to use the Symfony Expression syntax in TypoScript conditions to look-up user groups
- Examples:

```
[4 in frontend.user.userGroupIds]

[2 in backend.user.userGroupIds]
```

Changes for Integrators and Developers

Page and File Tree in the Record Selector

- The same technologies used for the page and file tree are now also implemented in modal windows in the backend:
  - **Record Selector**

    (e.g. selecting a shortcut target or a storage page of a plugin)
  - **File Selector**

    (e.g. selecting a file for an IRRE-based FAL-based file reference)
  - **Link Picker**

    (e.g. selecting a link to a specific page, content element, file, folder, or custom records such as a related news)
- Existing TSconfig options remain functional, for example:
  - `options.pageTree.altElementBrowserMountPoints`
  - `options.folderTree.altElementBrowserMountPoints`

Changes for Integrators and Developers

New TCA type language

- A new TCA field type language has been added
- It displays all available languages for the current context
- The field supersedes the options "foreign_table=sys_language" and "special=languages"
- See changelog for further details and migration notes

< TYPO3 v11.2

```
'config' => [
  'type' => 'select',
  'renderType' => 'selectSingle',
  'foreign_table' => 'sys_language',
  'items' => [ ... ],
  'default' => 0
]
```

>= TYPO3 v11.2

```
'config' => [
  'type' => 'language'
]
```

Changes for Integrators and Developers

PSR-14 Event on Site Configuration Import

- Extensions (site packages and "distributions") can provide site configurations
- These configurations are deployed when the extensions are installed
- A PSR-14 event is now dispatched when the site configuration is imported:

```
\TYPO3\CMS\Extensionmanager\Event\
    AfterExtensionSiteFilesHaveBeenImportedEvent
```

- This enables custom code to access details from the new site configuration

Changes for Integrators and Developers

FAL Folder Object

- The FAL object `TYPO3\CMS\Core\Resource\Folder` now contains a new method `getFile()`
- This function provides a convenient way to retrieve a file
- Example:

  Old approach (still possible)

  ```
  $folder->getStorage()->getFileInFolder("filename.ext", $folder)
  ```

  New method (available since TYPO3 v11.2)

  ```
  $folder->getFile("filename.ext")
  ```

Changes for Integrators and Developers

FormEngine's JavaScript Change Handling

- The JavaScript function `TBE_EDITOR.fieldChanged()` is marked as **deprecated**
- Migration option:

```
// Previous invocation
TBE_EDITOR.fieldChanged('table', 'field_name', 42);

// Migrate to event-based handling
document
  .querySelector('[name="data[table][field_name][42]"]')
  .dispatchEvent(new Event('change', {bubbles: true, cancelable: true}));
```

Changes for Integrators and Developers

Password Reset

- Since TYPO3 v10 LTS, the following global configuration disables the password reset functionality for backend users:

  ```
  $GLOBALS['TYPO3_CONF_VARS']['BE']['passwordReset'] = false
  ```

- Changes in TYPO3 v11.3:
  - The configuration only affects the login screen
  - Administrators can still execute a password reset for users through the backend
  - Administrators can still execute the respective CLI command
- A new User TSconfig option disables the function in the backend:

  ```
  options.passwordReset = 0
  ```

Changes for Integrators and Developers

PSR-3 Logging Messages and Context

- To fully adhere to the PSR-3 specification, the TYPO3 Core now uses placeholders when executing logging messages:

```
$this->logger->alert(
  ’Text text text {placeholder} text text.’,
  [’placeholder’ => $variable]
);
```

- We encourage extension developers to review their code and follow the specification
- **Do not** implement logging this way:

```
$this->logger->alert(’Text text text ’ . $variable . ’ text text.’);
```

Changes for Integrators and Developers

Deactivate Aspect Ratios

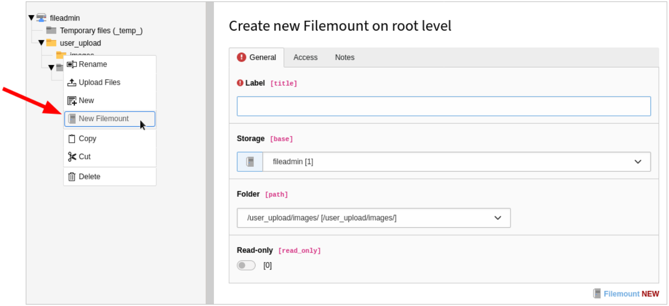
- Setting the key "`disabled`" in the TCA deactivates a specific aspect ratio:

```
$GLOBALS[’TCA’][’tt_content’][’types’][’textmedia’][’columnsOverrides’][’assets’][’config’]
  [’overrideChildTca’][’columns’][’crop’][’config’] = [
  ’cropVariants’ => [
    ’default’ => [
      ’allowedAspectRatios’ => [
        ’4:3’ => [
          ’disabled’ => true
        ]
      ]
    ]
  ]
];
```

Changes for Integrators and Developers

Create Filemount

- TYPO3 administrators can now create filemounts through a new option in the context menu
- For example in the Filelist backend module:



Create new Filemount on root level

Label [title]

Storage [base]
fileadmin [1]

Folder [path]
/user_upload/images/ [/user_upload/images/]

Read-only [read_only]
[0]

Filemount NEW

Changes for Integrators and Developers

Sharing Backend Links

- A new constant `SHAREABLE_URL` is available in the UriBuilder
- Developers can generate deep links (URLs) of pages in the backend using this constant
- For example:

```
$uri = $uriBuilder->buildUriFromRoute(
  $routeName, $arguments, UriBuilder::SHAREABLE_URL
);
```

Changes for Integrators and Developers

List Module: Selectable Columns

- Backend users can now show/hide columns for each table without the need to switch to the single table view
- Integrators can disable the "column selector" for specific tables or for all tables by applying User TSconfig:

```
# disable the column selector for the table "tt_content"
mod.web_list.table.tt_content.displayColumnSelector = 0

# disable the column selector for all tables in general
mod.web_list.displayColumnSelector = 0
```

Changes for Integrators and Developers

XML Sitemap

- A new TypoScript configuration `excludePagesRecursive` has been added to exclude parts of the page tree from the Sitemap
- The option accepts a comma-separated list of page IDs
- For example:

```
plugin.tx_seo {
  config {
    xmlSitemap {
      sitemaps {
        pages {
          config {
            # exclude pages recursively
            excludePagesRecursive = 2,3
          }
        }
      }
    }
  }
}
```

Changes for Integrators and Developers

TCA: t3editor

- A new TCA option `readOnly` has been added to the "t3editor" render type to prevent users from editing the content
- TCA code example:

```
'config' => [
  'type' => 'text',
  'renderType' => 't3editor',
  'format' => 'html',
  'readOnly' => true
]
```

Changes for Integrators and Developers

Symfony Mailer

- Integrators can now configure the Symfony Mailer using DSN:

  ```
  [MAIL][transport] = 'dsn'
  ```

- This makes it possible to use any 3rd-party mail transport provider such as Amazon SES, Gmail, MailChimp, and many more
- Configuration example (Amazon SES):

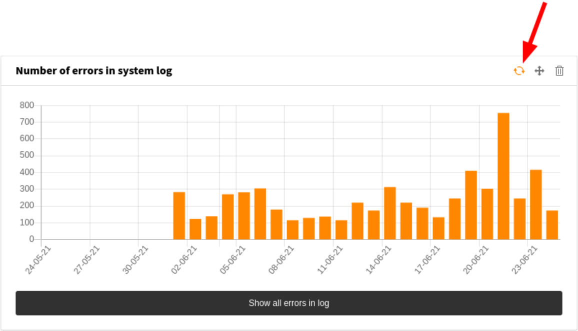
  ```
  [MAIL][dsn] = 'ses+api://ACCESS_KEY:SECRET_KEY@default'
  ```

- Dependencies must be to be installed using Composer
- See Symfony documentation for further details about available providers

Changes for Integrators and Developers

Refresh Dashboard Widgets

- If implemented, backend users can refresh dashboard widgets:



Changes for Integrators and Developers

Refresh Dashboard Widgets

- Set the configuration option `refreshAvailable` to `true`
- Example YAML configuration:

```
dashboard.widget.foobarWidget:
  class: ’TYPO3\CMS\Dashboard\Widgets\RssWidget’
  arguments:
    $view: ’@dashboard.views.widget’
    $cache: ’@cache.dashboard.rss’
    $options:
      rssFile: ’https://example.com/rss’
      lifeTime: 3600
      refreshAvailable: true
  tags:
    ...
```

Changes for Integrators and Developers

FileDumpController

- The `FileDumpController` accepts two new parameters:
  - `dl`: force the download of a file
  - `fn`: download the file with an alternative file name
- See the changelog for further details and a code example
- A new *File*-ViewHelper can be used to leverage the function (see chapter "Extbase and Fluid" for further details)

Changes for Integrators and Developers

PageLayoutController

- The `PageLayoutController` now contains the new method `getModuleTemplate()`
- This gives developers more options to manipulate the page module when using the `drawHeaderHook` and `drawFooterHook`
- The new method can be used to retrieve the corresponding ModuleTemplate instance as the example code on the next page shows

