

TYP03 Version 11.4 – What's New

Summary of the new features, changes and improvements

Created by:
Michael Schams

TYPO3 Version 11.4 - What's New

Chapter Overview

Introduction

Backend User Interface

Changes for Integrators and Developers

Extbase and Fluid

Deprecated/Removed Functions

Miscellaneous

Sources and Authors

Introduction

The Facts

Introduction

TYPO3 Version 11.4 - The Facts

- Release date: 07 September 2021
- Release type: Sprint Release



Introduction

Executive Summary

We are almost there! The TYPO3 v11 release with long-term support is approaching fast. TYPO3 version 11.4 is the **feature freeze** version and marks the last phase of the v11 development cycle before the TYPO3 v11 LTS release.

This means that no new features are planned until the LTS release in October 2021. We are confident that TYPO3 v11.4 is stable and can be used in most projects in production.

Check it out: install the new version or run a test update of your existing instances and report any issues or hiccups you possibly encounter.

Introduction

System Requirements

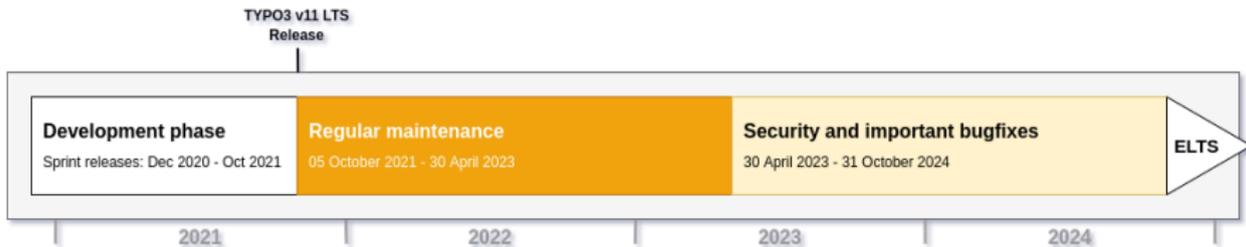
- PHP version 7.4 or 8.0
- PHP settings:
 - `memory_limit` \geq 256M
 - `max_execution_time` \geq 240s
 - `max_input_vars` \geq 1500
 - compilation option `--disable-ipv6` must not be used
- Most database servers supported by **Doctrine DBAL** also work with TYPO3. Tested DB engines are for example:



Introduction

Development, Support, and Maintenance Timeline

TYPO3 v11



Extended long-term support (ELTS)

The [TYPO3 GmbH](#) offers further support options for TYPO3 v11 LTS even after 31 October 2024 for up to two additional years.

Introduction

TYP03 v11 Release Dates

Planned release dates and their primary focus:

- 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 TYP03 backend
- v11.3 13/Jul/2021 UX improvements and PHP v8 support
- **v11.4 07/Sep/2021 Feature freeze**
- v11.5 05/Oct/2021 LTS Release (long-term support)

