

TYP03 Version 10 LTS – What's New

Summary of the new features, changes and improvements

Created by:
Michael Schams

05/May/2020

Creative Commons BY-NC-SA 3.0



TYPO3 Version 10 LTS - What's New

Chapter Overview

Introduction

Dashboard

Backend User Interface

Form Framework

In-depth Changes

Deprecated/Removed Functions

Installation and Upgrade

Sources and Authors

Introduction

Key facts and figures about TYPO3 v10 LTS

Introduction

TYPO3 Version 10 LTS

- Release date: 21 April 2020
- Release type: LTS (long-term support)



Introduction

Executive Summary

TYPO3 v10.4 (also called TYPO3 v10 LTS indicating this is a long-term support version) is our new flagship and, without doubt, one of the most advanced PHP-based open-source content management systems on the market to date.

After publishing five sprint releases since July 2019, we can proudly claim that we have equipped TYPO3 with the top modern PHP libraries and that we have introduced some fantastic new enterprise features.

This document summarizes the most important changes between TYPO3 v9 LTS and v10 LTS from a technical perspective.

Introduction

System Requirements

- PHP version 7.2, 7.3 or 7.4
- PHP settings:
 - `memory_limit` \geq 256M
 - `max_execution_time` \geq 240s
 - `max_input_vars` \geq 1500
 - compilation option `--disable-ipv6` must not be used
- Required PHP extensions:
filter, hash, openssl, pcre \geq 8.38, session, SPL, standard, xml, zip and zlib

Introduction

System Requirements

- Webserver such as Apache, Nginx, IIS, etc.
- All databases supported by **Doctrine DBAL** are also supported by TYPO3. For example:



- 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

Development Timeline

Sprint Releases published:

- v10.0 23/Jul/2019 Pave the way for exciting new concepts and APIs
- v10.1 01/Oct/2019 Routing Improvements and Site Handling v2
- v10.2 03/Dec/2019 Fluid/Rendering Engine Improvements
- v10.3 25/Feb/2020 Feature Freeze
- v10.4 21/Apr/2020 LTS Release (long-term support)

Introduction

Long-term Support



- TYPO3 version 10.4 is a LTS release (long-term support)
- Regular maintenance and bugfixes until October 2021
- Security and critical bugfixes until April 2023

Extended Support

[TYPO3 GmbH](#) offers extended long-term support (ELTS) for TYPO3 v10 LTS until April 2026.

Dashboard

System information, news, and much more for backend users

Dashboard

Backend View

A dashboard has been introduced that shows important system information to the currently logged-in backend user.

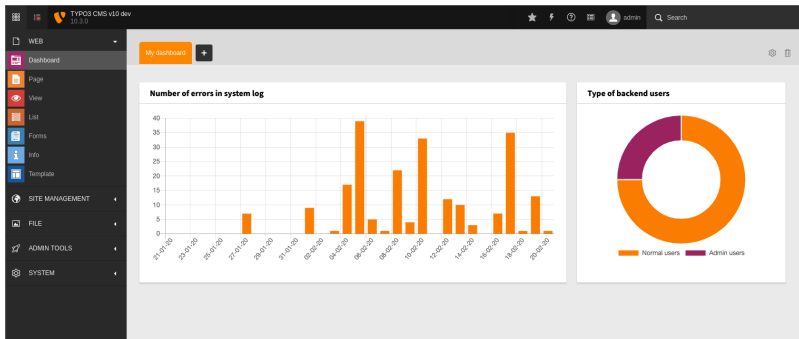
The screenshot displays the TYPO3 CMS v10 dev backend dashboard. The top navigation bar includes the user name 'admin' and a search function. The left sidebar lists various system components: WEB, Dashboard, Page, View, List, Forms, Info, Templates, SITE MANAGEMENT, FILE, ADMIN TOOLS, and SYSTEM. The main content area is titled 'My dashboard' and contains three primary sections:

- TYPO3 CMS 10.3.0-dev**: A section providing an overview of the CMS as an enterprise-class, open-source system, along with a link to the official website (typo3.org) and information about its license (GNU/GPL).
- Getting Started with TYPO3**: A section for new users, featuring a 'Getting Started Tutorial' button.
- TYPO3 news**: A section for the latest news, including updates on maintenance releases (9.5.14 LTS and 8.7.31 LTS), dashboard updates, and structured content initiatives.

Dashboard

Backend View

Users can create their own dashboards and add, remove, and re-arrange "widgets".



Dashboard

Options for Integrators

- Dashboard *presets* can be configured for new users or for users who deleted all their dashboards.
- This can be used to show a "Getting Started" dashboard by default.
- Example TSconfig:

```
options.dashboard.dashboardPresetsForNewUsers = default, dashboardPreset-myPreset
```

- Multiple dashboard presets can be defined in a comma separated list.

Dashboard

Custom Widgets

- TYPO3 v10 LTS comes with a number of widgets out of the box (for example: general information, failed backend logins, TYPO3 news, links to the documentation, etc.)
- Developers can easily build custom widgets as extensions.
- Register and configure widgets in a YAML file:
`EXT:myextension/Configuration/Services.yaml`
- The "Dashboard" system extension provides some typical widgets types (bar chart, call-to-action button, doughnut chart, list, number with icon, and rss widget)
- Read more about the dashboard in the [TYPO3 documentation](#).

Backend User Interface

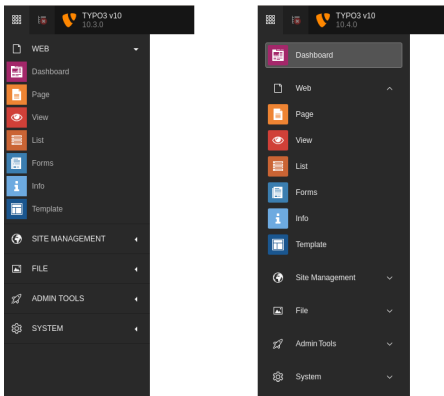
Backend User Interface

The TYPO3 administration interface is now better than ever

Backend User Interface

Backend UI Adjustments

Slightly altered UI of the backend modules column.



Backend User Interface

Filelist Sorting

Files can now be sorted by their meta data title in the "File Links" content element.

The screenshot displays the configuration interface for the "File Links" content element. A dropdown menu is open, showing sorting options: "no sorting", "by file extension [extension]", "by filename [name]", "by file type [type]", "by file size [size]", "by file creation date [creation_date]", "by file modification date [modification_date]", "by file metadata title [title]" (highlighted), and "by file metadata title [title]" (selected). Below the dropdown, the "Sorting direction" field is set to "Ascending" and the "Target" field is set to "None".

Filelinks Layout

Display File Size Information [filelink_size] [0]

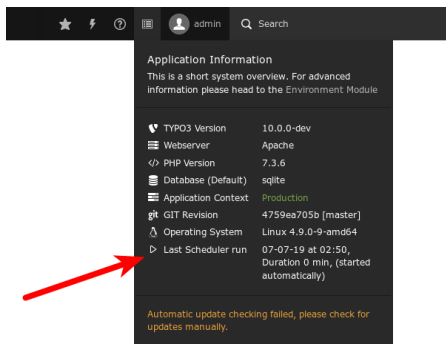
Display description [uploads_description] [0]

Display file/Icon/thumbnall [uploads_type]

Backend User Interface

System Information Toolbar

The system information toolbar now shows information about the TYPO3 scheduler.



Application Information

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

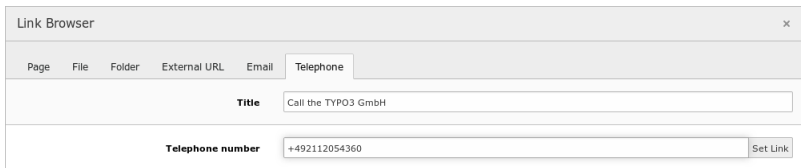
♥ TYPO3 Version	10.0.0-dev
☰ Webservice	Apache
</> PHP Version	7.3.6
🗄 Database (Default)	sqlite
☰ Application Context	Production
git GIT Revision	4759ea705b [master]
🐧 Operating System	Linux 4.9.0-9-amd64
▷ Last Scheduler run	07-07-19 at 02:50, Duration 0 min, (started automatically)

Automatic update checking failed, please check for updates manually.

Backend User Interface

Link Handler

A new link handler has been added that lets backend users set links to phone numbers using the `tel:` protocol.



The screenshot shows a 'Link Browser' window with a close button (x) in the top right corner. Below the title bar is a navigation bar with tabs: Page, File, Folder, External URL, Email, and Telephone. The 'Telephone' tab is selected. The main content area contains two form fields: 'Title' with the value 'Call the TYPO3 GmbH' and 'Telephone number' with the value '+492112054360'. A 'Set Link' button is located to the right of the telephone number field.

```
<div class="ce-bodytext">
  <p>
    Give the TYPO3 GmbH a
    <a href="tel:+492112054360" title="Call the TYPO3 GmbH">call</a>
    if you have any further questions.
  </p>
</div>
```

Backend User Interface

EXT:seo: Backend View

The SEO system extension now supports change frequencies and priorities for the Sitemap. Page properties (tab "SEO") feature two new fields.

Sitemap

Change frequency [sitemap_changefreq] **Priority** [sitemap_priority]

None

- None
- Always [always]
- Hourly [hourly]
- Daily [daily]**
- Weekly [weekly]
- Monthly [monthly]
- Yearly [yearly]
- Never [never]

Backend User Interface

EXT:seo: Configuration Options for Integrators

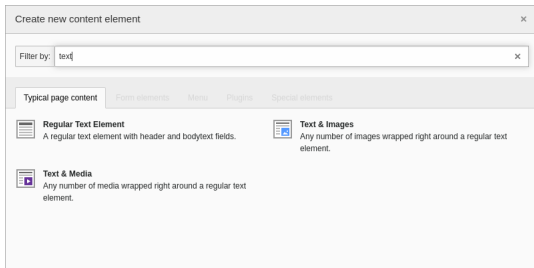
These settings can also be defined in TypoScript, mapped to fields in the database.

```
plugin.tx_seo {
    config {
        xmlSitemap {
            sitemaps {
                <unique key> {
                    provider = TYPO3\CMS\Seo\XmlSitemap\RecordsXmlSitemapDataProvider
                    config {
                        ...
                        changeFreqField = news_changefreq
                        priorityField = news_priority
                        ...
                    }
                }
            }
        }
    }
}
```

Backend User Interface

New Content Element Search

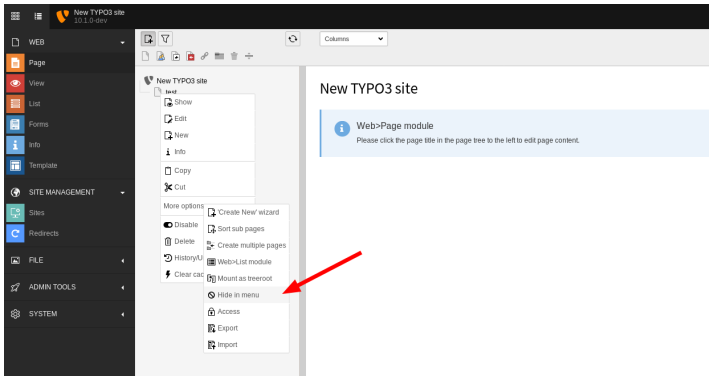
Backend users can now search for content element types in the "New Content Element" wizard:



Backend User Interface

Hide/Show in Menu

A new entry was added to the context menu to show/hide pages in the menu.



Backend User Interface

Extension Manager

The Extension Manager now shows links to extension documentation.

Get Extensions

Time since last update 1 min

Actions	Extension	Key	Version	Last Updated	Description	Manual	State
	Zabbix Client	zabbix_client	0.2.5	16-11-19	Client for zabbix monitoring system. Secure up...		stable
	Dynamic Content Elements (DCE)	dce	2.3.1	16-11-19	Best flexform based content elements since 2012....		stable
	Minify for vhs assets	vhsminify	2.0.3	16-11-19	minify for vhs assets		beta
	Mailman Extension	mailmanext	1.2.0	15-11-19	An extension to manage mailman mailinglists		beta
	AOE Database Sequenzer	aoe_dbsequenzer	3.0.0	15-11-19	With this extension you can ensure different...		stable
	Translatable Labels for Editors	translatelabels	2.0.6	15-11-19	This extension extends the TYPO3 translation...		beta
	Modules	modules	3.2.0	15-11-19	Modules - Little helper for creating...		stable
	[NITSAN] News Slider	ns_news_slider	2.0.1	15-11-19	Do you want cool sliders for most popular TYPO3...		stable

Backend User Interface

Extension Manager

System and 3rd-party extensions can now be listed separately in the Extension Manager.

Installed Extensions

Search

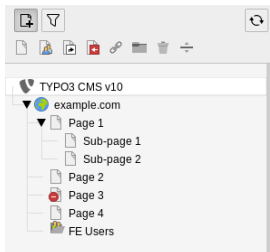
All System Local

Upd.	AID	Extension	Key	Version	State	Type	Actions
		Make BE user password really secure	be_secure_pw	9.1.0	stable	Local	
		Bootstrap Package	bootstrap_package	11.0.2	stable	Local	
		The official Introduction Package	introduction	4.2.0	stable	Local	
		Reverse Proxy - Cache Manager	proxycachemanager	3.0.1	stable	Local	
		TYPO3 CMS Backend Styleguide and Testing use cases	styleguide	10.0.0	stable	Local	

Backend User Interface

Pagetree Accessibility

Backend users can now use their keyboard to navigate through the pagetree. For example the arrow keys, "home", "end", "enter", "space", etc. This is in accordance to the best practices as described in [WAI-ARIA Authoring Practices 1.1](#) by the W3C.