Changes for Integrators and Developers

PageLayoutController

- Example: add a button to the modules’ button bar

```
public function drawHeaderHook(array $parameters, PageLayoutController
     $parentObject)
{
  $moduleTemplate = $parentObject->getModuleTemplate();
  $icon = $moduleTemplate->getIconFactory()->getIcon(
    ’actions-link’, Icon::SIZE_SMALL
  );
  $buttonBar = $moduleTemplate->getDocHeaderComponent()->getButtonBar();
  $linkButton = $buttonBar
    ->makeLinkButton()
    ->setHref(’/typo3/some/url’)
    ->setTitle(’My custom button’)
    ->setClasses(’custom-link-class’)
    ->setIcon($icon);
  $buttonBar->addButton($linkButton);
}
```

Changes for Integrators and Developers

Event Listener

- Event listeners do not require the tag "event" in the file Services.yaml anymore
- TYPO3 automatically detects events based on the type hint in the method signature by using reflection
- The last line in the following example is not required anymore:

```
TYPO3\CMS\Backend\Security\CategoryPermissionsAspect:
  tags:
    - name: event.listener
      identifier: 'backend-user-permissions'
      method: 'addUserPermissionsToCategoryTreeData'
      event: TYPO3\CMS\Core\Tree\Event\ModifyTreeDataEvent
```

Changes for Integrators and Developers

Custom Icon Registration

- Extension developers can now register custom icons through a Service Container
- The new approach improves the performance of requests
- Store the configuration in file `Configuration/Icons.php`:

```
<?php
return [
  ’myicon’ => [
    ’provider’ => \TYPO3\CMS\Core\Imaging\IconProvider\SvgIconProvider::class,
    ’source’ => ’EXT:my_extension/Resources/Public/Icons/myicon.svg’
  ]
];
```

Changes for Integrators and Developers

New TCA Type: `category`

- A new TCA field type "`category`" has been added to the TYPO3 Core
- This supersedes the `CategoryRegistry` and the method `ExtensionManagementUtility->makeCategorizable()` (which we will mark as deprecated in the future)
- TYPO3 automatically generates the TCA configuration and also adds the database field
- Usage example:

```
$GLOBALS[’TCA’][$myTable][’columns’][’categories’] = [
  ’config’ => [
    ’type’ => ’category’
  ]
];

ExtensionManagementUtility::addToAllTCAtypes($myTable, ’categories’);
```

Changes for Integrators and Developers

New TCA Type: `category`

- Developers can define relations, for example:

```
$GLOBALS['TCA'][$myTable]['columns']['mainCategory'] = [
  'config' => [
    'type' => 'category',
    'relationship' => 'oneToOne'
  ]
];
```

- Possible values are:
  - `manyToMany` (default)
  - `oneToMany`
  - `oneToOne`
- See changelog for further details

Changes for Integrators and Developers

Form Framework: `Datepicker.js`

- File `DatePicker.js` has been moved into the directory `Resources/Public/JavaScript/Frontend/` of the Form Framework
- Integrators are now also able to configure an alternative custom publicly accessible directory if required

Changes for Integrators and Developers

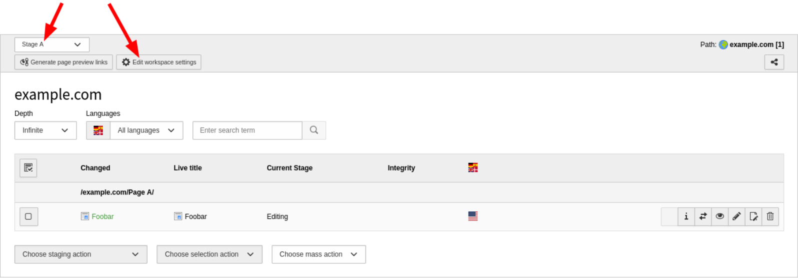
Workspaces

- Several changes have been made to the "Workspaces" functionality:
  - The underlying code received several updates and improvements.
  - Some conceptual issues have been addressed.
  - Several bugs have been fixed.
- Further planned updates are for example:
  - Stabilizing existing functionality.
  - Update the documentation.
  - improve the user experience.
  - Introduce new features.
- Read more about the initiative in the article:
  "Workspaces – One of TYPO3's Most Underused Features".

Changes for Integrators and Developers

Workspaces Module

- The UI of the Workspaces module has been improved
  (e.g. a dropdown menu can be used to switch between workspaces)
- Administrators can now edit the Workspace settings directly
  (access through the List module is not required anymore)



Changed	Live title	Current Stage	Integrity
/example.com/Page A/			
Foobar	Foobar	Editing	

Changes for Integrators and Developers

Database Identifier Quotation

- TCA keys that contain SQL fragments use a new quoting mechanism when the TYPO3 Core builds the database queries (fields such as `foreign_table_where`, `MM_table_where`, etc.)
- This is important to ensure DBAL compatibility and makes a database connection unnecessary during TCA cache warmup
- A feature flag "`runtimeDbQuotingOfTcaConfiguration`" lets integrators/developers force the old behaviour if required
- This flag will be removed in TYPO3 v12.0

Changes for Integrators and Developers

Important Fields

- The following two fields are now always enabled/shown in the TCA:
  - `tt_content.colPos`
  - `pages.slug`
- This prevents integrators from accidentally misconfigure the access permissions for backend users
- The following TSconfig overwrites the default settings and hides these fields if required:

```
TCEFORM.pages.slug.disabled = 1
TCEFORM.tt_content.colPos.disabled = 1
```

Changes for Integrators and Developers

New Option: `showNewRecordLink`

- Developers can now disable the "New record" link of TCA inline elements independently from the "+" button and all other "level links"
- A new option `showNewRecordLink` has been introduced for this
- For example (to disable the "New record" button):

```
’inlineField’ => [
  ’label’ => ’Inline without New record link’,
  ’config’ => [
    ’type’ => ’inline’,
    ’appearance’ => [
      ’showNewRecordLink’ => false
    ]
  ]
]
```

- The following option only manages the display of the "+" button now:

```
[’config’][’appearance’][’enabledControls’][’new’]
```

Changes for Integrators and Developers

SoftReference Parsers

- Developers can now register user-defined soft reference parsers
- For example (in file `Configuration/Services.yaml`):

```
Vendor\MyExtension\SoftReference\MySoftReferenceParser:
  tags:
    - name: softreference.parser
      parserKey: your_key
```

- Parsers have to implement the following new interface:

```
TYPO3\CMS\Core\DataHandling\SoftReference\SoftReferenceParserInterface
```

- The old registration method has been marked as **deprecated**:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['GLOBAL']['softRefParser']
```

Changes for Integrators and Developers

Indexed Search

- The system extension `EXT:indexed_search` now features a new configuration option: "`basic.deleteFromIndexAfterEditing`"
- If enabled (which is the default) and a page or its content is updated, the page and its content are removed from the search index
- If this option is disabled, backend users have to manually clear the index as required

Changes for Integrators and Developers

Icons in Reports Module

- Developers can now provide icon identifiers in the registration array of the Reports backend module
- For example:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['reports']['tx_reports']['status'] = [
  'title' =>
    'LLL:EXT:reports/Resources/Private/Language/locallang_reports.xlf:status_report_title',
  'icon' =>
    'module-reports', // icon identifiers are now possible
  'description' =>
    'LLL:EXT:reports/Resources/Private/Language/locallang_reports.xlf:status_report_description',
  'report' =>
    \TYPO3\CMS\Reports\Report\Status\Status::class
];
```

Changes for Integrators and Developers

TCA Configuration: `fileFolder`

- The TCA configuration "`fileFolder`" has been streamlined
- Options are now available as "`fileFolderConfig`":

TYPO3 < v11.4

```
’fieldname’ => [
  ’config’ => [
    ’type’ => ’select’,
    ’renderType’ => ’selectSingle’,
    ’fileFolder’ => ’EXT:myext/Resources/Public/Icons’,
    ’fileFolder_extList’ => ’svg’,
    ’fileFolder_recursions’ => 1
  ]
]
```

TYPO3 >= v11.4

```
’fieldname’ => [
  ’config’ => [
    ’type’ => ’select’,
    ’renderType’ => ’selectSingle’,
    ’fileFolderConfig’ => [
      ’folder’ => ’EXT:myext/Resources/Public/Icons’,
      ’allowedExtensions’ => ’svg’,
      ’depth’ => 1
    ]
  ]
]
```

- Note the sub-array "`fileFolderConfig`" and the new option names (e.g. "`depth`") in TYPO3 >= v11.4

Changes for Integrators and Developers

TCA Configuration: `fileFolderConfig`

- Integrators can overwrite the new TCA configuration in TSconfig
- For example:

```
config.fileFolderConfig.folder
config.fileFolderConfig.allowedExtensions
config.fileFolderConfig.depth
```

- Read more:
  - TCA configuration "FileFolder" (TYPO3 documentation)
  - Further details about the change in the changelog

Changes for Integrators and Developers

CKEditor Plugin "Autolinking"

- URLs such as `example.com` typed in the RTE are automatically converted into an external link
- This is done by CKEditor’s plugin "AutoLink"
- Since TYPO3 v11.4, the default schema of these links is: **https**
- Editors can, of course, manually update the schema to http if required

Changes for Integrators and Developers

Doctrine DBAL: New Method `notInSet()`

- TYPO3's DBAL ExpressionBuilder now also supports the method `notInSet()`
- This works with all officially supported DB engines such as SQLite, MySQL/MariaDB, PostgreSQL, and MSSQL
- The following example queries all frontend users who do not directly belong to the usergroup ID 5:

```
$queryBuilder = GeneralUtility::makeInstance(ConnectionPool::class)
  ->getQueryBuilderForTable('fe_users');
$result = $queryBuilder
  ->select('*')
  ->from('fe_users')
  ->where($queryBuilder->expr()->notInSet('usergroup', '5'))
  ->execute();
```