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

<https://typo3.org/article/a-first-glimpse-of-typo3-v11>

Introduction

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!**

Introduction

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/11.4
$ tar xzf typo3_src-11.4.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-11.4.0 typo3_src
$ ln -s typo3_src/index.php
$ ln -s typo3_src/typo3
$ touch FIRST_INSTALL
```

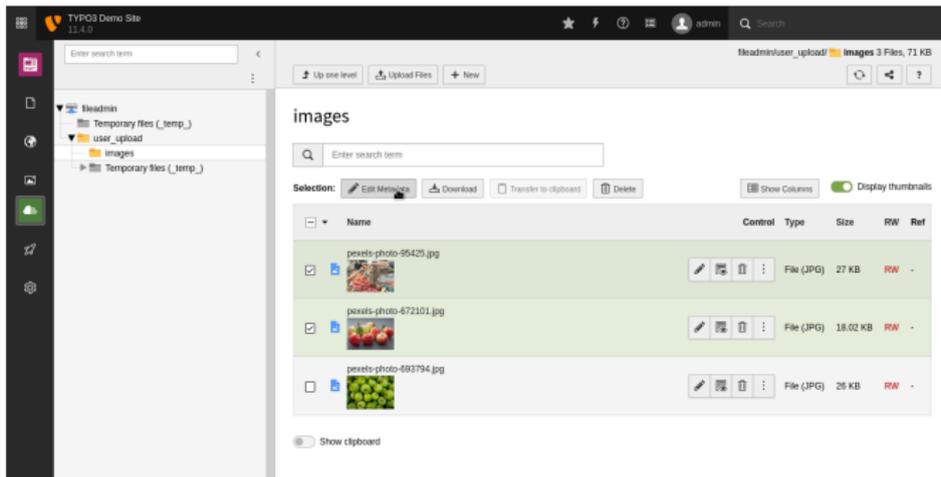
- See [Installation and Upgrade Guide](#) for details about Microsoft Windows systems

Chapter 1: Backend User Interface

Backend User Interface

Filelist Module (1)

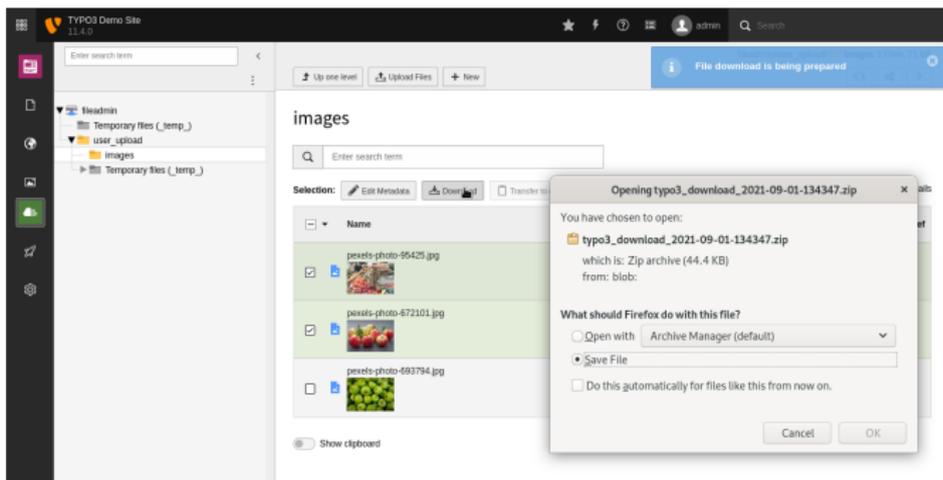
It is now also possible to execute actions in the **File** -> **Filelist** module directly without transferring files to TYPO3's clipboard first.



Backend User Interface

Filelist Module (2)

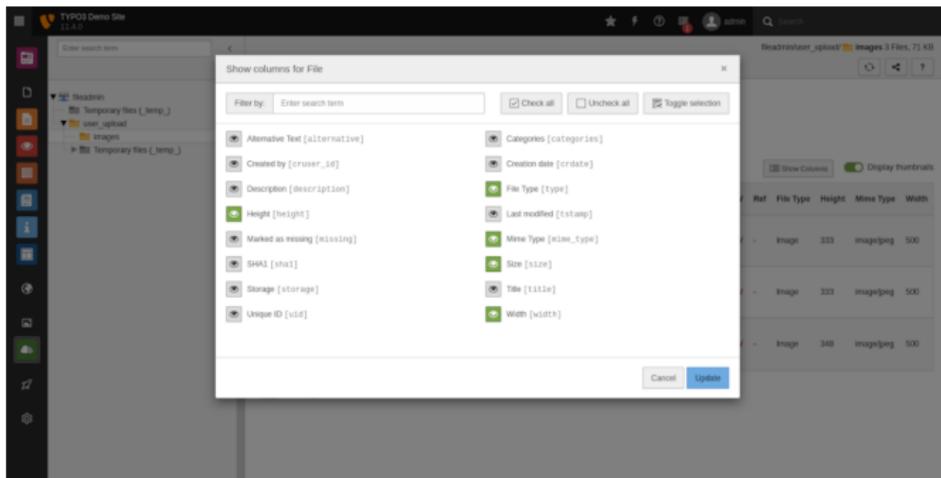
Backend users can now download multiple files or folders in one go by instructing TYPO3 to create a ZIP archive of them.



Backend User Interface

Filelist Module (3)

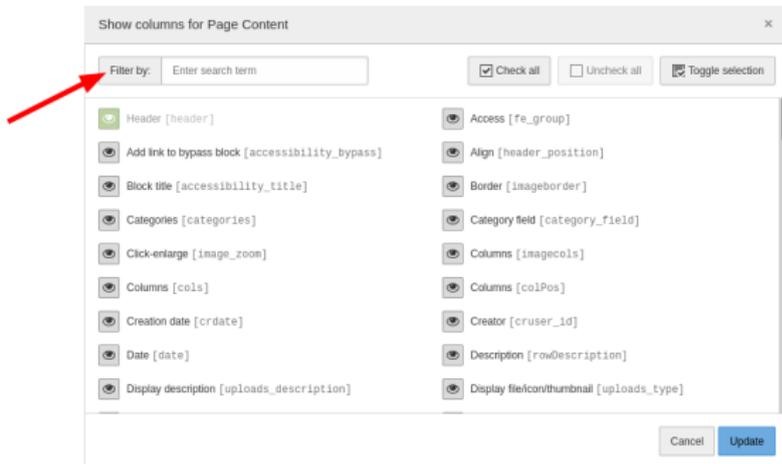
The Column Selector is now also available in **File** -> **Filelist**. Backend users can select which fields for files/folders should be shown.



Backend User Interface

Columns Selector Filter

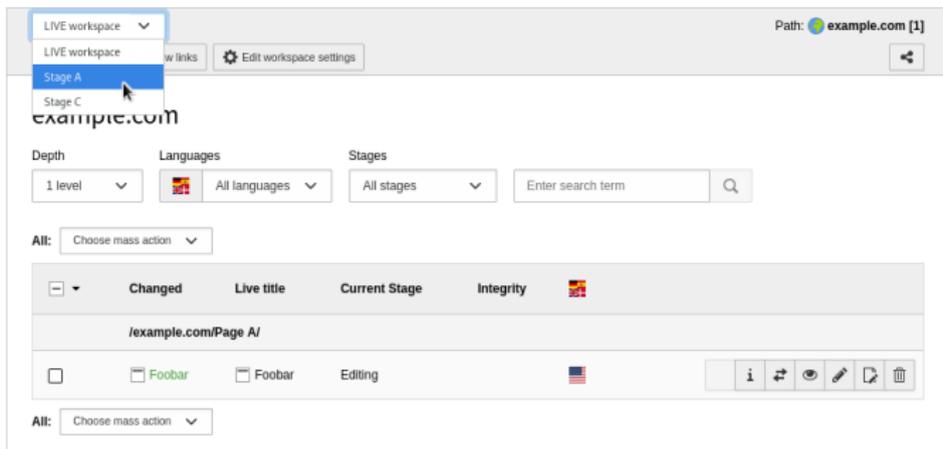
The column selector in the record list now features a filter at the top. This makes it easier for backend users to enable/disable the right items in particular for tables with many columns.



Backend User Interface

Workspaces Module (1)

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.