Backend User Interface

Backend User Module

- A new detail view of backend user records shows all relevant data.
- Additional fields have been added to the function to compare users.
- This function also takes subgroups into account now.
- The user interface of the module will be adjusted and optimized further.
- These changes make it easier for integrators/administrators to check and compare user permissions without switching to the user.

Backend User Interface

Backend User Module

- Integrators are now able to compare individual backend usergroups.

The screenshot displays a comparison interface for two user groups: Department A [1] and Department B [2].

Record information:

Inherit settings from groups	Direct	Inherit	Direct	Inherit
Department A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Department B	<input checked="" type="checkbox"/>

...

DB mountpoints:

Branch A [1]	Branch A [1]
	Branch B [2]

File Mounts:

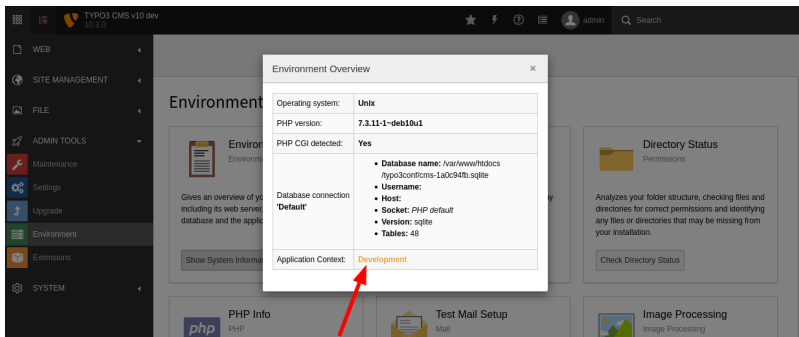
Fileoperation permissions:

Department A [1]	Department B [2]
<input checked="" type="checkbox"/> Directory: Read	<input checked="" type="checkbox"/> Directory: Read
<input checked="" type="checkbox"/> Directory: Write	<input checked="" type="checkbox"/> Directory: Write
<input checked="" type="checkbox"/> Directory: Add	<input type="checkbox"/> Files: Read
<input checked="" type="checkbox"/> Directory: Rename	<input type="checkbox"/> Files: Write
<input checked="" type="checkbox"/> Directory: Move	<input type="checkbox"/> Files: Add
<input checked="" type="checkbox"/> Directory: Delete	<input type="checkbox"/> Files: Move
<input type="checkbox"/> Files: Read	<input type="checkbox"/> Files: Copy
<input type="checkbox"/> Files: Write	<input type="checkbox"/> Files: Delete
<input type="checkbox"/> Files: Add	
<input type="checkbox"/> Files: Rename	
<input type="checkbox"/> Files: Replace	
<input type="checkbox"/> Files: Move	
<input type="checkbox"/> Files: Copy	
<input type="checkbox"/> Files: Delete	

Backend User Interface

Environment Overview

The current application context is now shown in the Environment module:
ADMIN TOOLS → **Environment** → **Environment Overview**.



The screenshot displays the TYPO3 Backend User Interface. The left sidebar shows the navigation menu with 'ADMIN TOOLS' expanded to 'Environment'. The main content area shows the 'Environment Overview' modal window. A red arrow points to the 'Application Context' field, which is set to 'Development'.

Environment Overview	
Operating system:	Unix
PHP version:	7.3.11-1-deb10u1
PHP CGI detected:	Yes
Database connection 'Default':	<ul style="list-style-type: none">• Database name: /var/www/html/docs TYPO3conf/cms-1a0c94fb.sqlite• Username:• Host:• Socket: PHP default• Version: sqlite• Tables: 48
Application Context:	Development

Backend User Interface

Feature Toggles

The visual appearance of feature toggles has been improved:

TYPO3 v9 LTS

Feature Toggles

Enable and disable certain core features. Feature toggles in the core are used to globally use new features of TYPO3 that may be activated on new installations but upgrading installations can still use the old behaviour.

Available features:

- fluidBasedPageModule (default on): Use the rewritten page layout backend module which is based on Fluid. Can be toggled off to use the legacy PageLayoutView on installations which require the hooks etc. which are associated with PageLayoutView
- form.legacyUploadMimeType (default off): If on, some mime types are predefined for the "FileUpload" and "ImageUpload" elements of the "form" extension which always allows file uploads of these types, no matter the specific form element definition.
- redirects.hitCount (default off): If on, and if extension "redirects" is loaded, each performed redirect is counted and last hit time is logged to the database.
- security.frontend.keepSessionDataOnLogout (default off): If on, session data is kept in an anonymous session after frontend user logged out. As this is a potential security risk, it is recommended to disable this option if not specifically needed.
- rearrangedRedirectMiddlewares (default off): If on, the middlewares "base redirect" and "redirects" are switched, so that "redirects" (if extended is installed) is executed before "base redirect". The new ordering aims to be a better default shipped by the TYPO3 core.
- login.extbase (default off): If activated, and if extension "login" is loaded, extbase based code will be used instead of pbase version

Save

TYPO3 v10 LTS

Feature Toggles

Enable and disable certain core features. Feature toggles in the core are used to globally use new features of TYPO3 that may be activated on new installations but upgrading installations can still use the old behaviour.

Available features:

- Fluid based page module
Use the rewritten page layout backend module which is based on Fluid. Can be toggled off to use the legacy PageLayoutView on installations which require the hooks etc. which are associated with PageLayoutView
Default setting: Enabled
- Form: legacy upload mime types
If on, some mime types are predefined for the "FileUpload" and "ImageUpload" elements of the "form" extension which always allows file uploads of these types, no matter the specific form element definition.
Default setting: Enabled

Default: hit count

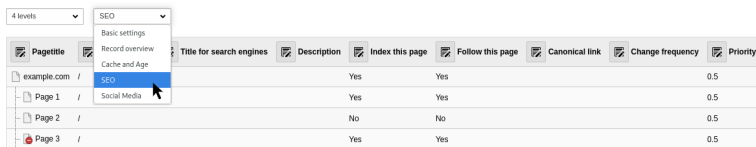
Save

Backend User Interface

Info Module

- SEO and Social Media details have been added to the Info module:
WEB → **Info** → **Pagetree Overview**.

Pagetree overview

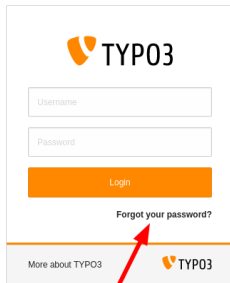


4 levels		SEO	Title for search engines	Description	Index this page	Follow this page	Canonical link	Change frequency	Priority
example.com /		Basic settings Record overview Cache and Age SEO Social Media			Yes	Yes			0.5
Page 1 /					Yes	Yes			0.5
Page 2 /					No	No			0.5
Page 3 /					Yes	Yes			0.5

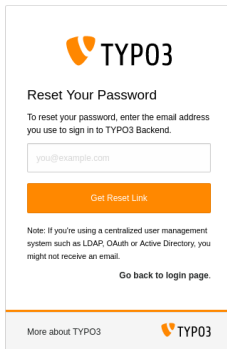
Backend User Interface

Password Reset for Backend Users

Backend users can now request a password recovery email to reset their access details.



The screenshot shows the TYPO3 login interface. At the top is the TYPO3 logo. Below it are two input fields: 'Username' and 'Password'. An orange 'Login' button is positioned below the fields. Underneath the button is the text 'Forgot your password?'. At the bottom of the page, there is a footer with 'More about TYPO3' and the TYPO3 logo. A red arrow points from the bottom of the page up to the 'Forgot your password?' link.



The screenshot shows the TYPO3 password reset interface. At the top is the TYPO3 logo. Below it is the heading 'Reset Your Password'. The text reads: 'To reset your password, enter the email address you use to sign in to TYPO3 Backend.' Below this is an input field containing the email address 'you@example.com'. An orange 'Get Reset Link' button is located below the input field. A note follows: 'Note: If you're using a centralized user management system such as LDAP, OAuth or Active Directory, you might not receive an email.' At the bottom of the note is the text 'Go back to login page.' The footer contains 'More about TYPO3' and the TYPO3 logo.

Backend User Interface

Password Reset for Backend Users

- Password resets for backend users are only valid for 4 hours. This time limit is not configurable.
- To strengthen security, the function can be disabled for admin users or for all users.
- If users share one email address, an alternative email text is used.
- TCA field `be_users.email` must not be set to `eval=email`.
- The function only works for users, who:
 - have an email address set,
 - have a password set, and
 - are not disabled/deleted.

Backend User Interface

Password Reset for Backend Users

- Password recovery emails can also be triggered on the command line.

```
typo3v10:/var/www/typo3v10$ ./bin/typo3 backend:resetpassword --help
Description:
  Trigger a password reset for a backend user.

Usage:
  backend:resetpassword <backendurl> <email>

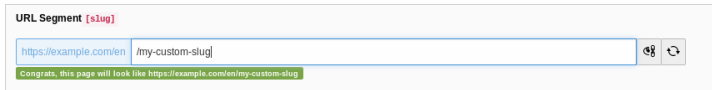
Arguments:
  backendurl      The URL of the TYPO3 Backend, e.g. https://www.example.com/typo3/
  email           The email address of a valid backend user

Options:
  -h, --help          Display this help message
  -q, --quiet         Do not output any message
  -V, --version       Display this application version
  --ansi             Force ANSI output
  --no-ansi          Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
```

Backend User Interface

Slug Updates and Redirects

- When backend users change the URL path of a page (the so-called "slug"), the old URL becomes unavailable.
- This possibly results in a "page not found" error for this page, including the URLs of all sub-pages.

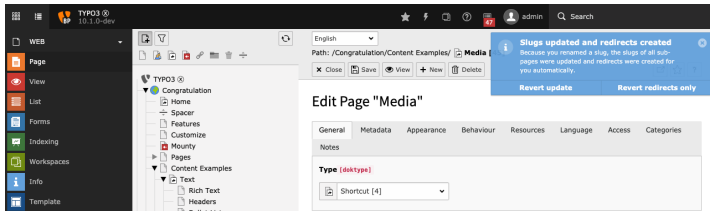


- Two actions prevent this from happening now:
 - slugs for all sub-pages are automatically updated
 - redirects from the old to the new URLs are created

Backend User Interface

Slug Updates and Redirects

- Backend users are informed about these actions and they can easily roll back the changes with a click of a button if required:



Form Framework

Creating and managing forms made easier

Form Framework

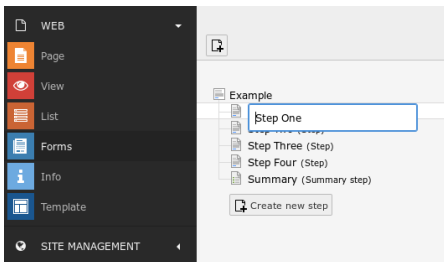
Multi-step Wizard

- A new JavaScript module `MultiStepWizard` has been introduced, that adds the following features:
 - Navigation to previous steps.
 - Steps support descriptive labels such as "Start" or "Finish", rather than the numerical indicator "Step x of y".
 - Optimized configuration structure.
- See [change log](#) for JavaScript code examples.
- This new features improves the user experience significantly: backend users will notice an enhanced form creation wizard.

Form Framework

Edit Titles

Labels of form elements can now be edited by double clicking on the title in the structure tree.



Form Framework

Form Setup

- Three files were used previously: `BaseSetup.yaml`, `FormEditorSetup.yaml`, and `FormEngineSetup.yaml`.
- This has been streamlined and consolidated into one file now: `FormSetup.yaml`.
- This file contains the basic setup including imports of the configuration for validators, form elements and finishers.
- All previously used inheritances and mixins have been resolved which makes it very easy to understand the entire configuration.

Form Framework

Translations

- The following option has been renamed:
translationFile → translationFiles
- The default translation files are now registered at index 10:
 - EXT:form/Resources/Private/Language/locallang.xlf
 - EXT:form/Resources/Private/Language/Database.xlf
- Custom form YAML configuration files need to be updated.

OLD:
translationFile: path/to/locallang.xlf

NEW:
translationFiles:
 20: path/to/locallang.xlf

Form Framework

YAML Files

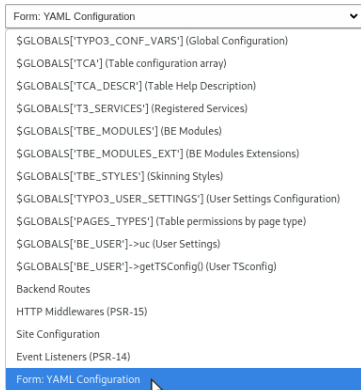
- YAML files now use the TYPO3 core YAML file loader.
- This enabled features such as:
 - Import of other YAML files via `imports` directive.
 - Replacement of `%placeholders%`.

Form Framework

YAML Configuration

If the system extension `EXT:form` is installed, the parsed YAML configuration can be displayed under **SYSTEM** → **Configuration**.

This also requires administrators to activate `EXT:lowlevel` of course.