Changes for Integrators and Developers

The `HtmlParser` Class

- The HTML Parser now supports the attribute "`srcset`"
- This is particularly important for responsive images
- Example 1:

```
<picture>
  <source media="(max-width: 799px)" srcset="small-image.jpg">
  <source media="(min-width: 800px)" srcset="large-image.jpg">
</picture>
```

- Example 2:

```
<img srcset="small-image.jpg 640w, large-image.jpg 1024w"
  sizes="(max-width: 799px) 640px, 1024px"
  src="large-image.jpg" />
```

Changes for Integrators and Developers

Global Namespace

- When TYPO3 warms up caches, the code of the following files are now scoped into the global namespace:

  `ext_localconf.php` and `ext_table.php`
- This means that extension developers **must not** wrap the content of these files in a local namespace anymore
- Extension authors can now import classes by the `use`-statement:

```
<?php
use TYPO3\CMS\Core\Utility\ExtensionManagementUtility;
ExtensionManagementUtility::addUserTSConfig(’
  options.saveDocView = 1
  options.saveDocNew = 1
  options.saveDocNew.pages = 0
  options.saveDocNew.sys_file = 0
  options.saveDocNew.sys_file_metadata = 0
  options.disableDelete.sys_file = 1
’);
```

Changes for Integrators and Developers

TCA: foreign_table_where

- The `foreign_table_where` setting in the TCA now also supports the following markers:
  - `###SITE:<KEY>###`
    (e.g. `###SITE:rootPageId###`)
  - `###SITE:<KEY>.<SUBKEY>###`
    (e.g. `###SITE:mySetting.categoryPid###`)
- This enables developers to access settings from the site configuration

Changes for Integrators and Developers

TYPO3 CLI

- New command added to TYPO3's command line interface (CLI):

  ```
  ./bin/typo3 impexp:export [options] [--] [<filename>]
  ```

- The "export" completes the existing "import" functionality
- The command exports the entire TYPO3 page tree or parts of it
- Integrators and administrators can influence the export through a wide range of options (e.g. file type, page tree depth, etc.)
- Exports can be re-imported into other TYPO3 instances or used as initial page trees in a distribution

Changes for Integrators and Developers

TYPO3 CLI

- New command added to TYPO3's command line interface (CLI):

  ```
  ./bin/typo3 cache:warmup [--group <all|system|di|pages|...>]
  ```

- Integrators can now warm up all caches (default) or selectively through the available cache groups
- Specific cache groups can be defined with the `--group` option: `system`, `pages`, `di`, or `all` (default: `all`)
- Note: not *all* caches can be warmed up (yet)
  (e.g. frontend cache, RequireJS assets cache, Extbase caches, Fluid templates are currently excluded)

Changes for Integrators and Developers

TYPO3 CLI

- New command added to TYPO3's command line interface (CLI):

  ```
  ./bin/typo3 cache:flush [--group <all|system|di|pages|...>]
  ```

- Integrators can now flush all caches (default) or selectively through the available cache groups
- Specific cache groups can be defined with the `--group` option: `system`, `pages`, `di`, or `all` (default: `all`)

Changes for Integrators and Developers

Composer-based Installations

- The file "`PackageStates.php`" has become obsolete in Composer-based installations
- All extensions added to the system by Composer are now considered to be active
- Therefore, the Extension Manager and the TYPO3 CLI don't offer the option to activate/deactivate extensions anymore
- Composer-based TYPO3 installations don't need the file "`ext_emconf.php`" in extensions anymore
- Developers may want to keep it for the time being though to retain compatibility with TYPO3 instances in non-Composer mode
- See changelog for further details

Changes for Integrators and Developers

Composer-based Installations

- TYPO3 extensions are Composer packages and should be installed using Composer. Do not store extensions in the `typo3conf/ext/` directory, if they haven't been installed using Composer
- Migration: consider to move them into a separate directory and add the directory as a *repository*, for example:

```
...
"repositories": [{
  "type": "path",
  "url": "./packages/*/"
}],
...
```

- See changelog for further details

Changes for Integrators and Developers

Performance Optimization

- To optimize the performance even further, TYPO3 becomes stricter in terms of booting up the core configuration
- Accessing the database or utilizing the Cache Manager in the following files triggers a deprecation error:
  - `ext_localconf.php`
  - `ext_tables.php`
  - `Configuration/TCA/*`
- Extension authors should make sure that their extensions use proper initialization in the right places

Changes for Integrators and Developers

PSR-14 Events

- The following PSR-14 events have been added to the TYPO3 Core:
    - `ProcessFileListActionsEvent`
      (event fires after the actions in the Filelist module are generated)
    - `BootCompletedEvent`
      (event fires on every request when TYPO3 is fully booted)
    - `ModifyFileDumpEvent`
      (event fires before dumping a file)
    - `ModifyClearCacheActionsEvent`
      (event allows for modifying "cache"-items in the backend top toolbar)
    - `AfterFileCommandProcessedEvent`
      (event allows for executing additional tasks, after a file operation has been performed)

Changes for Integrators and Developers

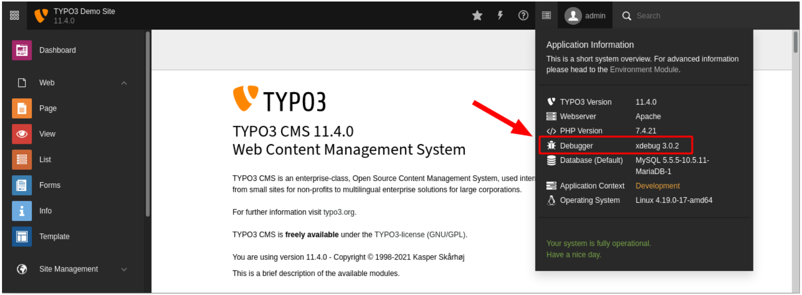
PSR-14 Events

- The following PSR-14 events have been added to the TYPO3 Core:
  - `ModifyRecordListTableActionsEvent`
  - `ModifyRecordListHeaderColumnsEvent`
  - `ModifyRecordListRecordActionsEvent`
- These events can be used to modify the behaviour of table listings
- They can be used to either add, change, or remove columns and actions
- See changelog for further details

Changes for Integrators and Developers

Application Information

If an active debugger can be determined by the PHP function `extension_loaded()`, the debugger is listed in the Application Information.



TYPO3

TYPO3 CMS 11.4.0
Web Content Management System

TYPO3 CMS is an enterprise-class, Open Source Content Management System, used inter
from small sites for non-profits to multilingual enterprise solutions for large corporations.

For further information visit typo3.org.

TYPO3 CMS is **freely available** under the TYPO3-license (GNU/GPL).

You are using version 11.4.0 - Copyright © 1998-2021 Kasper Skårhøj

This is a brief description of the available modules.

Application Information

This is a short system overview. For advanced information please head to the Environment Module.

TYPO3 Version	11.4.0
Webserver	Apache
PHP Version	7.4.21
Debugger	xdebug 3.0.2
Database (Default)	MySQL 5.5.5-10.5.11-MariaDB-1
Application Context	Development
Operating System	Linux 4.19.0-17-amd64

Your system is fully operational.
Have a nice day.

Changes for Integrators and Developers

MM-tables

- TYPO3's schema analyzer detects "MM" database tables and automatically creates them based on the TCA definition
- A table column TCA config defines MM tables by: `type='select'`, `type='group'`, or `type='inline'`
- Extension developers are encouraged to **remove** the `CREATE TABLE` statements from the `ext_tables.sql` files of "MM" tables
- The schema analyzer takes care of required fields, in addition to `uid_local` and `uid_foreign`

Changes for Integrators and Developers

LoggerInterface Injection

- Services are no longer required to use

  ```
  Psr\Log\LoggerAwareInterface
  Psr\Log\LoggerAwaireTrait
  ```

- Developers can add a constructor argument

  ```
  Psr\Log\LoggerInterface
  ```

- For example:

  ```
  use Psr\Log\LoggerInterface;
  class MyClass {
    private LoggerInterface $logger;
    public function __construct(LoggerInterface $logger) {
      $this->logger = $logger;
    }
  }
  ```

Changes for Integrators and Developers

Attribute "Channel" for Logger Injection (PHP v8)

- TYPO3 v11.4 introduces a PHP v8 specific feature: *attributes*
- This PHP feature allows developers to add meta data to classes, methods, variables, etc.
- Developers can now control the component name that an injected Logger is created with
- For example (note the line: `#[Channel('security')]`):

```
use Psr\Log\LoggerInterface;
use TYPO3\CMS\Core\Log\Channel;
#[Channel('security')]
class MyClass
{
  private LoggerInterface $logger;
  public function __construct(LoggerInterface $logger)
  {
    $this->logger = $logger;
  }
}
```

Changes for Integrators and Developers

TCA Option treeConfig

- A new setting startingPoints has been introduced to the treeConfig TCA configuration
- The setting startingPoints lets developers set multiple records as roots for tree records, based on a CSV list of page IDs
- For example:

```
// Example TCA config
'config' => [
  'treeConfig' => [
    'startingPoints' => '1,2,3',
  ]
]
```

- Developers can also access values from the site configuration by using the syntax: "###SITE:...###"