The screenshot displays the TYPO3 Backend User Interface for the Workspaces Module. At the top, a dropdown menu is open, showing options for switching workspaces: "LIVE workspace", "Stage A", and "Stage C". The "LIVE workspace" option is currently selected. Below the dropdown, there are buttons for "w links" and "Edit workspace settings". The path is shown as "Path: example.com [1]".

The main interface includes several filters and a search bar:

- Depth: 1 level
- Languages: All languages
- Stages: All stages
- Search: Enter search term

Below the filters, there is a "Choose mass action" dropdown. The main content area shows a table of workspace entries:

Changed	Live title	Current Stage	Integrity	
/example.com/Page A/				
<input type="checkbox"/>	<input type="checkbox"/> Foobar	<input type="checkbox"/> Foobar	Editing	

At the bottom, there is another "Choose mass action" dropdown and a set of icons for actions like info, refresh, eye, edit, and delete.

Backend User Interface

Workspaces Module (2)

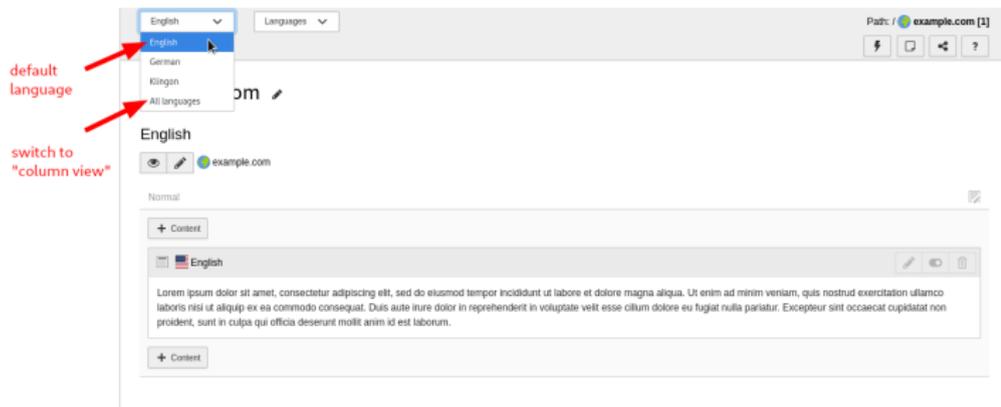
A new dropdown menu in the **Workspaces module** enables users to filter items for a specific stage, e.g. "Editing" or "Ready to publish".

The screenshot displays the TYPO3 Backend User Interface for the Workspaces module. At the top, there is a header with a 'Stage A' dropdown, a 'Path: Page A [2]' indicator, and buttons for 'Generate page preview links' and 'Edit workspace settings'. Below the header, the main content area is titled 'Page A'. It features several filters: 'Depth' (set to '1 level'), 'Languages' (set to 'All languages'), and 'Stages' (set to 'All stages'). A search