Form Framework

Record Browser

- The record browser can now be configured to use custom tables:

```
TYPO3:
  CMS:
    Form:
      prototypes:
        standard:
          formElementsDefinition:
            MyCustomElement:
              formEditor:
                editors:
                  # ...
                  300:
                    identifier: myRecord
                    # ...
                    browsableType: tx_myext_mytable
                    propertyPath: properties.myRecordUid
                    # ...
```

Form Framework

Record Browser

- The button icon of the record browser are now configurable:

```
TYPO3:
  CMS:
    Form:
      prototypes:
        standard:
          formElementsDefinition:
            MyCustomElement:
              formEditor:
                editors:
                  # ...
                  300:
                    identifier: contentElement
                    # ...
                    browsableType: tt_content
                    iconIdentifier: mimetypes-x-content-text
                    propertyPath: properties.contentElementUid
                    # ...
```

Form Framework

Record Browser

- A new *RenderFormValue-ViewHelper* lets integrators/developers access single form values in templates:

```
<p>
  The following message was just sent by
  <formvh:renderFormValue renderable="{page.rootForm.elements.name}" as="formValue">
    {formValue.processedValue}
  </formvh:renderFormValue>:
</p>

<blockquote>
  <formvh:renderFormValue renderable="{page.rootForm.elements.message}" as="formValue">
    {formValue.processedValue}
  </formvh:renderFormValue>
</blockquote>
```

Form Framework

Fieldset Labels

- The section element `Fieldset` is now accessible in templates.
- By default this affects the **SummaryPage** form element as well as the **EmailToReceiver** and **EmailToSender** finishers.
- Typical use-case:
A form with a shipping and a billing address. Both sections could have a field with the same name, e.g. `street`. It is now possible to distinguish between both fields by using fieldset labels.

Form Framework

File Uploads

- Predefined `allowedMimeTypes` of the following form elements have been marked **deprecated**:
 - `FileUpload`
 - `ImageUpload`
- All valid MIME types must be explicitly listed in the form definition now (predefined MIME types will be removed in TYPO3 v11)

Form Framework

Form Mixins

- Mixins have been marked as **deprecated** and should not be used anymore.
- This affects all inheritances from `TYPO3\CMS\Form mixins.*`.
- Migration options:
 - Embed the essential parts from `TYPO3\CMS\Form mixins.*`, or
 - migrate them to custom mixins.

Form Framework

Multiple Recipients

- Mails sent by the *EmailFinisher* can now have multiple recipients.
- The following new options have been introduced:
 - `recipients` (To)
 - `replyToRecipients` (Reply-To)
 - `carbonCopyRecipients` (CC)
 - `blindCarbonCopyRecipients` (BCC)

Form Framework

Multiple Recipients

- This change requires a manual migration of single value options to their list value successors.

Old Finisher configuration:

```
finishers:  
-  
  identifier: EmailToReceiver  
  options:  
    recipientAddress: user@example.com  
    recipientName: 'Firstname Lastname'
```

New Finisher configuration:

```
finishers:  
-  
  identifier: EmailToReceiver  
  options:  
    recipients:  
      user@example.com: 'Firstname Lastname'
```

- See [change log](#) for more migration examples.

Form Framework

Plaintext/HTML

- Mails sent by the *EmailFinisher* can now feature both plaintext and/or HTML part.
- At the same time the option `format` has been marked deprecated and will be removed in TYPO3 v11.
- Existing values will be automatically migrated:
 - `format:html` → `addHtmlPart:true`
 - `format:plaintext` → `addHtmlPart:false`
 - a missing "format" → `addHtmlPart:true`
- Developers should be aware of the following two constants which have been marked deprecated:
 - `EmailFinisher::FORMAT_PLAINTEXT`
 - `EmailFinisher::FORMAT_HTML`

Form Framework

Sorting of Forms

- Forms can now be sorted in either ascending or descending order.
- Two new settings were introduced: `sortByKeys` and `sortAscending`.
- Forms are initially sorted by their name and their file UID (ascending).
- To change the sorting, the following configuration needs to be added in the YAML configuration file:

```
TYPO3:
  CMS:
    Form:
      persistenceManager:
        sortByKeys: ['name', 'fileUid']
        sortAscending: true
```

In-depth Changes




In-depth Changes

Improvements and new features for integrators and developers

In-depth Changes

Backend User Switch

- A log message is written if an admin user switches to another backend user:

E	Time	User	Type	Action	Details	Actions
	04:11:52	 fred (via admin) LIVE	LOGIN	LOGOUT	User fred logged out from TYPO3 Backend (msg#255.2.1)	
	04:11:48	 admin LIVE	LOGIN	LOGOUT	User admin switched to user fred (be_users:2) (msg#255.2.1)	
	04:08:38	 admin [-99]	LOGIN	LOGIN	User admin logged in from 172.72.72.1 (msg#255.1.1)	



In-depth Changes

TypoScript Changes

- TypoScript property `config.cache` now supports keyword "current" to refer to the current page. For example:

```
config.cache.all = fe_users:current
```

- The Page/User TSconfig setting `TSFE.constants` has been removed.
 - Include TypoScript conditions in `setup/constants` and use a proper configuration in file `ext_localconf.php`.
- The following two options to configure the size of popup windows have been removed:
 - `options.popupWindowSize`
 - `options.rte.popupWindowSize`

In-depth Changes

TypoScript Changes

- The database field `nextLevel` of the database table `sys_template` has been removed.
 - Replace the record (the UID is stored in the field `nextLevel`) with a condition to add TypoScript for subpages. For example: `[tree.level > 1]`
- The following values are **not allowed** anymore:
 - `typolink.addQueryString.method = POST`
 - `typolink.addQueryString.method = GET,POST`
 - `typolink.addQueryString.method = POST,GET`
 - Change the assignments in TypoScript, Fluid and PHP to GET.

In-depth Changes

Task Center and `EXT:sys_action`

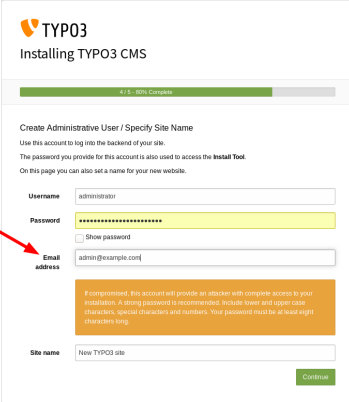
- The system extensions `EXT:taskcenter` and `EXT:sys_action` have been removed from the core.
- They are now available as separate extensions from the [TER](#) and at [GitHub](#).
- The Dashboard replaces the Task Center and `EXT:sys_action`.

In-depth Changes

Administrator Email Address

An email address can now be entered as part of the installation process. This address is used for the initial administrator backend user.

The same option exists in the Install Tool's Maintenance module **Create Administrative User**.



The screenshot shows the 'Installing TYPO3 CMS' screen. At the top, there is a progress bar indicating '47.5 - 80% Complete'. Below the progress bar, the heading is 'Create Administrative User / Specify Site Name'. Underneath, there are instructions: 'Use this account to log into the backend of your site. The password you provide for this account is also used to access the Install Tool. On this page you can also set a name for your new website.' The form contains several fields: 'Username' with the value 'administrator', 'Password' with a masked field and a 'Show password' checkbox, 'Email address' with the value 'admin@example.com', and 'Site name' with the value 'New TYPO3 site'. A red arrow points to the 'Email address' field. A warning box is present with the text: 'If compromised, this account will provide an attacker with complete access to your installation. A strong password is recommended, include lower and upper case characters, special characters and numbers. Your password must be at least eight characters long.' A 'Continue' button is located at the bottom right.

In-depth Changes

Caches

- Caching framework does not support the ApcBackend anymore
 - Use **APCu** instead - note the "u".

OLD:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['rootline']['backend'] =  
\TYPO3\CMS\Core\Cache\Backend\ApcBackend::class;
```

NOW:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['rootline']['backend'] =  
\TYPO3\CMS\Core\Cache\Backend\ApcuBackend::class;
```

- Extbase caches `extbase_reflection` and `extbase_datamapfactory_datamap` have been consolidated and are now available as a single cache named "extbase".

In-depth Changes

Cache Storage Type

- TYPO3 features a flexible caching system with a default configuration that is ideal for most use cases.
- The storage type can now be configured to fine-tune the caches and increase performance depending on the individual environment.
 - Choose the **database** storage for a standard environment or if a network file system (NFS) is used for example.
 - Choose the **file system** if a distributed database setup is used for example.
 - Choose **custom cache settings** to configure the storage type for each cache independently.
- For more complex installations, memory-based caches such as [Redis](#) or [Memcached](#) should be considered.

In-depth Changes

Cache Storage Type

Admin Tools ➡ Settings ➡ Configuration Presets ➡ Cache Settings:

▼ Cache settings

...

Prefer database storage for caching [Active]

Prefer File storage for caching

Custom configuration

Custom cache settings:

<code>SYSCaching/cacheConfigurations/hash/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYSCaching/cacheConfigurations/pages/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYSCaching/cacheConfigurations/pagesection/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYSCaching/cacheConfigurations/imagesizes/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>
<code>SYSCaching/cacheConfigurations/outline/backend</code>	<input type="text" value="TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend"/>

In-depth Changes

Cache Dependency Injection

- Extension developers are encouraged to inject caches directly rather than using the CacheManager.
- This requires a few simple changes as shown below.
- **Previously:**

```
class MyClass
{
    /**
     * @var TYPO3\CMS\Core\Cache\Frontend\FrontendInterface
     */
    private $cache;

    public function __construct()
    {
        $cacheManager = GeneralUtility::makeInstance(CacheManager::class);
        $this->cache = $cacheManager->getCache('my_cache');
    }
}
```

In-depth Changes

Cache Dependency Injection

- In **TYPO3 v10 LTS**, the class should look as follows:

```
class MyClass
{
    /**
     * @var TYPO3\CMS\Core\Cache\Frontend\FrontendInterface
     */
    private $cache;

    public function __construct(FrontendInterface $cache)
    {
        $this->cache = $cache;
    }
}
```


In-depth Changes

Cache Dependency Injection

- ...and the following container service configuration is required:

```
services:  
  cache.my_cache:  
    class: TYPO3\CMS\Core\Cache\Frontend\FrontendInterface  
    factory: ['@TYPO3\CMS\Core\Cache\CacheManager', 'getCache']  
    arguments: ['my_cache']  
  
MyClass:  
  arguments:  
    $cache: '@cache.my_cache'
```

In-depth Changes

Caching Framework

- The following caches have been renamed:

- `cache_core` → `core`
- `cache_hash` → `hash`
- `cache_pages` → `pages`
- `cache_pagesection` → `pagesection`
- `cache_runtime` → `runtime`
- `cache_rootline` → `rootline`
- `cache_imagesizes` → `imagesizes`

- New method to access the caches:

OLD:

```
$cacheManager->getCache('cache_core');
```

NEW:

```
$cacheManager->getCache('core');
```

- The prefix `cf_` has been removed from the database tables.

In-depth Changes

Conflicting Redirects

- A new Symfony command has been introduced to detect redirects that conflict with page URLs.
- Execute the command in the CLI:
(optional parameter `--site` limits the check to a specific site)

```
typo3v10:~/www$ ./bin/typo3 redirects:checkintegrity
Redirect (Host: *, Path: /page-1) conflicts with /page-1
typo3v10:~/www$
```

- The command is also available as a scheduler task:

	ID	Task	Type	Frequency	Parallel Execution	Last Execution	Next Execution	    
<input type="checkbox"/>	1	Execute console commands (scheduler) redirects:checkintegrity	Recurring	86400	No	-	15-09-19 06:12	

In-depth Changes

Conflicting Redirects

- A list of detected conflicting redirects can also be accessed in the Reports module:

Conflicting Redirects

1 conflicting redirects
These redirects cause a conflict as there are pages that are still accessible with the same URL.

- /page-1
Source Domain: *
Source Path: /page-1

- **Note:** The command needs to be executed again to "reset" the list. Solving the issue (e.g. by removing the redirect) does not clear the list.

In-depth Changes

Distribution Packages

- Distributions can now provide site configuration file(s).
- Create a directory/file in the distribution package as follows:
`Initialisation/Site/<siteIdentifier>/config.yaml`
- Similar to assets, which are moved to `fileadmin/`, site configurations are moved to the `config/` folder.
- If the target directory already exists, no change is made to the existing configuration.

In-depth Changes

Application Context in CLI

- The current Application Context is now shown next to the TYPO3 version number in CLI requests:

```
typo3v10:~/www$ ./bin/typo3 --version
TYPO3 CMS 10.1.0 (Application Context: Production)
typo3v10:~/www$ █
```

In-depth Changes

Vimeo Video Rendering

- The parameter `api=1` in Vimeo video URLs allows API interactions with the video player (e.g. adding buttons to control the video).
- Integrators can now set this parameter in two different ways.
 - Using TypeScript:

```
lib.contentElement.settings.media.additionalConfig.api = 1
```

- In Fluid using the Media-ViewHelper:

```
<f:media
  file="{file}"
  alt="{file.properties.alternative}"
  title="{file.properties.title}"
  additionalConfig="{api: 1}"
/>
```

In-depth Changes

File Uploads

- It is now possible to configure the default action when uploading files in the file list module using drag'n drop.
- User TSConfig:

```
# Set default to replace:
```

```
options.file_list.uploader.defaultAction = replace
```

```
# Set default to rename:
```

```
options.file_list.uploader.defaultAction = rename
```

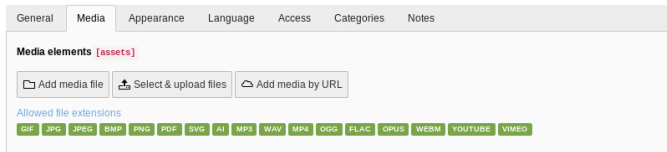
```
# Set default to cancel:
```

```
options.file_list.uploader.defaultAction = cancel
```


In-depth Changes

Media Element Buttons

- Buttons **"Add media by URL"** and **"Select & upload files"** can now be enabled/disabled independently from each other.