Changes for Integrators and Developers

New Class `LinkResult`

- A new class has been added to the TYPO3 Core:
  `TYPO3\CMS\Frontend\Typolink\LinkResult`
- This class implements the `LinkResultInterface`
- It represents a created link to a resource (page, email, etc.)
- The object contains the base result of a link generated by TypoLink and allows developers to handle link data in a more consistent way
- It also simplifies typolink handling in, for example, JSON output

Changes for Integrators and Developers

TCA Option `groupName`

- The TYPO3 Core now makes use of the TCA option "`groupName`"
- This option allows to group records in the "new record wizard"
- The following keywords are used by the Core:
  `system`, `frontendaccess`, `backendaccess`, and `content`
- Extensions that ship their own groups and keywords continue to work

Changes for Integrators and Developers

TCA Option `internal_type`

- When the TCA type "`group`" is used, the option "`internal_type`" defines which type of record can be referenced (valid values are "`folder`" and "`db`")
- Since "`db`" is commonly used, TYPO3 now uses this value as the default
- This also means that extension developers can omit the option "`internal_type`" in most cases

Changes for Integrators and Developers

New PSR-14 Event

- The following new PSR-14 event has been introduced:
  `ModifyResolvedFrontendGroupsEvent`
- This event allows developers to attach frontend user groups dynamically during a frontend request

Changes for Integrators and Developers

Bootstrap 5 Compatible Templates

- Bootstrap version 5 compatible templates have been added to the Form Framework
- These template variants are stored in `EXT:form/Resources/Private/Frontend/Version2/`
- In many cases, only the appropriate CSS classes need to be defined to customize the look and feel of the templates
- However, TYPO3 continues to render the old variants by default to prevent an existing setup to break
- The new templates need to be explicitly enabled
- See changelog for further details

Extbase and Fluid

The Extbase framework and the Fluid templating engine

Extbase and Fluid

Plugin/Module Registration

- Registering plugins/modules require fully-qualified class names and no vendor name as the extension name (first argument).
- This was marked as deprecated in TYPO3 v10 and is now mandatory.
- This applies to `ExtensionUtility::configurePlugin()`and `ExtensionUtility::registerModule()`.
- For example:

```
\TYPO3\CMS\Extbase\Utility\ExtensionUtility::configurePlugin(
  'Form',
  'Formframework',
  [
    \TYPO3\CMS\Form\Controller\FormFrontendController::class => 'render, perform'
  ],
  [
    \TYPO3\CMS\Form\Controller\FormFrontendController::class => 'perform'
  ],
  \TYPO3\CMS\Extbase\Utility\ExtensionUtility::PLUGIN_TYPE_CONTENT_ELEMENT
);
```

Extbase and Fluid

PSR-7 Responses

- Extbase now creates a PSR-7-compatible response object and passes it back through the request handling stack.
- The following interfaces/classes are not used anymore and were removed:
  - `TYPO3\CMS\Extbase\Mvc\ResponseInterface`
  - `TYPO3\CMS\Extbase\Mvc\Response`
- If a controller action should return custom data instead of the default view, a PSR-7-compatible response object can be returned:

```
public function listAction()
{
  // do your action stuff
  return new \TYPO3\CMS\Core\Http\HtmlResponse($this->view->render());
}
```

Extbase and Fluid

PSR-7 `ForwardResponse`

- The following class has been introduced to forward a request to another controller action: `TYPO3\CMS\Extbase\Http\ForwardResponse`
- This is a PSR-7 compatible way to return a response object.
- The old method has been marked as deprecated:

  `ActionController::forward()`

Extbase and Fluid

PSR-7 `ForwardResponse`

- Code example:

```
<?php

use Psr\Http\Message\ResponseInterface;
use TYPO3\CMS\Extbase\Mvc\Controller\ActionController;
use TYPO3\CMS\Extbase\Http\ForwardResponse;

class FooController extends ActionController
{
  public function listAction(): ResponseInterface
  {
    // do something
    return new ForwardResponse(’show’);
  }

  public function showAction(): ResponseInterface
  {
    // do something
  }
}
```

Extbase and Fluid

PSR-7 ResponseInterface

- The aim to fully comply with PSR-7 affects the data types that controller actions may return.
- Return values can be either nothing (`void`), `null`, a string, or an object.
- Actions should now return an instance of `Psr\Http\Message\ResponseInterface`.
- Everything else triggers a warning in the deprecation log.
- Code example (after migration):

```
public function listAction(): ResponseInterface
{
  $items = $this->itemRepository->findAll();
  $this->view->assign('items', $items);
  return $this->htmlResponse();
}
```

Extbase and Fluid

PSR-7 Request/Response

- To follow unified standards and state-of-the-art technologies for extension developers, Extbase should feature a clean request/response handling in accordance to the PSR-7 standard
- Therefore, the class `TYPO3\CMS\Extbase\Mvc\Request` now implements the PSR-7 `ServerRequestInterface`
- This lets developers access request-related information from the core
- The interface is, for example, available within Extbase controllers: `$this->request`
- See the changelog for further details

Extbase and Fluid

PSR-7 Request/Response

- As part of this change, the following class has been streamlined: `TYPO3\CMS\Extbase\Mvc\Request`
- Methods marked as **deprecated**:
  - `getBaseUri()`
    (possible migration: `$baseUri = $normalizedParams->getSiteUrl()`)
  - `getRequestUri()`
    (possible migration: `$requestUrl = $normalizedParams->getRequestUrl()`)
  - `setDispatched()` and `isDispatched()`
    (see changelog for migration options)

Extbase and Fluid

PSR-7 Request/Response

- Since the method `setMethod()` has been removed, the following exception is no longer available:

  ```
  \TYPO3\CMS\Extbase\Mvc\Exception\InvalidRequestMethodException
  ```

- Also, the following class has been marked as **deprecated**:

  ```
  \TYPO3\CMS\Extbase\Mvc\Web\ReferringRequest
  ```

Extbase and Fluid

PSR-7 Request/Response

- Extbase actions **must** return a `ResponseInterface` in the future
- Extensions are affected if:
  - They throw a `StopActionException`
  - They use the method `redirect()`
  - They use the method `redirectToUri()`
- Migration example:

```
// Code before:
public function fooAction()
{
   $this->redirect('otherAction');
}

// Code after:
public function fooAction(): ResponseInterface
{
   return $this->redirect('otherAction');
}
```

Extbase and Fluid

RegularExpression Validator

- Custom error messages can be returned by using the new option `errorMessage` of the RegularExpressionValidator:

```
class MyModel extends \TYPO3\CMS\Extbase\DomainObject\AbstractEntity
{
  /**
   * @var string
   * @TYPO3\CMS\Extbase\Annotation\Validate(
   *   "RegularExpression",
   *   options={
   *     "regularExpression": ’/^SO[0-9]$/’,
   *     "errorMessage": ’LLL:EXT:my_ext/path/to/xlf:translation.key’
   *   }
   * )
   */
  protected $customField;
}
```

Extbase and Fluid

ActionController

- Several properties and methods of the class `TYPO3\CMS\Extbase\Mvc\Controller\ActionController` have been marked as internal.
- Although these properties and methods remain operational, developers are adviced to review if custom developed extension are affected.
- Internal properties and methods will possibly be removed without deprecation notification in TYPO3 v12 or later.
- See change log for a list of affected properties and methods.

Extbase and Fluid

Dependency Injection

- Property Dependency Injection by using `@Extbase\Inject` has been marked as deprecated.
- Migration options:
  - **Constructor Injection**
    Allows for extension compatibility with multiple TYPO3 versions.
  - **Setter Injection**
    Similar pros and cons as constructor dependency injection.
  - **Core Property Injection**
    Requires configuration of services in the file `Services.yaml`.

Extbase and Fluid

Dependency Injection

- The previously deprecated option to use dependency injection for non-public properties has been removed.
- The code example below shows a working solution (note the `public` visibility!)

```
class Foo
{
  /**
   * @var Service
   * @TYPO3\CMS\Extbase\Annotation\Inject
   */
  public $service;
}
```

Extbase and Fluid

Custom Error Messages for Validation Results

- A new method `addErrorForProperty()` allows developers to add custom error messages
- This applies to properties of Extbase models
- Example in a validator class:

```
public function isValid(): void
{
  // validation
  $this->addErrorForProperty(
    'object.property.name',
    $this->translateErrorMessage(
      'validator.errormessage',
      'myextension'
    ),
    1626134400 // timestamp
  );
}
```

Extbase and Fluid

Parameter Type Evaluation

- Extbase's *legacy* type detection of arguments has been marked as **deprecated**
- Developers should use PHP's type hints in argument lists
- Example (**old** code):

```
/**
 * @param \MyVendor\MyExtension\MyModel $item
 */
public function myAction($item);
```

new code:

```
public function myAction(\MyVendor\MyExtension\MyModel $item);
```

Extbase and Fluid

Recursive Property Transformation

- Developers can now use the property "`_recursive`" in the JsonView configuration to resolve recursive properties
- This simplifies the implementation of, for example, directories containing directories or comments containing comments as replies, etc.
- Examples:
  - 1:n relations

```
$configuration = [
  ’directories’ => [
    ’_descendAll’ => [
      ’_recursive’ => [’directories’]
    ]
  ]
];
```

  - 1:1 relations

```
$configuration = [
  ’comment’ => [
    ’_recursive’ => [’comment’]
  ]
];
```