bar is also present. Below the filters, there are 'All:' and 'Selection:' dropdown menus. The 'Stages' dropdown is open, showing 'All stages', 'Editing' (highlighted), and 'Ready to publish'. Below this, a table lists workspace items with columns for 'Changed', 'Live title', 'Current Stage', and 'Integrity'. The table shows one item with a green background, 'Foobar', in the 'Editing' stage. At the bottom, there are more 'All:', 'Selection:', and 'Choose selection action' dropdowns.

Backend User Interface

Languages/Translations

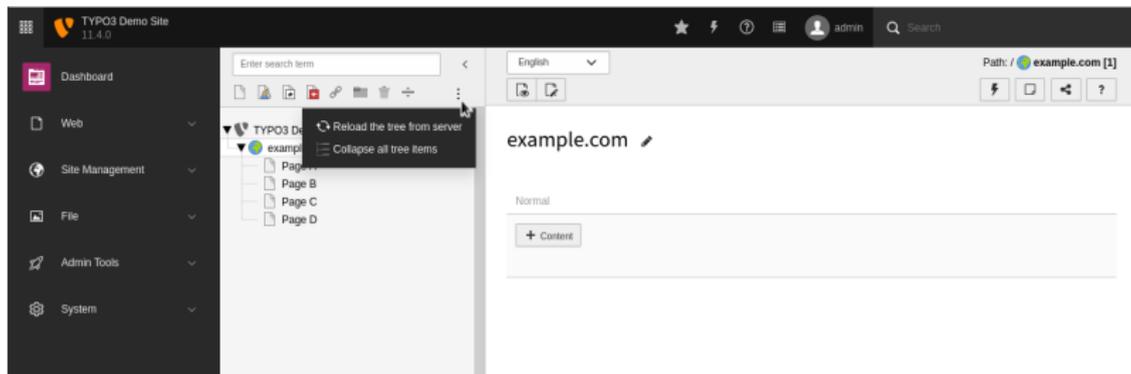
The "Languages" view under **Web** -> **Page** is now more consistent with the column view. The default language only shows content elements of this language. Backend users can use the "All languages" item to switch to the column view to show available languages.



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.

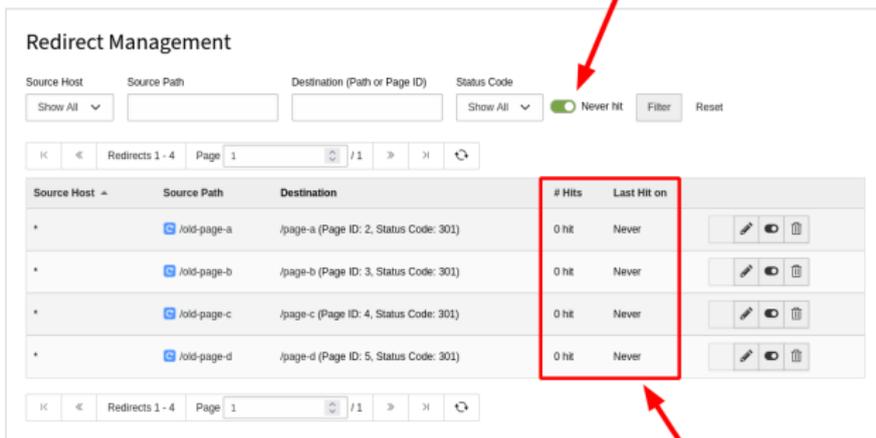


Backend User Interface

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.



The screenshot displays the 'Redirect Management' interface. At the top, there are search filters for 'Source Host', 'Source Path', 'Destination (Path or Page ID)', and 'Status Code'. A 'Show All' dropdown is next to each. A 'Never hit' toggle switch is turned on, highlighted by a red arrow. Below the filters is a 'Filter' button and a 'Reset' button. The main content area shows a table of redirects. The table has columns for 'Source Host', 'Source Path', 'Destination', '# Hits', and 'Last Hit on'. The '# Hits' and 'Last Hit on' columns are highlighted with a red box, and a red arrow points to the 'Never' value in the 'Last Hit on' column. The table contains four rows of redirects, all with '0 hit' and 'Never' status. At the bottom, there is a pagination bar showing 'Redirects 1 - 4' and 'Page 1'.