- The example below hides both buttons:

```
$GLOBALS['TCA']['pages']['columns']['media']['config']['appearance'] = [  
    'fileUploadAllowed' => false,  
    'fileByUrlAllowed' => false,  
];
```

In-depth Changes

Admin Panel

- The Admin Panel features a new panel **USER_INT** under the "Info" module.



The screenshot displays the Admin Panel interface. At the top, there are tabs for 'Info', 'General', 'PHP', 'REQUEST', and 'USER_INT'. The 'USER_INT' tab is selected. Below the tabs, the title 'USER_INT' is shown, followed by the identifier 'INT_SCRIPT.4c62089c320a4e49719267400a6131c7'. A table with two columns, 'Key' and 'Value', is displayed. The table contains two rows: one for 'TYPE' with the value '"FUNC"' and one for 'userFunc' with the value '"Vendor\\MyExtension\\ExampleTime->printTime"'. At the bottom of the panel, there is a status bar with icons for messages, info (load time: 94 ms), debug (0 errors/warnings), and settings.

Key	Value
TYPE	"FUNC"
userFunc	"Vendor\\MyExtension\\ExampleTime->printTime"

In-depth Changes

Site Configuration

- When a new page is created on the root level, a standard site configuration is automatically generated with it.
- As a result, a basic TYPO3 site can be set up quickly.
- The site configuration features:
 - a pre-defined identifier (e.g. autogenerated-1-c4ca4238a0)
 - an entry point (e.g. <https://example.com/autogenerated-1>)
 - a default language (e.g. English)

In-depth Changes

Site Configuration

- The site title can now be configured in **Site Configuration** → **Sites**.
- This lets integrators specify different site titles per language.
- The field in the template record is obsolete and has been marked as **deprecated**.
- The field `sys_template.sitetitle` (database and TCA) will be removed in TYPO3 v11.
- The site title is used for the page title as well as for future `schema.org` integrations.

In-depth Changes

Site Configuration

- It is now possible to use environment variables in imports of site configuration YAML files:

```
imports:  
  -  
    resource: 'Env_%env("foo")%.yaml'
```

In-depth Changes

Frontend Login

- TYPO3 v10 LTS now includes an Extbase-version of the frontend login functionality.
- This solution has a few advantages:
 - Modify the templates more easily.
 - Send out HTML-based password recovery emails.
 - Adjust and modify validators to enforce password restrictions.
- The new Extbase plugin is available out-of-the-box for new installations.
- Existing TYPO3 instances will continue to use the old templates.
- Integrators can switch between the "old" and the "new" plugin by using a feature toggle.

In-depth Changes

Frontend Login

- A password recovery form has been added as part of the Extbase plugin.
- Users can request a password change and will receive an email with a link which redirects them to the form.
- Default password validation rules:
 - `NotEmptyValidator` - passwords cannot be empty.
 - `StringLengthValidator` - passwords must have a minimum length.

In-depth Changes

Frontend Login

- These validation rules can be customized.
- For example:

```
plugin.tx_felogin_login {
    settings {
        passwordValidators {
            10 = TYPO3\CMS\Extbase\Validation\Validator\AlphanumericValidator
            20 {
                className = TYPO3\CMS\Extbase\Validation\Validator\StringLengthValidator
                options {
                    minimum = 12
                    maximum = 32
                }
            }
            30 = \Vendor\MyExtension\Validation\Validator\MyCustomPasswordPolicyValidator
        }
    }
}
```


In-depth Changes

Multiple Sitemaps

- It is now possible to configure multiple sitemaps.
- Syntax:

```
plugin.tx_seo {
  config {
    <sitemapType> {
      sitemaps {
        <unique key> {
          provider = TYPO3\CMS\Seo\XmlSitemap\RecordsXmlSitemapDataProvider
          config {
            ...
          }
        }
      }
    }
  }
}
```

In-depth Changes

HTML5 attribute `nomodule`

- The HTML5 attribute `nomodule` is now supported when including JavaScript files in TypoScript.

```
page.includeJSFooter.file = path/to/classic-file.js  
page.includeJSFooter.file.nomodule = 1
```

- This attribute prevents a script from being executed in browsers that support module scripts.
- Read more about the standard in the [specification](#) and about the concept of [modules](#).

In-depth Changes

Link Validator

- The Link Validator now supports additional configuration for external links.
- Values for `httpAgentUrl` and `httpAgentEmail` should be provided.
- Settings `headers`, `method` and `range` are advanced settings.

```
mod.linkvalidator {  
  linktypesConfig {  
    external {  
      httpAgentName = ...  
      httpAgentUrl = ...  
      httpAgentEmail = ...  
      headers {  
      }  
      method = HEAD  
      range = 0-4048  
    }  
  }  
}
```

In-depth Changes

Link Validator

- Link Validator now marks broken **external** links in the RTE too.
- This feature was only available for internal links.
- It is recommended to run the Link Validator as a Scheduler task to regularly crawl for broken links.

In-depth Changes

Localization Management Platform

- The SaaS solution "[Crowdin](#)" is now used as the localization/translation management platform for TYPO3.
- We encourage everyone to participate and improve the localization.
- Crowdin can be used to translate language labels of the TYPO3 core as well as of TYPO3 extensions.
- Read more about the initiative in [this article](#) and in the [TYPO3 documentation](#).



In-depth Changes

Fluid-based HTML Emails

- TYPO3 now supports sending template-based HTML and plain-text emails.
- Emails are built by using the Fluid templating engine.
- Email templates can be customized by overwriting the paths to the template files:

```
$GLOBALS['TYPO3_CONF_VARS']['MAIL']['templateRootPaths'][700] =  
    'EXT:my_site_extension/Resources/Private/Templates/Email';
```

```
$GLOBALS['TYPO3_CONF_VARS']['MAIL']['layoutRootPaths'][700] =  
    'EXT:my_site_extension/Resources/Private/Layouts';
```

In-depth Changes

Fluid-based HTML Emails

- Fluid-based templated emails are used for the following components for example:
 - Install Tool test email (see example on the next slide).
 - Workspace notification email on stage change.
 - Notification email on backend user login.

In-depth Changes

Fluid-based HTML Emails

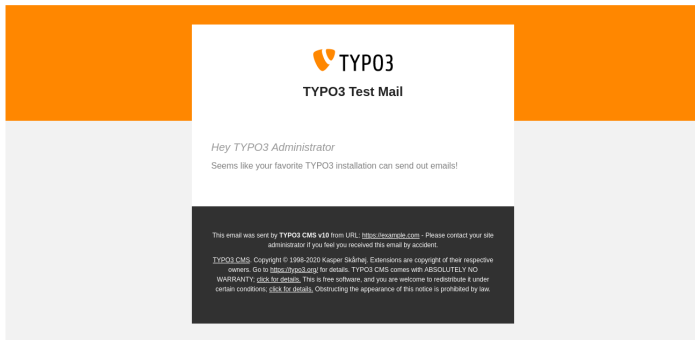
Test email sent from the Install Tool:

Test TYPO3 CMS mail delivery from site "New TYPO3 site"



From TYPO3 CMS install tool

[Details](#) [Plain text](#)



In-depth Changes

TypoScript

- The old `PIDupinRootline` condition has been re-implemented in TypoScript using the Symfony expression language.
- Old TypoScript condition syntax:

```
[PIDupinRootline = 30]
    page.10.value = I'm on any subpage of page with UID 30.
[END]
```

- New TypoScript condition syntax:

```
[30 in tree.rootLineParentIds]
    page.10.value = I'm on any subpage of page with UID 30.
[END]
```

In-depth Changes

Lazy Loading for Images

- The HTML attribute `loading` can now be set for ``-tags.
- Browsers which support this feature won't load these images until they are in the viewport.
- The behavior can be modified by the following TypoScript constant:

```
styles.content.image.lazyLoading = lazy
```

- Valid values are: `lazy` (default), `eager`, and `auto`.
- The Fluid `Image-ViewHelper` also supports lazy loading now:

```
<f:image src="{fileObject}" treatIdAsReference="true"  
  loading="lazy" />
```

In-depth Changes

Default values for `lockIP/lockIPv6`

- The default values for `lockIP` settings have been changed.
- The following four system variables are now **disabled** by default:
 - `[FE] ['lockIP']`
 - `[FE] ['lockIPv6']`
 - `[BE] ['lockIP']`
 - `[BE] ['lockIPv6']`
- The old default values ("4" for the backend and "2" for the frontend) caused problems for example for clients with IPv4 and IPv6 address support.

In-depth Changes

SEO: Sitemap.xml

- The default path to the file `Sitemap.xml` of the system extension `EXT:seo` can be customized now:

```
# Globally for all sitemaps:
```

```
plugin.tx_seo.config.xmlFile = EXT:myext/Resources/Public/CSS/mySite.xml
```

```
# For all sitemaps of a specific type:
```

```
plugin.tx_seo.config.<sitemapType>.sitemaps.xmlFile = EXT:myext/Resources/Public/CSS/mySite.xml
```

```
# For a specific sitemap:
```

```
plugin.tx_seo.config.<sitemapType>.sitemaps.<sitemap>.config.xmlFile =
```

```
EXT:myext/Resources/Public/CSS/mySite.xml
```

- The default path reads:

```
EXT:seo/Resources/Public/CSS/Sitemap.xml
```

In-depth Changes

Reference Index

- Progress bars are shown for each database table during Reference Index update.

```
typo3v10:~/www$ ./bin/typo3 referenceindex:update -c
Reference Index being TESTED (nothing written, remove the "--check" argument)
-----
Update index of table be_groups
-----
1/1 [ ██████████ ] 100%
Update index of table pages
-----
1/1 [ ██████████ ] 82%
```

In-depth Changes

Scheduler

- Multiple tasks can be executed when using the option `--task`

```
typo3v10:~/www$ ./bin/typo3 scheduler:run --task 1 --task 2
```

- Verbose output can be enabled by `-v` and `-vv`

```
typo3v10:~/www$ ./bin/typo3 scheduler:run -vv
```

In-depth Changes

Page Type Handling

- TYPO3's internal handling of page types has changed.
- The option `pages.doktype` defines a numeric value that represents the type, e.g. standard page, folder, shortcut, link to external URL, etc.
- Pages of certain types (e.g. folder and recycler) were excluded when content was read from a specific page or records retrieved.
- This limitation has now been removed and custom page doktypes with a number >200 are now possible.
- Integrators and developers who have used page doktypes, e.g. in TypoScript, are advised to check if the previous behavior was misused and requires an update now.

In-depth Changes

Workspaces

- A new expression language variable has been added: `workspace`.
- This variable can be used to match a given expression against common workspace parameters.
- Currently, the following parameters are supported:
`workspaceId`, `isLive`, and `isOffline`.
- For example:

```
[workspace.workspaceId === 3]
  # Current workspace ID is 3
[end]
```


In-depth Changes

Workspaces

- For many years, the TYPO3 core set `pid` to `-1` of unpublished records.
- TYPO3 now handles versioned records by validating the following three fields:
 - `t3ver_wsuid` (the workspace ID the record is versioned in)
 - `t3ver_state` (the type of the versioned record)
 - `t3ver_oid` (the live version of a record)
- Therefore, `pid=-1` is not required anymore.
- The Upgrade Wizard converts all `pid` fields of versioned records into the real `pid` values.
- New installations are not affected by this change.

In-depth Changes

Runtime-Activated Packages

- The following global configuration option has been marked **deprecated**:
`$GLOBALS['TYPO3_CONF_VARS']['EXT']['runtimeActivatedPackages']`
- The use of runtime-activated extensions slows down a TYPO3 instance significantly.
- Integrators are advised to take necessary steps, if such warnings appear in the deprecation log:

Support for runtime activated packages will be removed in TYPO3 v11.0.

In-depth Changes

Page Not Found Handling

- The following global TYPO3 settings have been removed:
 - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageNotFound_handling']`
 - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageNotFound_handling_statheader']`
 - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageNotFound_handling_accessdeniedheader']`
 - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageUnavailable_handling']`
 - `$GLOBALS['TYPO3_CONF_VARS']['FE']['pageUnavailable_handling_statheader']`
- The Site Handling introduced in TYPO3 v9 replaces these settings.

In-depth Changes

Page Not Found Handling

- The message "*Page is being generated*" and the corresponding temporary HTTP 503 response have been removed.

Page is being generated.

If this message does not disappear within 30 seconds, please reload.

- Instead of offloading the work to wait for the final page content, concurrent requests now wait for the real page content to be rendered.

In-depth Changes

Frontend Login: Extbase

- The frontend user login (`EXT:felogin`) has been converted to Extbase and Fluid.
- The following changes have been implemented:
 - ❶ Prefix `"ll_"` has been removed from locallang keys.
 - Update your TypoScript if you have overwritten language labels and remove the prefix `"ll_"` from your keys.
 - ❷ Existing FlexForm structure has been reworked.
 - Execute the Upgrade Wizard to migrate the FlexForm values.

In-depth Changes

Languages

- ISO Codes:
 - The unused database field `static_lang_isocode` has been removed.
 - `EXT:static_info_tables` can be installed to reimplement functionality if required.
 - Developers are advised to fetch all metadata for a language using the Site Configuration and the SiteLanguage API instead.
- Language Files:
 - Usage of the global array `$GLOBALS[LOCAL_LANG]` has been deprecated.
 - The 2nd and 3rd arguments of `LanguageService->includeLLFile()` have been deprecated.