Extbase and Fluid

Extbase ObjectManager

- The TYPO3 v11 Core does not use the Extbase ObjectManager anymore
- All related components have been marked as **deprecated**
- For example, calling `ObjectManager->get()` now writes a deprecation warning to the log
- Developers are encouraged to use Symfony's DependencyInjection component since TYPO3 v10
- The ObjectManager and all related components will be removed in TYPO3 v12
- See changelog for migration options

Extbase and Fluid

Generic Extbase Domain Classes

- The following *generic* Extbase domain models and repositories have been marked as **deprecated**:
  - `TYPO3\CMS\Extbase\Domain\Model\BackendUser`
  - `TYPO3\CMS\Extbase\Domain\Model\BackendUserGroup`
  - `TYPO3\CMS\Extbase\Domain\Model\FrontendUser`
  - `TYPO3\CMS\Extbase\Domain\Model\FrontendUserGroup`

  - `TYPO3\CMS\Extbase\Domain\Repository\BackendUserRepository`
  - `TYPO3\CMS\Extbase\Domain\Repository\BackendUserGroupRepository`
  - `TYPO3\CMS\Extbase\Domain\Repository\FrontendUserRepository`
  - `TYPO3\CMS\Extbase\Domain\Repository\FrontendUserGroupRepository`
  - `TYPO3\CMS\Extbase\Domain\Repository\CategoryRepository`
- See changelog for migration options

Extbase and Fluid

Fluid StandaloneView and ContentObjectRenderer

- Extensions that create an instance of the StandaloneView should not pass an instance of the ContentObjectRenderer as a constructor argument
- Otherwise, TYPO3 generates a deprecation error

Extbase and Fluid

Collection Types of Non-persistent Objects

- Collection types are used to define a specific class that should be within Extbase’s ObjectStorage class, for example:

```
/**
 * @param ObjectStorage<Item> $items
 */
public function setItems(ObjectStorage $items): void
{
  $this->items = $items;
}
```

- These docblocks are analyzed so that the PropertyMapper knows how to map incoming requests
- Developers can now use collection types in docblock annotations also for non-persistent objects

Extbase and Fluid

Deprecated PHP Classes

- The following interface has been marked as **deprecated**:
  `TYPO3\CMS\Extbase\Mvc\View\ViewInterface`
- To further decouple Fluid from Extbase the following class has been marked as deprecated:
  `TYPO3\CMS\Extbase\Mvc\Controller\ControllerContext`

Fluid View Clean-up

- Clean-up work has been done to simplify and streamline the Fluid view
  - The following classes have been marked as **deprecated**:
    `TYPO3\CMS\Extbase\Mvc\View\EmptyView`
    `TYPO3\CMS\Extbase\Mvc\View\AbstractView`
  - The method `canRender()` has been removed from the following interface:
    `TYPO3\CMS\Extbase\Mvc\View\ViewInterface`

Extbase and Fluid

Fluid Widgets

- Fluid "widgets" and all widget functionality have been removed.
- The technology violates the design pattern "separation of concern" and caused problems in regards to using PSR-7 request objects in Extbase.
- Impact of this change:
  - Fluid templates that use an existing widget will no longer work.
  - Custom widgets will no longer work and have to be replaced.
- Developers can use the Pagination API instead of `<f:paginate...>` for example.
- See change log for further details.

Extbase and Fluid

TypolinkViewhelper

- The ViewHelper `<f:link.typolink ...>` features a new argument `textWrap` which wraps the link title.
- For example:

```
<f:link.typolink parameter="123" textWrap="<span>|</span>" />
```

- Output:

```
<a href="page-123"><span>My page title</span></a>
```

- Note: don't forget to escape quotes:

```
<f:link.typolink parameter="123" textWrap="<span class=\"small\">|</span>" />
```

Extbase and Fluid

New *File*-ViewHelper

- A new ViewHelper has been implemented:

  ```
  TYPO3\CMS\Fluid\ViewHelpers\Link\FileViewHelper
  ```

- The ViewHelper lets extension authors create links to both public and non-public files easily
- For example:

  ```
  <f:link.file file="{file}" download="true"
    filename="alternative-name.jpg">Download</f:link.file>
  ```

- The argument `file` accepts a `File` object, a `FileReference`, or a `ProcessedFile`
- An alternative download file name can be specified using the argument `filename`

Extbase and Fluid

Hidden-ViewHelper

- A new argument `respectSubmittedDataValue` has been added to the *Hidden*-ViewHelper
- This argument controls the usage of previously submitted values for the respective fields
- Typical use cases are sub requests, when Extbase dispatches a FowardResponse
- Example:

```
<f:form.hidden
  property="hiddenProperty"
  value="{form.hiddenProperty}"
  respectSubmittedDataValue="false" />
```

Extbase and Fluid

Fluid ViewHelpers

- The Fluid ViewHelpers "`f:uri.external`" and "`f:link.external`" now use **https** as the default scheme
- For example:

  ```
  <f:link.external uri="example.com">
    external link
  </f:link.external>
  ```

  This generates the link:

  ```
  <a href="https://example.com">external link</a>
  ```

- To overwrite the default value, developers can explicitly set the scheme as follows (e.g. "http"):

  ```
  <f:link.external uri="example.com" defaultScheme="http">
    external link
  </f:link.external>
  ```

Extbase and Fluid

Fluid ViewHelpers

- The Fluid ViewHelper "`f:form.password`" now also supports the attribute "`autocomplete`"
- This attribute can be used, for example, to instruct password managers how to treat the password input field
- For example:

```
<f:form.password name="newPassword" value="" autocomplete="new-password" />
```

- Typical values are:
  "`on`", "`off`", "`current-password`", "`new-password`"
- See MDN specification for further details about possible values

Extbase and Fluid

Fluid ViewHelpers

- All Fluid ViewHelper classes will be declared "`final`" in TYPO3 v12
- Third-party extensions can no longer extend them with own variants
- Extension developers are advised to use the abstract classes provided by Fluid (if they are not marked `internal`)
- Developers are encouraged to make appropriate adjustments to their custom ViewHelpers in TYPO3 v11 already

Extbase and Fluid

Fluid ViewHelpers

- A new ViewHelper has been introduced:
  `<f:transform.html>`
- The ViewHelper resolves and transforms individual TYPO3 link handlers such as `t3://`-URIs
- It is independent from any TypoScript configuration and works without mixing concerns in `ContentObjectRenderer::parseFunc`

```
<f:transform.html selector="a.href,div.data-uri">
  <a href="t3://page?uid=1" class="page">visit</a>
  <div data-uri="t3://page?uid=1" class="page trigger">visit</div>
</f:transform.html>
```

Extbase and Fluid

Fluid ViewHelpers

- The following Fluid ViewHelpers have been marked as **deprecated**:
  - `<f:uri.email email="{email}">`

Extbase and Fluid

Miscellaneous

- The method `canProcessRequest()` of the following class has been removed: `TYPO3\CMS\Extbase\Mvc\Controller\ActionController`

Deprecated/Removed Functions

The TYPO3 Core cleaned up and ready for the future

Deprecated/Removed Functions

Database-related Changes

- The following two database fields were removed from various tables:
  - `t3ver_tstamp`
  - `t3ver_count`
- The following database table is not used anymore and was removed:
  - `tx_extensionmanager_domain_model_repository`

Deprecated/Removed Functions

HTML tags `<style>` and `<link>`

- The HTML tags `<style>` and `<link>` do not use the attribute `type` by default anymore.
- This change applies to the TYPO3 backend and frontend output.
- If required, the attribute can be re-added by using TypoScript options or by Fluid AssetCollector attributes.

Deprecated/Removed Functions

Constants `TYPO3_MODE` and `TYPO3_REQUESTTYPE`

- The following global constants have been marked as deprecated:
  - `TYPO3_MODE`
  - `TYPO3_REQUESTTYPE`
  - `TYPO3_REQUESTTYPE_FE`
  - `TYPO3_REQUESTTYPE_BE`
  - `TYPO3_REQUESTTYPE_CLI`
  - `TYPO3_REQUESTTYPE_AJAX`
  - `TYPO3_REQUESTTYPE_INSTALL`
- Extension developers are highly encouraged to drop the usage of these constants and prepare their extensions for TYPO3 v11.
- See change log for migration options.