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

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)

Chapter 2: Changes for Integrators and Developers

Changes for Integrators and Developers

New TCA Type: `category` (1)

- 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 (2)

- 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

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

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

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

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

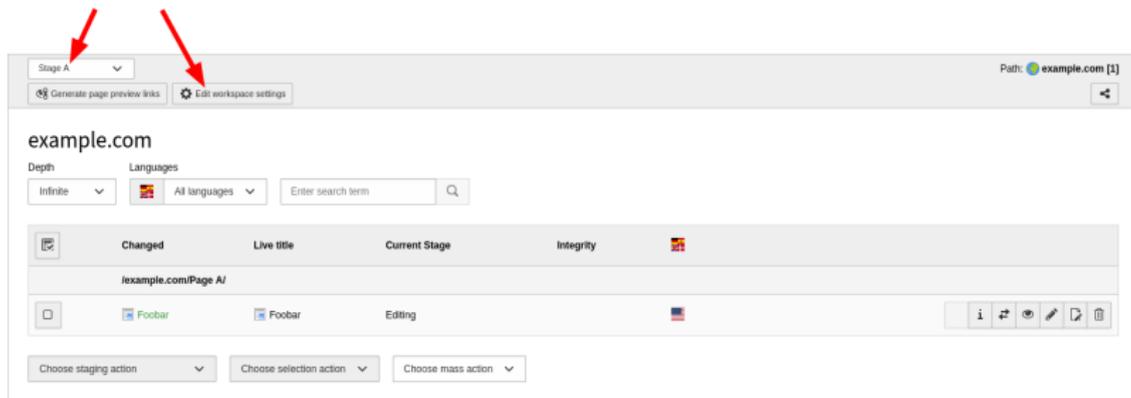
Forms Framework: `Datepicker.js`

- 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 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)



The screenshot displays the TYPO3 Workspaces module interface. At the top left, a dropdown menu is set to 'Stage A', with a red arrow pointing to it. To its right is a button labeled 'Edit workspace settings', also indicated by a red arrow. Below the header, the main content area shows a table with columns for 'Changed', 'Live title', 'Current Stage', and 'Integrity'. A table entry is visible for 'example.com/Page A/' with a 'Foobar' live title and 'Editing' current stage. At the bottom, there are three dropdown menus for 'Choose staging action', 'Choose selection action', and 'Choose mass action'.

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

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:

```

```

Changes for Integrators and Developers

TYPO3 CLI (1)

- 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 (2)

- 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 (3)

- 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

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

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

Composer-based Installations (1)

- 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 (2)

- 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

PSR-14 Events (1)

- 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 (2)

- 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

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\LoggerAwareTrait
```

- 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

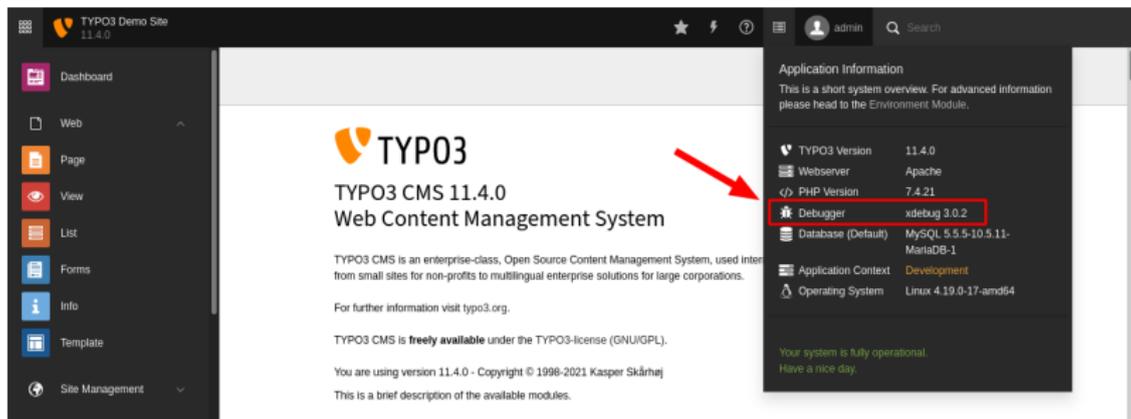
New Class `LinkResult`

- 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

Application Information

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



The screenshot displays the 'Application Information' section of the TYPO3 CMS 11.4.0 interface. The page title is 'TYPO3 CMS 11.4.0 Web Content Management System'. The main content area lists system details:

- TYPO3 Version: 11.4.0
- Webserver: Apache
- PHP Version: 7.4.21
- Debugger: xdebug 3.0.2 (highlighted with a red box and a red arrow)
- Database (Default): MySQL 5.5.5-10.5.11-MariaDB-1
- Application Context: Development
- Operating System: Linux 4.19.0-17-amd64

Below the list, there is a status message: 'Your system is fully operational. Have a nice day.'

Chapter 3: Extbase and Fluid

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

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

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

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 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

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`

Chapter 4: Deprecated/Removed Functions

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 (1)

- 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 (2)

- 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:

```
impexp:export --includeRelated
impexp:export --includeStatic
impexp:export --excludeDisabledRecords
impexp:export --excludeHtmlCss
impexp:export --saveFilesOutsideExportFile

impexp:import --updateRecords
impexp:import --ignorePid
impexp:import --forceUid
impexp:import --importMode
impexp:import --enableLog
```

Replacement:

```
impexp:export --include-related
impexp:export --include-static
impexp:export --exclude-disabled-records
impexp:export --exclude-html-css
impexp:export
    --save-files-outside-export-file

impexp:import --update-records
impexp:import --ignore-pid
impexp:import --force-uid
impexp:import --import-mode
impexp:import --enable-log
```

Deprecated/Removed Functions

Miscellaneous (1)

- 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()`
`DatabaseDataProvider::setRootUid()`
`DatabaseDataProvider::getRootUid()`

Deprecated/Removed Functions

Miscellaneous (2)

- 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 (3)

- 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 (4)

- 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()
```

Chapter 5: Miscellaneous

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

Miscellaneous

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

Chapter 6: 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_11.3.0
- [TYPO3 v11 Changelog](#)
- `typo3/sysexst/core/Documentation/Changelog/11.3/*`

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/>

Licensed under Creative Commons BY-NC-SA 3.0