In-depth Changes

New Mail API

- SwiftMailer has been superseded by more modern libraries:
 - `symfony/mime` for creating mail messages
 - `symfony/mailer` for sending emails
- PHP function `mail()` is no longer supported.
 - It is recommended to switch to `sendmail` or `smtp` instead.
- Custom SwiftMailer plugins or transports require a migration.
- See the [Symfony Documentation](#) for further details how to leverage the new Mail API capabilities.

In-depth Changes

PSR Standards

- TYPO3 v10 LTS follows these [PSR standards](#):
 - PSR-0 / PSR-4 Autoloading
 - PSR-1 / PSR-2 Coding Standards
 - PSR-3 Logging
 - PSR-7 / PSR-15 / PSR-17 HTTP Request / Response handling)
 - PSR-11 Dependency Injection (Service Container)
 - PSR-14 Event Dispatcher
 - PSR-18 HTTP Client

In-depth Changes

PSR-3 Logging Interface

- TYPO3's Logging Framework (in particular LogLevel and LogManager) now uses the [PSR-3 Logger Interface](#).
- PSR-3 is a standardized method that allows libraries to receive a `Psr\Log\LoggerInterface` object and to write logs to it in a simple and universal way.
- This lets developers use custom loggers and to interact with other logging systems.

In-depth Changes

PSR-11 Symfony's DependencyInjection

- The package `symfony/dependency-injection` has been integrated and is used to manage system-wide dependency management and dependency injection for classes.
- This approach aims to replace the Extbase dependency injection container and object manager.
- Therefore, classes should be adjusted and avoid (whenever possible):
 - `\TYPO3\CMS\Extbase\Object\ObjectManager`
 - `\TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance()`

In-depth Changes

PSR-11 Symfony's DependencyInjection

- Configuration options include:
 - Autowiring (see example below)
 - Manual wiring (see [change log](#))
 - Advanced functionality (see [change log](#))

```
# Configuration/Services.yaml
services:
  _defaults:
    autowire: true
    autoconfigure: true
    public: false

Your\Namespace\:
  resource: '../Classes/*'
```

- See [Symfony documentation](#) for further details.

In-depth Changes

PSR-14 Event Dispatching

- A new "EventDispatcher" system has been added which aims to replace the hooks and Signal/Slots concepts.
- It is based on the [PSR-14 standard](#) which allows developers to inject logic into an application easily and consistently.
- PSR-14 consists of the following four components:
 - An **EventDispatcher** object that is used to trigger an event.
 - A **ListenerProvider** object that contains registered listeners for all events.
 - One or multiple **Event** objects which are called from the TYPO3 core or extensions ("Emitter").
 - One or multiple **Listeners** (usually in extensions and PHP packages) that are registered.

In-depth Changes

PSR-14 Event Dispatching

Implementation example

- 1 Add `event.listener` tag to the file `Configuration/Services.yaml`:

```
services:
  Vendor\Example\EventListener\NullMailer:
    tags:
      - { name: event.listener, identifier: 'myListener', event: TYPO3\CMS\Core\Mail\Event\
        AfterMailerInitializationEvent, before: 'redirects, anotherIdentifier' }
```

- 2 Implement your event object:

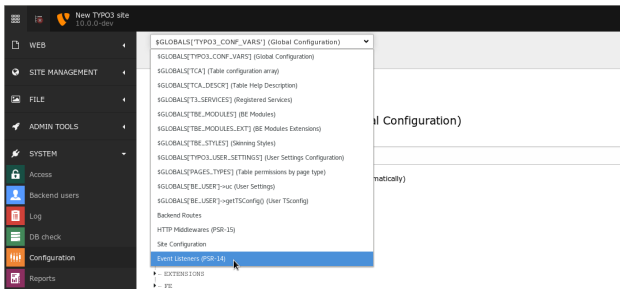
```
namespace Vendor\Example\EventListener;

class NullMailer
{
    public function __invoke(AfterMailerInitializationEvent $event): void
    {
        $event->getMailer()->injectMailSettings(['transport' => 'null']);
    }
}
```

In-depth Changes

PSR-14 Event Dispatching

- List of available Event Listeners can be accessed in the backend:
(requires system extension EXT:lowlevel)



In-depth Changes

PSR-14 Event Dispatching

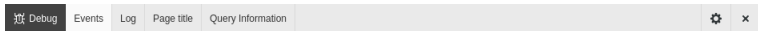
- Best practices:
 - Add only one Listener per PHP class and use `__invoke()` as the method name.
 - Add "Event" suffix to the class name when creating a new Event PHP class.
 - Move the Event PHP class file to an appropriate folder e.g. `Classes/Database/Event`.
 - Use dependency injection in form of a constructor argument to receive the `EventDispatcher` object if possible.
- Additional note:

Events provided by the TYPO3 core follow TYPO3's deprecation policy, except for its constructor arguments which may vary.

In-depth Changes

PSR-14 Events in Admin Panel

- The Admin Panel shows all PSR-14 events that have been dispatched in the current request.



In-depth Changes

PSR-17 HTTP Message Factories

- The [PSR-17](#) HTTP Message Factories implementation has been added.
- HTTP Message Factory interfaces should be used as dependencies for request handlers or services that create PSR-7 message objects.
- PSR-17 consists of six factory interfaces:
 - `\Psr\Http\Message\RequestFactoryInterface`
 - `\Psr\Http\Message\ResponseFactoryInterface`
 - `\Psr\Http\Message\ServerRequestFactoryInterface`
 - `\Psr\Http\Message\StreamFactoryInterface`
 - `\Psr\Http\Message\UploadedFileFactoryInterface`
 - `\Psr\Http\Message\UriFactoryInterface`
- See [documentation](#) for a code example.

In-depth Changes

PSR-18 HTTP Client

- The [PSR-18](#) HTTP Client implementation has been added.
- It lets developers generate HTTP requests based on PSR-7 message objects without relying on a specific HTTP client implementation.
- It does not replace the existing [Guzzle](#) wrapper, but provides a more generic alternative.
- PSR-18 consists of a client interface and three exception interfaces:
 - `\Psr\Http\Client\ClientInterface`
 - `\Psr\Http\Client\ClientExceptionInterface`
 - `\Psr\Http\Client\NetworkExceptionInterface`
 - `\Psr\Http\Client\RequestExceptionInterface`
- See [documentation](#) for a code example.

In-depth Changes

JavaScript Options and Functions

- The following JavaScript files have been removed:
 - `jsfunc.inline.js`
 - `jsfunc.evalfield.js`
 - Use `TYPO3\CMS\Backend\FormEngineValidation` instead.
- Additional submit handlers could previously be added by the option `additionalJavaScriptSubmit`. This option has been removed.
 - Create and register an AMD module instead.
- The global JavaScript function `top.openUrlInWindow()` has been marked deprecated.

In-depth Changes

JavaScript Options and Functions

- The global object `TBE_EDITOR.typo3form` and its backward layers `typo3FormFieldSet` and `typo3FormFieldGet` have been removed.
- File `md5.js` has been marked deprecated.
 - Load the AMD module `TYPO3\CMS\Backend\Hashing\Md5` via RequireJS instead.
- The following global JavaScript functions have been marked deprecated:
 - `top.rawurlencode()`
 - `top.str_replace()`
- Module `TYPO3\CMS\Backend.SplitButtons` has been deprecated.
 - Use `TYPO3\CMS\Backend\DocumentSaveActions` instead.

In-depth Changes

UTF-8-based Domains

- PHP has native functions to convert domains from UTF-8 into IDNA ASCII form (“punicode”), for example `idn_to_ascii()`.
- These can be used directly if the PHP extension “`intl`” is installed.
- If the PHP extension is not installed, the package `symfony/polyfill-intl-idn` provides the functions now.
- Previously, the package `algo26-matthias/idna-convert` was used which has been removed now.

In-depth Changes

BitSet Class

- New class has been introduced to efficiently handle boolean flags:
TYPO3\CMS\Core\Type\BitSet
- For example:

```
define('PERMISSIONS_NONE', 0b0); // 0
define('PERMISSIONS_PAGE_SHOW', 0b1); // 1
define('PERMISSIONS_PAGE_EDIT', 0b10); // 2
define('PERMISSIONS_PAGE_DELETE', 0b100); // 4
define('PERMISSIONS_PAGE_NEW', 0b1000); // 8
define('PERMISSIONS_CONTENT_EDIT', 0b10000); // 16
define('PERMISSIONS_ALL', 0b11111); // 31

$bitSet = new \TYPO3\CMS\Core\Type\BitSet(PERMISSIONS_PAGE_SHOW | PERMISSIONS_PAGE_NEW);
$bitSet->get(PERMISSIONS_PAGE_SHOW); // true
$bitSet->get(PERMISSIONS_CONTENT_EDIT); // false
```

In-depth Changes

Request Handler

- The following internal interface has been removed in favor of PSR-15 request handler and middleware interfaces:

`TYPO3\CMS\Core\Http\RequestHandlerInterface`

In-depth Changes

Request Handler

- The configuration of Extbase request handlers is no longer possible with TYPOScript.

Old method in TYPOScript:

```
config.tx_extbase {
    mvc {
        requestHandlers {
            Vendor\Example\Mvc\Web\FrontendRequestHandler = Vendor\Example\Mvc\Web\
                FrontendRequestHandler
        }
    }
}
```

New method in file `Configuration/Extbase/RequestHandlers.php`:

```
<?php
declare(strict_types = 1);

return [
    \Vendor\Example\Mvc\Web\FrontendRequestHandler::class,
];
```


In-depth Changes

Extbase and Fluid

- Registering plugins/modules require fully-qualified class names now
 - `\TYPO3\CMS\Extbase\Utility\ExtensionUtility::configurePlugin()`
 - `\TYPO3\CMS\Extbase\Utility\ExtensionUtility::registerModule()`
- Also omit vendor name in the extension name (first argument).
 - Use "ExampleBlog" instead of "Vendor.ExampleBlog".
- For example:

```
\TYPO3\CMS\Extbase\Utility\ExtensionUtility::configurePlugin(  
    'ExampleBlog', // previously: 'Vendor.ExampleBlog'  
    'pi1',  
    [  
        \Vendor\Example\Controller\BlogController::class => 'list,update,delete'  
    ],  
    [  
        \Vendor\Example\Controller\BlogController::class => 'list,update,delete'  
    ]  
);
```

In-depth Changes

Extbase and Fluid

- Property `extensionName` of `AbstractController` has been removed.
 - Use `\TYPO3\CMS\Extbase\Mvc\Request::getControllerExtensionName()` instead.

In-depth Changes

Extbase and Fluid

- Extbase models now support non fully-qualified class names in DocBlocks.

```
use TYPO3\CMS\Extbase\Persistence\ObjectStorage;
use ExtbaseTeam\BlogExample\Domain\Model\Comment;
```

```
class Post
{
    /**
     * @var ObjectStorage<Comment>
     */
    public $comments;
}
```

In-depth Changes

Extbase and Fluid

- Validators are not registered automatically in Extbase anymore.
- For a model named `Vendor\Example\Domain\Model\Blog`, Extbase automatically used the validator `Vendor\Example\Domain\Validator\BlogValidator`
- Validators need to be registered manually now:

```
use Vendor\Example\Domain\Model\Blog;
use TYPO3\CMS\Extbase\Annotation as Extbase;
use TYPO3\CMS\Extbase\Mvc\Controller\ActionController;

class BlogController extends ActionController
{
    /**
     * @Extbase\Validate(param="blog", validator="Vendor\Example\Domain\Validator\BlogValidator")
     */
    public function showAction(Blog $blog)
    {
        // ...
    }
}
```

In-depth Changes

Extbase and Fluid

- Class files now feature the "strict types" mode and type hints for scalars

```
<?php  
declare(strict_types=1);
```

- This results in fatal PHP errors if the method signatures in custom extensions are not compatible with the interfaces and/or parent classes.
- See [forge #87594](#) for a complete list of files and their changes.
- This task is still work in progress and further changes will be made.

In-depth Changes

Extbase and Fluid

- Omit the vendor name when registering plugins with
`\TYPO3\CMS\Extbase\Utility\ExtensionUtility::registerPlugin()`
- For example, use "Form" instead of "TYPO3\CMS\Form"
(first argument)

```
\TYPO3\CMS\Extbase\Utility\ExtensionUtility::registerPlugin(  
    'Form',  
    'Formframework',  
    'Form',  
    'content-form',  
);
```

In-depth Changes

Extbase and Fluid

- The following PSR-14-based events have been introduced for Extbase-related signals:

```
TYPO3\CMS\Extbase\Event\Mvc\AfterRequestDispatchedEvent
TYPO3\CMS\Extbase\Event\Mvc\BeforeActionCallEvent
TYPO3\CMS\Extbase\Event\Persistence\AfterObjectThawedEvent
TYPO3\CMS\Extbase\Event\Persistence\ModifyQueryBeforeFetchingObjectDataEvent
TYPO3\CMS\Extbase\Event\Persistence\ModifyResultAfterFetchingObjectDataEvent
TYPO3\CMS\Extbase\Event\Persistence\EntityAddedToPersistenceEvent
TYPO3\CMS\Extbase\Event\Persistence\EntityFinalizedAfterPersistenceEvent
TYPO3\CMS\Extbase\Event\Persistence\EntityUpdatedInPersistenceEvent
TYPO3\CMS\Extbase\Event\Persistence\EntityRemovedFromPersistenceEvent
TYPO3\CMS\Extbase\Event\Persistence\EntityPersistedEvent
```

- Existing signals have been replaced and should not be used anymore.