Deprecated/Removed Functions

Session Handling

- Requesting the frontend by using the following GET parameter does not have an effect anymore: `FE_SESSION_KEY`
  (an existing session will not be picked up)

Deprecated/Removed Functions

Backend Shortcuts and Bookmarks

- The following methods related to the shortcut/bookmark handling in the TYPO3 backend have been marked as deprecated:
  - `TYPO3\CMS\Backend\Template\ModuleTemplate->makeShortcutIcon()`
  - `TYPO3\CMS\Backend\Template\ModuleTemplate->makeShortcutUrl()`
  - `TYPO3\CMS\Backend\Template\Components\Buttons\Action\ShortcutButton->getSetVariables()`
  - `TYPO3\CMS\Backend\Template\Components\Buttons\Action\ShortcutButton->getGetVariables()`
  - `TYPO3\CMS\Backend\Template\Components\Buttons\Action\ShortcutButton->setGetVariables()`
  - `TYPO3\CMS\Backend\Template\Components\Buttons\Action\ShortcutButton->setSetVariables()`
  - `TYPO3\CMS\Backend\Template\Components\Buttons\Action\ShortcutButton->setModuleName()`
  - `TYPO3\CMS\Backend\Template\Components\Buttons\Action\ShortcutButton->getModuleName()`
- The following Fluid ViewHelper has been marked as deprecated:
  - `f:be.buttons.shortcut`
- Additionally, the argument `getVars` of the following ViewHelper has been marked as deprecated:
  - `be:moduleLayout.button.shortcutButton`

Deprecated/Removed Functions

Class `AbstractUserAuthentication`

- The following changes are related to the class `TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`
- Removed public properties and methods:
  - `AbstractUserAuthentication->auth_timeout_field`
  - `AbstractUserAuthentication->loginFailure`
  - `AbstractUserAuthentication->forceSetCookie`
  - `AbstractUserAuthentication->svConfig`
  - `AbstractUserAuthentication->warningEmail`
  - `AbstractUserAuthentication->warningPeriod`
  - `AbstractUserAuthentication->warningMax`
  - `AbstractUserAuthentication->checkLogFailures()`
- As part of the session handling restructure, these methods have been marked as deprecated:
  - `AbstractUserAuthentication->createSessionId()`
  - `AbstractUserAuthentication->fetchUserSession()`

Deprecated/Removed Functions

Workspaces

- Several internal changes have been made to the Workspaces.
- Developers who built extensions that use Workspace-functionality are adviced to review the changes and test their custom code.
- Changes are for example (but not limited to):
    - The "New Placeholder" records have been removed (92791).
    - The "Move Placeholders" have been removed (92497).
    - The "swapping mode" of elements has been removed (92206).
    - The `_ORIG_pid` field has a new meaning (92598).
- The following two methods have been marked as deprecated:
    - `TYPO3\CMS\Core\Domain\Repository\PageRepository->fixVersioningPid()`
    - `TYPO3\CMS\Backend\Utility\BackendUtility::fixVersioningPid()`

Deprecated/Removed Functions

Frontend Usergroup Retrieval

- The code to retrieve usergroups when frontend users log-in has been streamlined.
- It is now consistent with the mechanism in for backend users.
- The following methods and properties have been removed:

  `TYPO3\CMS\Core\Authentication\AuthenticationService`

  - `->getGroups()`
  - `->getSubGroups()`
  - `->db_groups`
- The following methods have been marked as internal or removed:

  `TYPO3\CMS\Core\Authentication\FrontendUserAuthentication`

  - `->TSdataArray`
  - `->userTS`
  - `->userTSUpdated`
  - `->userData_change`

Deprecated/Removed Functions

T3Editor

- The T3Editor has been refactored into a custom HTML element:

  `<typo3-t3editor-codemirror>`

- This element features a new JavaScript module:

  `TYPO3\CMS\T3editor\Element\CodeMirrorElement`

- Using the CSS class "`t3editor`" is now **deprecated**:

  `<textarea class="t3editor"> ...  </textarea>`

Deprecated/Removed Functions

SortableJS

- The JavaScript library `SortableJS` has been renamed to `sortablejs`.
- This change was required due to the import of TypeScript declarations of `SortableJS`.
- The previous name `Sortable` has been declared **deprecated**.
- Extension developers are advised to use the library with its new name.

Deprecated/Removed Functions

jQuery in Tooltips

- Passing jQuery objects to the methods `show()` and `hide()` of the module `TYPO3/CMS/Backend/Tooltip` has been marked as **deprecated**.
- This results in a deprecation warning in the browser's console.
- Migration options: Pass a single `HTMLElement` or a `NodeList` to the functions.

Deprecated/Removed Functions

TypoScriptParser Properties

- The following properties related to the TypoScript parser have been marked as **deprecated**:
  - `TYPO3\CMS\Core\TypoScript\Parser\TypoScriptParser->breakPointLN`
  - `TYPO3\CMS\Core\TypoScript\Parser\TypoScriptParser->parentObject`
  - `TYPO3\CMS\Core\TypoScript\TemplateService->ext_constants_BRP`
  - `TYPO3\CMS\Core\TypoScript\TemplateService->ext_config_BRP`

Deprecated/Removed Functions

Property `special` of TCA `type=select`

- The property `special` of the TCA `type=select` has been marked as **deprecated**
- Developers are adviced to use `itemsProcFunc` instead:

```
'config' => [
  'type' => 'select',
  // 'special' => 'tables',
  'itemsProcFunc' => MyExtension::class . '->method',
  ...
]
```

- Further details about `itemsProcFunc` can be found in the documentation

Deprecated/Removed Functions

Legacy-tree Navigation Frame

- Backend modules that use the old `file_navframe` registration option are marked as **deprecated**
- Extension developers are adviced to update the code in file `ext_tables.php`.

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::addModule(
  ...
  [
    // previously: 'navigationFrameModule' => 'file_navframe',
    'navigationComponentId' => 'TYPO3/CMS/Backend/Tree/FileStorageTreeContainer',
    ...
  ]
);
```

Deprecated/Removed Functions

Hook Deprecation

- The following hook has been marked as **deprecated**:
  `t3lib/class.t3lib_parsehtml_proc.php:transformation`

Deprecated/Removed Functions

Database Table `sys_language`

- Using the database table `sys_language` has been classified as **deprecated**
- Site languages used in site configurations are now independent of any `sys_language` record
- Developers are advised to read the available languages and associated information from the site configuration
- For example, use the new TCA type `language` instead of `foreign_table => sys_language` (see #57082 for details)

Deprecated/Removed Functions

HTTP Utility

- To properly follow the PSR-7 standard, the following methods from the HTTP Utility have been marked as **deprecated**:
  - `redirect()`
  - `setResponseCode()`
  - `setResponseCodeAndExit()`
- See the changelog for further details and migration options

Deprecated/Removed Functions

ViewHelper Deprecations

- The following ViewHelpers have been marked as **deprecated**:

```
TYPO3\CMS\Fluid\ViewHelpers\BaseViewHelper
TYPO3\CMS\Fluid\ViewHelpers\Be\ContainerViewHelper
TYPO3\CMS\Backend\ViewHelpers\ModuleLayoutViewHelper
TYPO3\CMS\Backend\ViewHelpers\ModuleLayout\MenuViewHelper
TYPO3\CMS\Backend\ViewHelpers\ModuleLayout\MenuItemViewHelper
TYPO3\CMS\Backend\ViewHelpers\ModuleLayout\Button\LinkButtonViewHelper
TYPO3\CMS\Backend\ViewHelpers\ModuleLayout\Button\ShortcutButtonViewHelp
```

Deprecated/Removed Functions

`getPublicUrl()` on FAL Objects

- When accessing `getPublicUrl()` on a FAL object, the relative path could be retrieved by setting the following variable:
  `$relativeToCurrentScript = true`
- As this is not required anymore (see further details in the changelog), this has been marked as **deprecated**
- Accessing `isRelativeToCurrentScript()` on the `GeneratePublicUrlForResourceEvent` is also marked as **deprecated**
- TYPO3's Extension Scanner detects the usage of these cases

Deprecated/Removed Functions

Method `arrayDiffAssocRecursive()`

- The behavior of the following method has changed:
  `TYPO3\CMS\Core\Utility\ArrayUtility::arrayDiffAssocRecursive()`
- The function now mimics the behavior of the PHP function `array_diff_assoc()` as the name suggests
- This needs to be enabled by setting the third parameter to `true` for backwards-compatibility (`$useArrayDiffAssocBehavior`)
- Developers who require the behavior of the PHP function `array_diff_key()` should use the new method `arrayDiffKeyRecursive()` of the same class

Deprecated/Removed Functions

The SoftReferenceIndex Class

- The following class has been marked as **deprecated**:
  - `TYPO3\CMS\Core\Database\SoftReferenceIndex`
- The following methods have been marked as **deprecated**:
  - `TYPO3\CMS\Backend\Utility\BackendUtility::softRefParserObj`
  - `TYPO3\CMS\Backend\Utility\BackendUtility::explodeSoftRefParserList`
- Custom SoftReference parsers must implement the following interface: (otherwise a deprecation error will be logged)
  - `TYPO3\CMS\Core\DataHandling\SoftReference\SoftReferenceParserInterface`

Deprecated/Removed Functions

Cache Backends

- The following caching framework backend implementations have been marked as **deprecated**:
  - `TYPO3\CMS\Core\Cache\Backend\PdoBackend`
  - `TYPO3\CMS\Core\Cache\Backend\WincacheBackend`
- Migration: switch to one of the other available cache backends

Deprecated/Removed Functions

JavaScript Deprecations

- The JavaScript object "`top.fsMod`" (used in the page/file tree in the TYPO3 backend) has been marked as **deprecated**
- The following globally available variables have been marked as **deprecated**:
  - `top.currentSubScript`
  - `top.currentModuleLoaded`
  - `top.nextLoadModuleUrl`
- The global function `jump()` has been marked as **deprecated**

Deprecated/Removed Functions

Edit Panel Deprecations

- The following TypoScript properties have been marked as **deprecated**:
  - `stdWrap.editPanel`
  - `stdWrap.editIcons`
- The following cObject has been marked as **deprecated**:
  - `EDITPANEL`
- The related methods and properties in the following PHP classes have been marked as **deprecated**:

```
TYPO3\CMS\Frontend\ContentObject\ContentObjectRenderer
TYPO3\CMS\Frontend\ContentObject\EditPanelContentObject
TYPO3\CMS\Frontend\Plugin\AbstractPlugin
TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController
```

Deprecated/Removed Functions

Edit Panel Deprecations

- The following PHP class has been marked as **deprecated**:
  - `TYPO3\CMS\Frontend\ContentObject\EditPanelContentObject`
- The following hook has been marked as **deprecated**:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']
  ['typo3/classes/class.frontendedit.php']
```