In-depth Changes

Extbase and Fluid - Class Mapping

- Persistence related class mapping using TypoScript is no longer supported:

```
config.tx_example_blog {
  persistence {
    classes {
      Vendor\Example\Domain\Model\Author {
        mapping {
          tableName = fe_users
          columns.name.mapOnProperty = fullname
        }
      }
    }
  }
}
```


In-depth Changes

Extbase and Fluid - Class Mapping

- The mapping needs to be implemented in a PHP file
Configuration/Extbase/Persistence/Classes.php:

```
<?php
declare(strict_types = 1);

return [
    \Vendor\Example\Domain\Model\Author::class => [
        'tableName' => 'fe_users',
        'properties' => [
            'fullname' => [
                'fieldName' => 'name'
            ]
        ]
    ]
];
```

- Note that property name and DB field have swapped!
Previously: <db-field>.mapOnProperty = <property>
New: properties.<property>.fieldname = <db-field>

In-depth Changes

cHash in UriBuilder and ViewHelpers

- The following two Extbase UriBuilder methods have been deprecated:
 - `UriBuilder->setUseCacheHash()`
 - `UriBuilder->getUseCacheHash()`
- This also impacts a number of Fluid ViewHelpers:
 - `f:form`
 - `f:link.action`
 - `f:link.page`
 - `f:link.typolink`
 - `f:uri.action`
 - `f:uri.page`
 - `f:uri.typolink`
 - `f:widget.link`
 - `f:widget.uri`
- ...as well as the TypoLink option "useCacheHash".

In-depth Changes

Lazy Loading Proxy

- A method `getUid()` has been added to the class `TYPO3\CMS\Extbase\Persistence\Generic\LazyLoadingProxy`.
- This allows developers to fetch the UID of the proxied object without fetching the object from the database.

In-depth Changes

ViewHelper editRecord

- An optional argument `fields` has been added to the `uri.editRecord` and `link.editRecord` ViewHelpers.
- If set, the FormEngine creates a form to only edit the given database field(s).
- The following example creates a link to edit the `tt_content.bodytext` field of record with the UID 42.

```
<be:link.editRecord uid="42" table="tt_content" fields="bodytext" returnUrl="foo/bar">  
  Edit record  
</be:link.editRecord>
```

In-depth Changes

AssetCollector

- The initial steps of integrating an AssetCollector have been implemented.
- The concept allows developers to add custom CSS/JS code (inline or external) multiple times, but TYPO3 outputs it only once.
- In this regards, two new Fluid ViewHelpers have been added:
 - `<f:asset.css>`
 - `<f:asset.script>`
- In the long run, the AssetCollector aims to replace the various existing TypoScript options that are rather confusing.

In-depth Changes

Fluid Templating Engine

- The TYPO3 core is fully compatible with Fluid version 2.6+ and 3.0+
- New installations without a dependency set will download and install Fluid version 3.x (`typo3fluid/fluid:^3`).
- If your project contains Fluid templates which are incompatible with version 3.0+, take one of the following actions:
 - Limit the max version: `typo3fluid/fluid:^2`
 - Update your Fluid templates.

In-depth Changes

Fluid Templating Engine

- The removal of white spaces in the default layout file of `EXT:fluid_styled_content` caused occasional issues and has been removed.

In-depth Changes

TCA Changes

- The following TCA options have been removed:
 - `$TCA[$tableName]['ctrl']['selicon_field_path']`
 - `$TCA[$tableName]['ctrl']['setDefaultOnCopy']`
 - When copying records, a `DataHandler` should be used to reset fields.
- The entire TCA of `sys_history` has been removed and the database field `pid` dropped. Accessing `$GLOBALS['TCA']['sys_history']` now triggers a PHP warning.

In-depth Changes

User Authentication Classes/Services

- The following abstract class has been restructured:
`TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`
- This also includes the following two direct sub-classes:
 - `BackendUserAuthentication`
 - `FrontendUserAuthentication`
- This change affects the properties:
 - `sessionTimeout`
 - `gc_time`
 - `sessionDataLifetime`
 - `loginType`

In-depth Changes

User Authentication Classes/Services

- The following class does not inherit from `AbstractService` anymore:
`\TYPO3\CMS\Core\Authentication\AbstractAuthenticationService`
- This possibly affects some of the hooks and custom authentication providers available.
- Developers are advised to review their custom authentication services and update their code if required.

In-depth Changes

Filelist Controllers

- The following controllers have been moved to `EXT:filelist`:
 - `CreateFolderController`
 - `EditFileController`
 - `FileUploadController`
 - `RenameFileController`
 - `ReplaceFileController`
- As a result, their namespace changed to `\TYPO3\CMS\Filelist\Controller\File`

Note: Use the TYPO3 FAL as API and add your own functionality with your own controller instead of reusing the **internal** controllers listed above.

In-depth Changes

Frontend Preview URL

- The following static method has been marked deprecated:
`\TYPO3\CMS\Backend\Utility\BackendUtility::getViewDomain()`
- Substitute the method by directly detecting a site based on a given page ID in the TYPO3 backend.
- For example:

```
$pageId = 123;  
$site = GeneralUtility::makeInstance(SiteFinder::class)->getSiteByPageId($pageId);  
$url = $site->getRouter()->generateUri($pageId, ['type' => 13]);
```

In-depth Changes

Frontend Request Workflow

- The Frontend Request Workflow has been reworked significantly.
- Components involved are all built using PSR-15 middlewares, the PSR-15 Request Handler, and the global TypoScriptFrontendController (TSFE) since TYPO3 v9.
- This impacts custom code, if the following hook and a frontend session is used:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['tslib/class.tslib_fe.php']['hook_eofe']
```

- Use a PSR-15 middleware instead of a hook, or explicitly call `storeSessionData` within the hook in PHP.

In-depth Changes

Frontend Request Workflow

- The following global variables have been removed:
 - `$GLOBALS['TYPO3_MISC']['microtime_start']`
 - `$GLOBALS['TYPO3_MISC']['microtime_end']`
 - `$GLOBALS['TYPO3_MISC']['microtime_BE_USER_start']`
 - `$GLOBALS['TYPO3_MISC']['microtime_BE_USER_end']`
- The TYPO3 core used them in the Admin Panel and HTTP header for example.
 - Use `TimeTracker->finish()` instead.

In-depth Changes

Locales

- Method `Locales::initialize()` has been marked deprecated.
 - Use `GeneralUtility::makeInstance(Locales::class)` or dependency injection to fetch an instance of the class `Locales` instead.
- Functionality of the following method has been marked deprecated:
`TypoScriptFrontendController->settingLocale()`.
 - Function is now available as `Locales::setSystemLocaleFromSiteLanguage()`.

In-depth Changes

Locales

- Public property

`TypoScriptFrontendController->sys_language_isocode` has been marked deprecated.

- Access the property via `SiteLanguage->getTwoLetterIsoCode()` and `sitelanguage:twoLetterIsoCode` instead.

In-depth Changes

Frontend Track User

- This following public properties of class

`\TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`
have been removed:

- `AbstractUserAuthentication->get_name`
- `AbstractUserAuthentication->getFallback`
- `AbstractUserAuthentication->getMethodEnabled`
- `AbstractUserAuthentication->get_URL_ID`

- Also the property `getMethodUrlIdToken` of class

`\TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController`.

- And the TypoScript setting `config.ftu`, as well as the global configuration `$GLOBALS['TYPO3_CONF_VARS']['FE']['get_url_id_token']`.

In-depth Changes

Constructor Injection in DataMapper

- The following class now uses constructor injection rather than setter injection: `\TYPO3\CMS\Extbase\Persistence\Generic\Mapper\DataMapper`
 - Avoid `GeneralUtility::makeInstance()` and `ObjectManager->get()`.
 - Use dependency injection instead (preferably constructor injection).

In-depth Changes

Context API

- The Context API features a new aspect "frontend.preview" that can be used to determine if the frontend is in preview mode:

```
GeneralUtility::makeInstance(Context::class)
->getPropertyFromAspect('frontend.preview', 'isPreview');
```

- This aspect replaces the following property which is marked deprecated now: `TypoScriptFrontendController->fePreview`

In-depth Changes

Context API

- Another new aspect `TypoScriptAspect` can be used to manipulate/check if `TemplateRendering` is forced.
- The setting `forceTemplateParsing` (TSFE and `TemplateService`) has been deprecated. The Context API should be used instead:

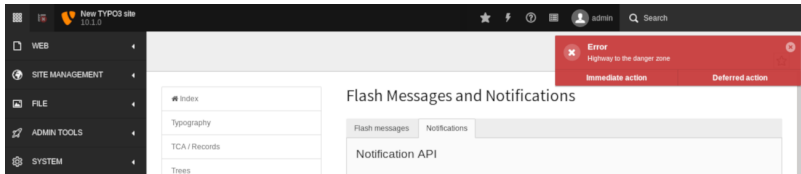
```
GeneralUtility::makeInstance(Context::class)
    ->getPropertyFromAspect('typoscript', 'forcedTemplateParsing');

$content->setAspect(
    'typoscript',
    GeneralUtility::makeInstance(TypoScriptAspect::class, true)
);
```

In-depth Changes

Notification Actions

- JavaScript notifications in the backend support actions (buttons) now.



In-depth Changes

Broadcast Channels and Messaging

- It is now possible to send and receive "broadcast messages" using JavaScript.

The API is considered **internal** for the time being and may change at any time until declared "stable".

- Example for **sending** a message:

```
require(['TYPO3\CMS\Backend/BroadcastService'], function (BroadcastService) {
  const payload = {
    componentName: 'my_extension',
    eventName: 'my_event',
    foo: 'bar'
  };
  BroadcastService.post(payload);
});
```

In-depth Changes

Broadcast Channels and Messaging

- Example for **receiving** the message:

```
define([], function() {
  document.addEventListener('typo3:my_component:my_event', (e) => eventHandler(e.detail));
  function eventHandler(detail) {
    // output contains key 'foo' as the payload
    console.log(detail);
  }
});
```

- See developer.mozilla.org for more details.

In-depth Changes

RequestFactory Middleware Handler

- It is now possible to define custom middleware handlers as an array.
- The RequestFactory builds a handler stack based on the `$GLOBALS['TYPO3_CONF_VARS']['HTTP']['handler']` array and injects it into the client.
- For example:

```
use \TYPO3\CMS\Core\Utility\GeneralUtility;
use \Vendor\MyExtension\Middleware\Guzzle\CustomMiddleware;
use \Vendor\MyExtension\Middleware\Guzzle\SecondCustomMiddleware;

# Add custom middleware to default Guzzle handler stack
$GLOBALS['TYPO3_CONF_VARS']['HTTP']['handler'][] =
    (GeneralUtility::makeInstance(CustomMiddleware::class))->handler();
$GLOBALS['TYPO3_CONF_VARS']['HTTP']['handler'][] =
    (GeneralUtility::makeInstance(SecondCustomMiddleware::class))->handler();
```


In-depth Changes

Custom File Processors

- Developers can now register their own file processors.
- Add the following code to file `ext_localconf.php`:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['fal']['processors']['ExampleImageProcessor'] = [  
    'className' => \Vendor\MyExtension\Resource\Processing\ExampleImageProcessor::class,  
    'before' => 'LocalImageProcessor',  
];
```

- Typical use cases:
 - add a watermark to images
 - compress uploaded files to a ZIP archive
 - store manipulated copies of images
 - etc.

In-depth Changes

Widget ViewHelpers

- Widget ViewHelpers set a session cookie in the frontend under certain circumstances.
- As this is not always desired (for example due to GDPR), this can be controlled now.
- A boolean `storeSession` has been introduced that lets developers enable/disable this feature.

```
<f:widget.autocomplete
  for="name"
  objects="{posts}"
  searchProperty="author"
  storeSession="false" />
```

In-depth Changes

PSR-14 Events in FAL

- Approximately 40 new [PSR-14](#) based Events have been introduced in the File Abstraction Layer (FAL).
- They replace existing Extbase Signal/Slots.
- Using the Signals continues to work (without producing any deprecation message!). However, the Signals in the FAL will likely be removed in TYPO3 v11.
- Extension authors are advised to migrate their code and use Events.
- Review the new PHP classes to learn more about PSR-14.

In-depth Changes

PSR-14 Events in the TYPO3 Core

- A number of new PSR-14 Events replace Signal/Slots in the TYPO3 core:
 - `TYPO3\CMS\Core\Imaging\Event\ModifyIconForResourcePropertiesEvent`
 - `TYPO3\CMS\Core\DataHandling\Event\IsTableExcludedFromReferenceIndexEvent`
 - `TYPO3\CMS\Core\DataHandling\Event\AppendLinkHandlerElementsEvent`
 - `TYPO3\CMS\Core\Configuration\Event\AfterTcaCompilationEvent`
 - `TYPO3\CMS\Core\Database\Event\AlterTableDefinitionStatementsEvent`
 - `TYPO3\CMS\Core\Tree\Event\ModifyTreeDataEvent`
 - `TYPO3\CMS\Backend\Backend\Event\SystemInformationToolbarCollectorEvent`

In-depth Changes

Tsconfig Parsing

- Two new PHP classes have been introduced to load and parse PageTsConfig:
 - TYPO3\CMS\Core\Configuration\Loader\PageTsConfigLoader
 - TYPO3\CMS\Core\Configuration\Parser\PageTsConfigParser
- For example:

```
// Fetch all available PageTS of a page/rootline:
$loader = GeneralUtility::makeInstance(PageTsConfigLoader::class);
$tsConfigString = $loader->load($rootLine);

// Parse the string and apply conditions:
$parser = GeneralUtility::makeInstance(
    PageTsConfigParser::class, $typoScriptParser, $hashCache
);

$pagesTsconfig = $parser->parse($tsConfigString, $conditionMatcher);
```

In-depth Changes

Prepared Statements

- The `pdo_mysql` driver uses prepared statements by default now.
- In older versions of TYPO3, *emulated prepared statements* were used. This means, all returned values of a query were strings.
- This behavior has changed and prepared statements are used which return native data types.
- For example: values of a column defined as integer are returned in PHP as `int`.
- This feature can be deactivated by setting the option `PDO::ATTR_EMULATE_PREPARES` in your database connection.

In-depth Changes

Denote Site Language Awareness

- A `SiteLanguageAwareInterface` has been introduced.
- The interface can be used to denote a class as aware of the site language.
- Routing aspects, that take the site language into account, are now using the `SiteLanguageAwareInterface` in addition to the `SiteLanguageAwareTrait`.

In-depth Changes

System Log API

- The following options have been removed from TYPO3's default configuration:
 - `$GLOBALS['TYPO3_CONF_VARS']['SYS']['systemLog']`
 - `$GLOBALS['TYPO3_CONF_VARS']['SYS']['systemLogLevel']`
- Extension authors are advised to use the Logging API and remove the systemLog options.

In-depth Changes

Native List Pagination

- Native support for the pagination of lists such as arrays or QueryResults of Extbase has been introduced.
- The PaginatorInterface defines a basic set of methods.
- The AbstractPaginator class holds the main pagination logic.
- This enables developers to implement all kinds of paginators.

```
use TYPO3\CMS\Core\Pagination\ArrayPaginator;

$itemList = ['apple', 'banana', 'strawberry', 'raspberry', 'anas'];
$currentPageNumber = 3;
$itemPerPage = 2;

$paginator = new ArrayPaginator($itemList, $currentPageNumber, $itemPerPage);
$paginator->getNumberOfPages(); // returns 3
$paginator->getCurrentPageNumber(); // returns 3
$paginator->getKeyOfFirstPaginatedItem(); // returns 5
$paginator->getKeyOfLastPaginatedItem(); // returns 5
```

In-depth Changes

Service API

- Argument `$excludeServiceKeys` is used for skipping certain services when using a chain.
- The argument has been changed from a comma-separated list to an array.
- This change affects the Service API within the following components:
 - `GeneralUtility::makeInstanceService()`
 - `ExtensionManagementUtility::findService()`
- Passing a comma-separated list still works but has been marked as **deprecated**.

In-depth Changes

Modify hreflang-tag

- It is now possible to modify hreflang tags before they get rendered.
- Developers can achieve this by registering an event listener for the following event:

`TYPO3\CMS\Frontend\Event\ModifyHrefLangTagsEvent`

In-depth Changes

Modify the CKEditor Configuration

- The following PSR-14-based events have been introduced which allow to modify the CKEditor configuration:

```
TYPO3\CMS\RteCKEditor\Form\Element\Event\AfterGetExternalPluginsEvent  
TYPO3\CMS\RteCKEditor\Form\Element\Event\BeforeGetExternalPluginsEvent  
TYPO3\CMS\RteCKEditor\Form\Element\Event\AfterPrepareConfigurationForEditorEvent  
TYPO3\CMS\RteCKEditor\Form\Element\Event\BeforePrepareConfigurationForEditorEvent
```

- The [change log](#) for an example.

In-depth Changes

API for AJAX Requests

- The **Fetch API** has been introduced to perform AJAX requests and to make TYPO3 less dependent on jQuery.
- The API provides a generic definition of Request and Response objects (and other things involved with network requests).
- Supported by all modern browsers, see [compatibility chart](#).
- The TYPO3 core uses the new API in the Install Tool, FormEngine, and context menus already.
- See the [change log](#) for some examples on how to use the Fetch API.

In-depth Changes

TCA Description Fields

- The description field in the TCA can now contain line breaks to make long texts more readable.

In-depth Changes

Classes `BasicFileUtility` and `ExtendedFileUtility`

- The following two legacy classes have been marked as **internal** and should not be used anymore:
 - `TYPO3\CMS\Core\Utility\File\BasicFileUtility`
 - `TYPO3\CMS\Core\Utility\File\ExtendedFileUtility`
- Extension developers should use the classes `ResourceStorage` and `ResourceFactory` for managing assets instead.

In-depth Changes

Console Commands: Symfony DI Support

- Command dependencies can now be injected via constructor or other injection techniques.
- Add the `console.command` tag to command classes.
- Use the tag attribute `command` to specify the command name.
- The optional tag attribute `schedulable` can be set to `false` to exclude the command from the TYPO3 scheduler.
- See [change log](#) for an example.

In-depth Changes

Action Buttons in Modals

- Modal popups now support action buttons.
- As an alternative to the existing `trigger` option, the new option `action` can be used.
- For example:

```
Modal.confirm('Header', 'Some content', Severity.error, [
  {
    text: 'Based on trigger()',
    trigger: function () {
      console.log('Vintage!');
    }
  },
  {
    text: 'Based on action()',
    action: new DeferredAction(() => {
      return new AjaxRequest('/any/endpoint').post({});
    })
  }
]);
```

In-depth Changes

JavaScript Event API

- A new Event API enables JavaScript developers to have a stable event listening interface.
- The API takes care of common pitfalls like event delegation and clean event unbinding.
- Each *event strategy* offers two ways to bind a listener to an event.
- The Event API offers several strategies to handle event listeners.
- See [change log](#) for examples and further details.

In-depth Changes

Miscellaneous

- Type of Twitter Card can be selected/configured now. This option renders the meta tag `twitter:card` in the frontend.

```
page {  
  meta {  
    twitter:card = summary_large_image  
    twitter:card.replace = 1  
  }  
}
```

- Only parameters that are needed to calculate the cHash are included in canonicalized URLs by default. Additional query parameters can now be configured:

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['additionalCanonicalizedUrlParameters'].
```

Note: only add parameters which change the content of your page. Otherwise search engines will likely classify your pages as duplicate content.

In-depth Changes

Miscellaneous

- When importing XML data using `EXT:impexp`, the File Deny Pattern applies now and rejects embedded PHP files for example.
- RTE image handling functionality has been removed completely. For image support in CKEditor, consider to use `EXT:rte_ckeditor_image` for example.
- A property within workspaces for *unpublishing* records has been removed in v10 (including the database field `sys_workspace.unpublish_time`). This feature was disabled in TYPO3 v4.5 and not used or provided by the TYPO3 core.

In-depth Changes

Miscellaneous

- When rendering HTML5 output, `<script>` tags do not include the attribute `type="text/javascript"` anymore.
- This can be re-enabled for the frontend by using TypoScript if required:

```
page {
  includeJS {
    myfile = EXT:example/Resources/Public/JavaScript/myfile.js
    myfile.type = text/javascript
  }
}
```

- The following deprecated system extensions have been removed:
 - EXT:rsaauth
 - EXT:fe_edit

In-depth Changes

Miscellaneous

- Directive `createDirs` in file `ext_emconf.php` not supported anymore.
 - Folders will not be created automatically during extension installation.
- The following two properties in class `TYPO3\CMS\Extbase\Mvc\Controller\ActionController` have been removed:
 - `$namespacesViewObjectNamePattern`
 - `$viewFormatToObjectNameMap`
- The following existing hook has been extended and can now also be used to validate passwords:
`$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['fellogin']['password_changed']`

In-depth Changes

Miscellaneous

- The following methods of class `\TYPO3\CMS\Core\Utility\VersionNumberUtility` have been marked deprecated:
 - `convertIntegerToVersionNumber()`
 - `splitVersionRange()`
 - `raiseVersionNumber()`
- Implement the methods as custom code.

In-depth Changes

Miscellaneous

- It is now possible to get the default value of a class property when using the ReflectionService.

```
$property = GeneralUtility::makeInstance(ReflectionService::class)
->getClassSchema(MyClass::class)
->getProperty('myProperty');
```

- Backend routes to modules without path configurations are now named `"/module/<main-module>/<sub-module>"` by default (for example: `"/module/web/ts"`.)
- Old routes still work (e.g. `"/web/ts/"`) but this syntax will be removed in TYPO3 v11.

In-depth Changes

Miscellaneous

- The FormEngine DataProvider `parentPageTca` has been removed.
 - Developers can access `$GLOBALS['TCA']['pages']` directly, instead of `$result['parentPageTca']`.
- The following database fields have been removed:
 - `tt_content.spaceBefore` (replaced by field `space_before_class`)
 - `tt_content.spaceAfter` (replaced by field `space_after_class`)
 - `pages.t3ver_id` (unused since TYPO3 v9)
- The PHP class `\TYPO3\CMS\Frontend\Page\PageRepository` has been moved from the "frontend" system extension into the core.
 - Replace with class: `\TYPO3\CMS\Core\Domain\Repository\PageRepository`

In-depth Changes

Miscellaneous

- 4th parameter of method `PageRepository->enableFields()` has been removed.
 - If developers use a 4th parameter in this method call, which is set to **"false"**, this can be removed safely.
 - If it is set to **"true"**, the code needs to be replaced with a separate instance of `PageRepository` with a custom `Context`.
- The internal method `File::_getMetaData()`, which is used to fetch meta data of a file, has been marked deprecated.
 - Use `$fileObject->getMetaData()->get()` to fetch the meta data instead.
- The route identifier **"xMOD_tximpexp"** has been marked deprecated.
 - Use `tx_impexp_export` or `tx_impexp_import` depending on the use case.

In-depth Changes

Miscellaneous

- The following abstract method has been removed:
`\TYPO3\CMS\Extbase\Configuration\AbstractConfigurationManager::getSwitchableControllerActions()`
 - Use the new method name `getControllerConfiguration()` instead (same PHP class).
- The global variable `$TBE_TEMPLATE` has been removed, including the related PSR-15 middleware (which was marked as internal).
 - Instantiate the `DocumentTemplate` class directly in the controller of the module.
 - Migrate to `ModuleTemplate` which is available since TYPO3 v7.
- The global variable `$GLOBALS['T3_VAR']` has been removed.

In-depth Changes

Miscellaneous

- The `hashParameters` for calculating the `hashBase` in the following class have been modified:

`TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController`

- `gr_list` has been replaced by `groupIds`.
 - `cHash` has been replaced by `dynamicArguments`.
 - `domainStartPage` has been replaced by `site` (site identifier).
- Two new events are dispatched when records are rolled back:
 - `TYPO3\CMS\Backend\History\Event\BeforeHistoryRollbackStartEvent`
 - `TYPO3\CMS\Backend\History\Event\AfterHistoryRollbackFinishedEvent`

In-depth Changes

Miscellaneous

- Method `set()` of the Query Builder now accepts a 4th argument to specify the type of the named parameter:

```
TYPO3\CMS\Core\Database\Query\QueryBuilder::set()  
(the default is \PDO::PARAM_STR)
```

In-depth Changes

Miscellaneous

- The IP locking functionality has been extended to also support IPv6 (frontend and backend).

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['lockIPv6'] = 2;  
$GLOBALS['TYPO3_CONF_VARS']['BE']['lockIPv6'] = 2;
```

- The public property `lockIP` in following PHP class has been removed: `\TYPO3\CMS\Core\Authentication\AbstractUserAuthentication`.
- Migration options:
 - Set `lockIP` and `lockIPv6` in `$GLOBALS['TYPO3_CONF_VARS']` [...].
 - Use the new IP-Locker API: `\TYPO3\CMS\Core\Authentication\IpLocker`.

Deprecated/Removed Functions

Deprecated/Removed Functions

The TYPO3 core cleaned up and ready for the future

Deprecated/Removed Functions

JavaScript Deprecations

- Two global JavaScript variables have been marked as **deprecated**:
 - `T3_THIS_LOCATION`
 - `T3_RETURN_URL`
- The well-known JavaScript function `jumpToUrl()` has been marked as **deprecated**. Migration options:
 - use `window.location.href = '...'`;
 - or use a link in HTML like `link`
- The JavaScript function `jumpExt()` has been marked as **deprecated**.