Deprecated/Removed Functions

TYPO3 CLI command `impexp`

- The following lowerCamelCase options of the TYPO3 CLI command "`impexp`" have been marked as **deprecated**:

Deprecated:	**Replacement**:
`impexp:export --includeRelated`	`impexp:export --include-related`
`impexp:export --includeStatic`	`impexp:export --include-static`
`impexp:export --excludeDisabledRecords`	`impexp:export --exclude-disabled-records`
`impexp:export --excludeHtmlCss`	`impexp:export --exclude-html-css`
`impexp:export --saveFilesOutsideExportFile`	`impexp:export --save-files-outside-export-file`
`impexp:import --updateRecords`	`impexp:import --update-records`
`impexp:import --ignorePid`	`impexp:import --ignore-pid`
`impexp:import --forceUid`	`impexp:import --force-uid`
`impexp:import --importMode`	`impexp:import --import-mode`
`impexp:import --enableLog`	`impexp:import --enable-log`

Deprecated/Removed Functions

PHP Function str_starts_with()

- The following methods have been marked as **deprecated**:
  - `StringUtility::beginsWith()`
  - `StringUtility::endsWith()`
  - `GeneralUtility::isFirstPartOfStr()`
  - `GeneralUtility::minifyJavaScript()`
- PHP v8 offers the function `str_starts_with()` that supports proper typing and features better performance

Deprecated/Removed Functions

Custom JSWindow Options

- The HMENU/TMENU options for opening links of a menu in a custom JavaScript popup have been marked as **deprecated**:

```
# deprecated:
page.123 = HMENU
page.123.1 = TMENU
page.123.1.JSWindow = 1
page.123.1.JSWindow.params = width=200,height=300,status=0,menubar=0
```

- Calling a frontend page with HMENU and JSwindow popups will trigger a PHP deprecation warning

Deprecated/Removed Functions

Including External CSS

- The option to use the `@import` syntax for including external CSS files through TypoScript has been marked as **deprecated**:

```
# deprecated:
page = PAGE
page.includeCSSLibs.file1 = fileadmin/benni.css
page.includeCSSLibs.file1.import = 1
```

- Migration options:
  - use the `<link>`-tag or
  - create an inlineCSS TypoScript manually

Deprecated/Removed Functions

Inline JavaScript in `fieldChangeFunc`

- The option to use an iframe in the navigation area for backend modules through "`fieldChangeFunc`" has been marked as **deprecated**:

```
// deprecated:
$this->data['parameterArray']['fieldChangeFunc']['example'] = "alert('demo');";
$parameterArray['fieldChangeFunc']['example'] = "alert('demo');";
```

- The TYPO3 Core natively supports *Web Components* which should be used instead
- See the changelog for migration options and a demo extension as an example

Deprecated/Removed Functions

`SectionMarkupGenerated` Events

- The following methods have been marked as **deprecated**:

```
TYPO3\CMS\Backend\View\Event\BeforeSectionMarkupGeneratedEvent->getPageLayoutView()
TYPO3\CMS\Backend\View\Event\BeforeSectionMarkupGeneratedEvent->getLanguageId()
TYPO3\CMS\Backend\View\Event\AfterSectionMarkupGeneratedEvent->getPageLayoutView()
TYPO3\CMS\Backend\View\Event\AfterSectionMarkupGeneratedEvent->getLanguageIdg()
```

Deprecated/Removed Functions

TypoScript `parseFunc.sword`

- The TypoScript option `parseFunc.sword` allows to wrap search words in a special wrap when `no_cache = 1` is set
- This functionality has been deprecated
- Installations are affected that are accessed with, for example, a request such as:

```
https://example.com/?no_cache=1&sword_list%5B%5D=search
```

Deprecated/Removed Functions

Authentication Objects

- The *first* arguments of the following methods have been marked as **deprecated**:

  `TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`

  - `->writeUC()`
  - `->unpack_uc()`

  `TYPO3\CMS\Core\Authentication\BackendUserAuthentication`

  - `->backendCheckLogin()`

- The *third* argument of the following method has been marked as **deprecated**:

  `TYPO3\CMS\Core\Authentication\BackendUserAuthentication`

  - `->isInWebMount()`

Deprecated/Removed Functions

Various `getInstance()` Methods

- The following methods have been marked as **deprecated**:

```
TYPO3\CMS\Core\Resource\Index\ExtractorRegistry::getInstance()
TYPO3\CMS\Core\Resource\Index\FileIndexRepository::getInstance()
TYPO3\CMS\Core\Resource\Index\MetaDataRepository::getInstance()
TYPO3\CMS\Core\Resource\OnlineMedia\Helpers\OnlineMediaHelperRegistry::getInstance()
TYPO3\CMS\Core\Resource\Rendering\RendererRegistry::getInstance()
TYPO3\CMS\Core\Resource\TextExtraction\TextExtractorRegistry::getInstance()
TYPO3\CMS\Form\Service\TranslationService::getInstance()
TYPO3\CMS\T3editor\Registry\AddonRegistry::getInstance()
TYPO3\CMS\T3editor\Registry\ModeRegistry::getInstance()
```

Deprecated/Removed Functions

RequireJS Callback Functions

- Using RequireJS callback functions has been marked as **deprecated** (for example: `$resultArray['requireJsModules']` with scalar `string`)
- These calls should be replace by new `JavaScriptModuleInstruction` declarations
- See changelog for further details and migration options

Deprecated/Removed Functions

Miscellaneous

- The User TSconfig option `lockToIP` has been removed.
- The function "**lock to domain**" has been removed for frontend/backend users and groups.
- The following Workspaces-related classes and interfaces have been removed:
  - `TYPO3\CMS\Workspaces\ColumnDataProviderInterface`
  - `TYPO3\CMS\Workspaces\Service\AdditionalColumnService`
  - `TYPO3\CMS\Workspaces\Service\AdditionalResourceService`
- The property `sendNoCacheHeaders` has been removed from the following class:

  `TYPO3\CMS\Authentication\AbstractUserAuthentication`

Deprecated/Removed Functions

Miscellaneous

- The following methods of the `ReferenceIndex` class have been marked as deprecated:
  - `ReferenceIndex->enableRuntimeCache()`
  - `ReferenceIndex->disableRuntimeCache()`
- The following methods of the `BackendUserAuthentication` class have been removed:
  - `BackendUserAuthentication->groupList`
  - `BackendUserAuthentication->includeGroupArray`
- The method `GeneralUtility::uniqueList()` has been marked as deprecated (use the new method `StringUtility::uniqueList()`instead)

Deprecated/Removed Functions

Miscellaneous

- Calling `BackendUtility::wrapClickMenuOnIcon()` with more than four arguments will trigger a deprecation warning now.
- The following methods have been marked as deprecated:
  - `GeneralUtility::hideIfNotTranslated()`
  - `GeneralUtility::hideIfDefaultLanguage()`

  Use the following BitSet-based PHP class instead:

  `TYPO3\CMS\Core\Type\Bitmask\PageTranslationVisibility`
- The following hooks have been marked as deprecated:
  - `['recordlist/Modules/Recordlist/index.php']['drawHeaderHook']`
  - `['recordlist/Modules/Recordlist/index.php']['drawFooterHook']`

  Use the following PSR-14 event instead:

  `TYPO3\CMS\Recordlist\Event\RenderAdditionalContentToRecordListEvent`

Deprecated/Removed Functions

Miscellaneous

- The following two options have been removed from TYPO3's default configuration:
  - `BE/loginSecurityLevel`
  - `FE/loginSecurityLevel`
- The following container entry has been marked as **deprecated**:
  `TYPO3\CMS\Core\Localization\LanguageService`
- The following class has been marked as **deprecated**:
  `TYPO3\CMS\Core\Service\AbstractService`
- The following method has been marked as **deprecated**:
  `BackendUtility::viewOnClick()`

Deprecated/Removed Functions

Miscellaneous

- The following methods of the `GeneralUtility` class have been marked as **deprecated**:
  - `rmFromList()`
  - `stdAuthCode()`
  - `compileSelectedGetVarsFromArray()`
- The callback function of the method `run()` (1st argument) of the following interface has been marked as **deprecated**:

```
TYPO3\CMS\Core\Core\ApplicationInterface->run($execute)
```

- The popular inline JavaScript functions `goToModule()` has been marked as **deprecated**

Deprecated/Removed Functions

Miscellaneous

- The following Form Framework setup mixins have been **removed**:
  `TYPO3.CMS.Form.mixins.*`
- The "`CategoryRegistry`" and the following method have been marked as **deprecated**:
  `ExtensionManagementUtility::makeCategorizable()`
  (use the new TCA type "`category`" instead)
- The following method has been marked as **deprecated**:
  `GeneralUtility::shortMD5()`
  `DatabaseTreeDataProvider::setRootUid()`
  `DatabaseTreeDataProvider::getRootUid()`

Deprecated/Removed Functions

Miscellaneous

- The following unused properties have been marked as **deprecated**:
  - In class `TYPO3\CMS\Frontend\`
    - `ContentObject->align`
    - `ContentObject->oldData`
    - `ContentObject->alternativeData`
    - `ContentObject->currentRecordTotal`
    - `ContentObject->recordRegister`
  - In class `TYPO3\CMS\Frontend\Controller\`
    - `TypoScriptFrontendController->cObjectDepthCounter`

Deprecated/Removed Functions

Miscellaneous

- Declaring the property `$cObj` with a public visibility in a class called by `ContentObjectRenderer->callUserFunction()` triggers a deprecation error
- The fourth argument "`$skipSorting`" of the following method has been marked as **deprecated**:
  `TYPO3\CMS\Core\Database\RelationHandler->writeForeignField()`

Deprecated/Removed Functions

Miscellaneous

- The validator "Non-XML text" has been removed from the UI of the form editor
- The following hooks have been marked as **deprecated**:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['FileDumpEID.php']['checkFileAccess']
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['fileList']['editIconsHook']
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['GLOBAL']['extTablesInclusion-PostProcessing']
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['typo3/class.db_list_extra.inc']['actions']
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_extfilefunc.php']['processData']
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['additionalBackendItems']['cacheActions']
```

- The return type `array` of the following function has been removed:
  (the function will return the `LinkResultInterface` in TYPO3 v12)

```
TYPO3\CMS\Frontend\Typolink\AbstractTypolinkBuilder::build()
```

Deprecated/Removed Functions

Miscellaneous

- Calling the DataHandler `process_cmdmap` with an incoming command for `inlineLocalizeSynchronize` and a payload consisting of comma-separated values triggers a deprecation warning
- The following hooks have been marked as **deprecated**:

```
$GLOBALS['TBE_MODULES_EXT']['xMOD_db_new_content_el']['addElClasses']
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['typo3/browse_links.php']['browserRendering']
```

- The following method has been marked as **deprecated**:

```
\TYPO3\CMS\Core\Utility\GeneralUtility->isAbsPath()
```

Deprecated/Removed Functions

Miscellaneous

- The following class has been marked as **deprecated**:

```
TYPO3\CMS\Backend\View\BackendTemplateView
```

- The following methods have been marked as **deprecated**:

```
TYPO3\CMS\Core\Database\RelationHandler->remapMM()

TYPO3\CMS\Core\Configuration\FlexForm\FlexFormTools->getArrayValueByPath()
TYPO3\CMS\Core\Configuration\FlexForm\FlexFormTools->setArrayValueByPath()

TPO3\CMS\Backend\Template\ModuleTemplate->getIconFactory()
TPO3\CMS\Backend\Template\ModuleTemplate->getPageRenderer()

\TYPO3\CMS\Backend\Template\Components\AbstractControl->setOnClick()
\TYPO3\CMS\Backend\Template\Components\AbstractControl->getOnClick()
```

Deprecated/Removed Functions

Miscellaneous

- The following property has been marked as **deprecated**:

  `TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController->ATagParams`

Miscellaneous

Miscellaneous

Miscellaneous

X-Redirect-By Header

- On redirects, TYPO3 now sends an additional HTTP header to indicate which of the following page types triggered the redirect:
  - Shortcut
  - Mountpoint pages (where applicable)
  - Link to external URL
- If enabled in the global configuration, the page ID is added to the output (should only be enabled for debugging purposes):

```
# See: $GLOBALS['TYPO3_CONF_VARS']['FE']['exposeRedirectInformation']

# exposeRedirectInformation is disabled:
X-Redirect-By: TYPO3 External URL
# exposeRedirectInformation is enabled:
X-Redirect-By: TYPO3 External URL at page with ID 456
```

Miscellaneous

Link Validator

- The Scheduler task to generate email report about broken links has been significantly reworked and improved and many ways:
  - Fluid-based emails are now used
  - Task configuration extended
  - Email contains more detailed reports
  - Code refactoring
- Additionally, a PSR-14 event has been introduced to let developers manipulate the link analyser result.
- See change log for further details.

Miscellaneous

Link Validator

- The LinkValidator supports several link types such as `db`, `file`, `external`, etc.
- The type `linkhandler`has been outdated for a long time, and consequently removed now.
- It's unlikely that this change requires a migration.
- However, integrators should check and remove the `linkhandler` type from the page TSconfig if it exists:

```
mod.linkvalidator.linktypes = db,file,external,linkhandler
```

- Also, the following configuration should not be used anymore:

```
mod.linkvalidator.linkhandler.reportHiddenRecords = 0
```

Miscellaneous

Security

- To enhance the security, TYPO3 v11 does not expose the same cookie for HTTP and HTTPS connections, if the instance is in the HTTPS context.
  - If the site runs on HTTPS: the cookie is only exposed via HTTPS.
  - If the site runs on HTTP: the cookie is available via HTTP and HTTPS.

  As part of this change, the following global configuration has been removed: `$TYPO3_CONF_VARS[SYS][cookieSecure]`
- TYPO3 removes all session data on frontend user logout
  (this may affect installations that rely on keeping the session, e.g. for cart functionality)

Miscellaneous

Backend User Language

- The backend user's preferred language is now stored in the database under `be_users.lang` even if the user changes the language
- Previously, the language information was only stored in this field initially and then transferred to the user configuration `user->uc[lang]`
- An upgrade wizard migrates existing `uc` values into the database field
- The `uc` entry is kept in sync for backwards-compatibility

Miscellaneous

XLIFF Version 1.2

- All language files in the TYPO3 Core use XLIFF version 1.2 now
- This allows us to use a validation schema
- The TYPO3-specific XLIFF file IDs `t3:id` have been dropped too
- Example:

< TYPO3 v11.2

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.0" xmlns:t3="http://typo3.org/schemas/xliff">
  <file t3:id="1521629231" source-language="en" ... >
  ...
```

>= TYPO3 v11.2

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
  <file source-language="en" ... >
  ...
```

Miscellaneous

Language Support

- The TYPO3 backend now supports two additional languages:
  - Shona (Bantu) (ISO 639-1 code: `sn`)
  - Welsh (ISO 639-1 code: `cy`)

- This change enables us to:
  - Fetch translated labels for these languages from translation.typo3.org and CrowdIn
  - Switch the backend interface to these languages
  - Use these languages in a site configuration
  - Create translation files (e.g. `<code>.locallang.xlf`)

Miscellaneous

Login Throttling

- TYPO3 now limits backend and frontend logins using the Symfony library "Rate Limiter"
- A PSR-15 middleware now uses the rate limit to block further authentication attempts if too many failed logins from the same IP address were detected
- This approach replaces the old solution that was not a state-of-the-art implementation from a security perspective
- See the changelog for configuration options, default values, and further details

Miscellaneous

Package Updates

- The following packages have been updated:
  - **EmailValidator** from v2.1.25 to v3.1.1 (https://github.com/egulias/EmailValidator)
  - **Bootstrap** to v5.1.0 (https://getbootstrap.com/)
  - **Guzzle** to v7.3.0 (https://github.com/guzzle/guzzle)
  - **guzzlehttp/psr7** to v2.0 (https://github.com/guzzle/psr7)

Miscellaneous

Packages/Libraries Updated

- The following packages and libraries have been updated:
  - CKEditor from version 4.15.0 to 4.16.0
  - Bootstrap v5 from beta-2 to beta-3
- Compatibility with Doctrine DBAL version 2.13 has been added

Miscellaneous

Miscellaneous

- TYPO3 v11.0 features jQuery version 3.5.x.
- "Indexed Search" does not log user-specific details in the search statistics for privacy reasons anymore.
- "Indexed Search" does not provide hooks for the "Crawler" extension anymore (`EXT:crawler`).

Miscellaneous

Miscellaneous

- Error pages are now generated through a TYPO3-internal sub-request to improve performance and address issues with load-balanced systems

Installation and Upgrade

There is no better time to check out TYPO3 v11 LTS

Installation and Upgrade

Composer-based Installation Method

- Installation using PHP Composer under Linux, macOS, and Windows 10:

```
$ cd /var/www/site/
$ composer create-project typo3/cms-base-distribution:^11 typo3v11
```

- Alternatively, create your custom `composer.json` file and run:

```
$ composer install
```

- The Composer Helper online tool makes package selection easy
- Further details are available in the Installation and Upgrade Guide
- **Using Composer is the recommended installation method!**

Installation and Upgrade

Classic Installation Method

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

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/11.5
$ tar xzf typo3_src-11.5.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-11.5.0 typo3_src
$ ln -s typo3_src/index.php
$ ln -s typo3_src/typo3
$ touch FIRST_INSTALL
```

- See Installation and Upgrade Guide for details about Microsoft Windows systems

Installation and Upgrade

Upgrade to TYPO3 v11 LTS

- Upgrades are only possible from TYPO3 v10 LTS.
- Sites using TYPO3 < v10 LTS should be updated to the latest v10.4.x first.
- General approach:
  - Check minimum system requirements (PHP, MySQL, etc.)
  - Review **deprecation_*.log** in old TYPO3 instance
  - Update all extensions to the latest compatible version
  - Run the Extension Scanner to detect potential issues
  - Deploy new sources and run Install Tool -> Upgrade Wizard
- Read more in the official TYPO3 Upgrade Guide.

Sources and Authors

Sources and Authors

Sources

TYPO3 News:

- https://typo3.org/project/news/

Release Infos:

- https://get.typo3.org/release-notes/11.x/TYPO3_CMS_11.5.0
- TYPO3 v11 ChangeLog
- typo3/sysext/core/Documentation/Changelog/11.5/*

TYPO3 Bug-/Issuetracker:

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

TYPO3 and Fluid Git Repositories:

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

Sources and Authors

TYPO3 CMS What's New Slides and Translations:

Pierrick Caillon, Richard Haeser, Jigal van Hemert,
Henrietta Kucsovan, Corina Miron, Sinisa Mitrovic,
Michael Schams, and Roberto Torresani

https://typo3.org/help/documentation/whats-new/