Deprecated/Removed Functions

JavaScript Deprecations

- The jQuery plugin `jquery.clearable`, that provides a button to clear an input field, has been marked as **deprecated**.
- Migration: use module `TYPO3\CMS\Backend\Input\Clearable` and the method `clearable()` on a native `HTMLInputElement`.

```
require(['TYPO3\CMS\Backend\Input\Clearable'], function() {
    const inputField = document.querySelector('#myinput');
    if (inputField !== null) {
        inputField.clearable();
    }

    const clear = Array.from(document.querySelectorAll('.t3js-clearable')).filter(inputElement => {
        return !inputElement.classList.contains('t3js-datetimepicker');
    });
    clear.forEach(clearableField => clearableField.clearable());
});
```

Deprecated/Removed Functions

CLI Command Handler

- CLI commands are handled using the `CommandApplication` class.
- This class is a wrapper around the [Symfony Console](#).
- The previously used interface and the class `CommandRequestHandler` have been marked as **deprecated**:
 - `TYPO3\CMS\Core\Console\RequestHandlerInterface`
 - `TYPO3\CMS\Core\Console\CommandRequestHandler`

Deprecated/Removed Functions

RecordHistory Handling

Changes made to the class `TYPO3\CMS\Backend\History\RecordHistory`:

- Visibility of properties `changeLog` and `lastHistoryEntry` changed to protected (and public getter function added).
- Visibility of methods `getHistoryEntry()` and `getHistoryData()` changed to protected.
- The following methods have been marked as **deprecated**:
 - `createChangeLog()`
 - `shouldPerformRollback()`
 - `getElementData()`
 - `performRollback()`
 - `createMultipleDiff()`
 - `setLastHistoryEntry()`

Deprecated/Removed Functions

XML Language Files

- The XLIFF format is used for language files since TYPO3 v4.6.
- The usage of XML language files is now marked as **deprecated** and triggers a warning/error.
- This includes executing the following XML-parser:
`TYP03\CMS\Core\Localization\Parser\LocallangXmlParser`

Deprecated/Removed Functions

FormEngine

- The following functions of the FormEngine have been marked as **deprecated**:
 - `setFormValueOpenBrowser()`
(use `FormEngine.openPopupWindow()` instead)
 - `setFormValueFromBrowseWin()`
(use `FormEngine.setSelectOptionFromExternalSource()` instead)
 - `setHiddenFromList()`
(use `FormEngine.updateHiddenFieldValueFromSelect()` instead)
 - `setFormValueManipulate()`
(no replacement, as this is internal logic)
 - `setFormValue_getFObj()`
(use `FormEngine.getFormElement()` instead)

Deprecated/Removed Functions

Signal/Slot

- The following Signal/Slots were replaced by PSR-14 events and therefore have been marked as **deprecated**:

- `TYPO3\CMS\Core\Imaging\IconFactory::buildIconForResourceSignal`
- `TYPO3\CMS\Core\Database\SoftReferenceIndex::setTypoLinkPartsElement`
- `TYPO3\CMS\Core\Database\ReferenceIndex::shouldExcludeTableFromReferenceIndex`
- `TYPO3\CMS\Core\Utility\ExtensionManagementUtility::tcaIsBeingBuilt`
- `TYPO3\CMS\Install\Service\SqlExpectedSchemaService::tablesDefinitionIsBeingBuilt`
- `TYPO3\CMS\Core\Tree\TableConfiguration\DatabaseTreeDataProvider::PostProcessTreeData`
- `TYPO3\CMS\Backend\Backend\ToolBarItems\SystemInformationToolBarItem::getSystemInformation`
- `TYPO3\CMS\Backend\Backend\ToolBarItems\SystemInformationToolBarItem::loadMessages`

Deprecated/Removed Functions

Application Context

- The method `GeneralUtility::getApplicationContext()` has been marked **deprecated**.
- The following method should be used instead:
`TYPO3\CMS\Core\Core\Environment::getContext()`.

Deprecated/Removed Functions

Switchable Controller Actions

- "Switchable Controller Actions" (SCA) have been marked as **deprecated**.
- SCA are used to override the allowed set of controllers and actions using TypoScript or Flexforms.
- Using the same plugin as an entry point for many different functionalities contradicts the idea of a plugin serving one specific purpose.
- Plugins that use SCA should be split into multiple different plugins.

Deprecated/Removed Functions

Global Constants

- The following two global constants have been marked as **deprecated**:
 - `TYP03_version`
 - `TYP03_branch`
- The following new PHP class should be used instead:
`TYP03\CMS\Core\Information\Typo3Version`

Deprecated/Removed Functions

Extbase: WebRequest/WebResponse

- The following two Extbase classes have been marked as **deprecated**:
 - TYPO3\CMS\Extbase\Mvc\Web\Request
 - TYPO3\CMS\Extbase\Mvc\Web\Response

Deprecated/Removed Functions

Simplified RTE Parser API

- The PHP class `RteHtmlParser` features a simplified API now.
- As a consequence, the following two methods have been marked as **deprecated**:
 - `TYPO3\CMS\Core\Html\RteHtmlParser->init()`
 - `TYPO3\CMS\Core\Html\RteHtmlParser->RTE_transform()`

Deprecated/Removed Functions

Console Commands Configuration

- As the console commands configuration has been migrated to Symfony service tags, the console command configuration file `Configuration/Commands.php` has been marked as **deprecated**.
- Use the dependency injection service tag `console.command` instead.

Deprecated/Removed Functions

TCA: `transOrigPointerField`

- Excluding the field pointed by the following TCA option led to inconsistent data stored in the database under certain circumstances:
`$GLOBALS['TCA'][$table]['ctrl']['transOrigPointerField']`
- Therefore, the targeted field can not be excluded anymore.
- A migration wizard removes the option from the TCA and adds a deprecation message to the deprecation log in case code needs to be updated.

Deprecated/Removed Functions

DocumentTemplate

- The following class has been marked as **deprecated**:
 - TYPO3\CMS\Backend\Template\DocumentTemplate
- It was used as a basis to render backend modules or HTML-based output in TYPO3 backend.
- Since TYPO3 v7, the new API via ModuleTemplate should be used instead.

```
use TYPO3\CMS\Backend\Template\ModuleTemplate;
...
$moduleTemplate = GeneralUtility::makeInstance(ModuleTemplate::class);
$content = $this->getHtmlContentFromMyModule();
$moduleTemplate->setTitle('My module');
$moduleTemplate->setContent($content);
return new HtmlResponse($moduleTemplate->renderContent());
```

Deprecated/Removed Functions

LinkValidator

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

```
TYPO3\CMS\Linkvalidator\Repository\BrokenLinkRepository  
->getNumberOfBrokenLinks()
```

- Use the following method in the same class instead:

```
BrokenLinkRepository::isLinkTargetBrokenLink()
```

Deprecated/Removed Functions

GeneralUtility

- The following GeneralUtility methods have been marked **deprecated**:
 - GeneralUtility::IPv6Hex2Bin()
 - GeneralUtility::IPv6Bin2Hex()
 - GeneralUtility::compressIPv6()
 - GeneralUtility::milliseconds()
 - GeneralUtility::linkThisUrl()
 - GeneralUtility::flushDirectory()

- Setting additional arguments besides the URL in GeneralUtility::getUrl() has been marked **deprecated**.
(this includes: \$includeHeader, \$requestHeaders, and \$report)

Deprecated/Removed Functions

GeneralUtility

- The following method has been removed from the GeneralUtility class: `GeneralUtility::isRunningOnCgiServerApi()`.
- This function is now available as `Environment::isRunningOnCgiServer()`.

Deprecated/Removed Functions

LanguageService Functionality

Changes of the LanguageService (also known as `$GLOBALS[LANG]`).

- The visibility of the following functions has changed:
 - `LanguageService->LL_files_cache` (now protected)
 - `LanguageService->LL_labels_cache` (now protected)
 - `LanguageService->getLLL()` (now protected)
 - `LanguageService->debugLL()` (now protected)
- The following function has been marked **deprecated**:
 - `LanguageService->getLabelsWithPrefix()`
- The following function has been marked **internal** now:
 - `LanguageService->loadSingleTableDescription()`

Deprecated/Removed Functions

Internal Libraries and Classes

- The following internal libraries have been removed:
 - "bootstrap-slider"
 - "jQuery.datatables"
- The internal class `TYPO3\CMS\Extbase\Domain\Model\FileMount` has been marked **deprecated**.
- *Reminder: Extension authors should never use libraries that are not marked as public API.*

Deprecated/Removed Functions

ContentObjectRenderer

- The following hooks within have been marked **deprecated**:

```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['tslib/class.tslib_content.php']...
```

- ['cObjTypeAndClass']
 - ['cObjTypeAndClassDefault']
 - ['extLinkATagParamsHandler']
 - ['typolinkLinkHandler']
- The following methods have been marked **deprecated**:
 - cImage()
 - getBorderAttr()
 - getImageTagTemplate()
 - getImageSourceCollection()
 - linkWrap()
 - getAltParam()

(all methods have been moved to the ImageContentObject class)

Deprecated/Removed Functions

Fluid AutoComplete ViewHelper

- The Fluid ViewHelper `<f:widget.autocomplete>` and the related controller have been marked **deprecated**.
- If you use or extend the following, you should take action:
 - `TYPO3\CMS\Fluid\ViewHelpers\Widget\AutocompleteViewHelper`
 - `TYPO3\CMS\Fluid\ViewHelpers\Widget\Controller\AutocompleteController`

Deprecated/Removed Functions

FileCollection Models

- The following FileCollection models have been marked **deprecated**:
 - `TYPO3\CMS\Extbase\Domain\Model\StaticFileCollection`
 - `TYPO3\CMS\Extbase\Domain\Model\FolderBasedFileCollection`
 - `TYPO3\CMS\Extbase\Domain\Model\AbstractFileCollection`
 - `TYPO3\CMS\Extbase\Property\TypeConverter\StaticFileCollectionConverter`
 - `TYPO3\CMS\Extbase\Property\TypeConverter\FolderBasedFileCollectionConverter`
 - `TYPO3\CMS\Extbase\Property\TypeConverter\AbstractFileCollectionConverter`

Deprecated/Removed Functions

Extbase SignalSlot Dispatcher

- During the TYPO3 v10 development over the last months, all Extbase signals provided by TYPO3 core have been migrated to PSR-14 Events.
- The Extbase "SignalSlot Dispatcher" has now been marked **deprecated**.
- Extension developers are advised to migrate to PSR-14 Events and Event Listeners.

Deprecated/Removed Functions

FileNameValidator API

- The logic for validating if the file name of a new (uploaded) or renamed is valid, is now available as a new FileNameValidator API:

TYPO3\CMS\Core\Resource\Security\FileNameValidator

- Migration options:

```
// OLD:  
GeneralUtility::verifyFilenameAgainstDenyPattern($filename)  
// NEW:  
GeneralUtility::makeInstance(FileNameValidator::class)->isValid($filename)
```

```
// OLD:  
FILE_DENY_PATTERN_DEFAULT  
// NEW:  
FileNameValidator::DEFAULT_FILE_DENY_PATTERN
```


Deprecated/Removed Functions

EXT:feLogin Hooks

- All legacy hooks of EXT:feLogin have been disabled and will be removed in TYPO3 v11:
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['beforeRedirect']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['postProcContent']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['password_changed']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['forgotPasswordMail']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['login_confirmed']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['login_error']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['loginFormOnSubmitFuncs']`
 - `$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['feLogin']['logout_confirmed']`
- See [change log](#) for migration options.

Deprecated/Removed Functions

Miscellaneous

- Mail functionality should not be included in the class `TYPO3\CMS\Frontend\ContentObject\ContentObjectRenderer`. Therefore, the method `sendNotifyEmail()` has been marked **deprecated** and will be removed in TYPO3 v11.
- The method `editOnClick()` used to generate JavaScript `onclick` targets has been marked as **deprecated** in the following class: `TYPO3\CMS\Backend\Utility\BackendUtility`.

Deprecated/Removed Functions

Miscellaneous

- The property `$environmentService` of the following class has been marked as **deprecated**: `TYPO3\CMS\Extbase\Mvc\Web\Response`.
(inject the environment service manually in the subclass if required)
- The following internal class has been marked as **deprecated**:
`TYPO3\CMS\Extbase\Mvc\Controller\AbstractController`.
- The following method has been marked as **deprecated**:
`TYPO3\CMS\Backend\Utility`
`BackendUtility::TYPO3_copyRightNotice`.
(use the new PHP class `Typo3Copyright` and its method `getCopyrightNotice()` instead)

Deprecated/Removed Functions

Miscellaneous

- The PHP property `TypoScriptFrontendController->divSection` has been marked as **deprecated**.
(use `$GLOBALS['TSFE']->additionalHeaderData[]` to add JavaScript within uncached content instead)
- As part of the Symfony 5.0 support in TYPO3 v10.2, incompatible environment related dependency injection services have been removed:
 - `env.is_unix`
 - `env.is_windows`
 - `env.is_cli`
 - `env.is_compoer_mode`

Installation and Upgrade

Installation and Upgrade

There is no better time to check out TYPO3 v10 LTS

Installation and Upgrade

Classic Installation Method

- Official *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/10  
$ tar xzf typo3_src-10.4.0.tar.gz  
$ cd htdocs  
$ ln -s ../typo3_src-10.4.0 typo3_src  
$ ln -s typo3_src/index.php  
$ ln -s typo3_src/typo3  
$ touch FIRST_INSTALL
```

- Symbolic links under Microsoft Windows:
 - Use junction under Windows XP/2000
 - Use mklink under Windows Vista and Windows 7 and higher

Installation and Upgrade

Installation Using PHP Composer

- Installation using *composer* under Linux, Mac OS X and Windows 10:

```
$ cd /var/www/site/
```

```
$ composer create-project typo3/cms-base-distribution typo3v10 ^10
```

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

```
$ composer install
```

Further details about Composer for the TYPO3 core and for TYPO3 extensions are available at:

<https://get.typo3.org/misc/composer>

Installation and Upgrade

Upgrade to TYPO3 v10 LTS

- Upgrades are only possible from TYPO3 v9 LTS.
- Sites using TYPO3 < v9 LTS should be updated to the latest v9.5.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
 - Deploy new sources and run Install Tool -> Upgrade Wizard
- Read more in the official [TYPO3 Installation and Upgrade Guide](#).

Sources and Authors

Sources and Authors

Sources

TYPO3 News:

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

Release Infos:

- https://get.typo3.org/release-notes/10.x/TYPO3\CMS_10.4.0
- [TYPO3 v10 change log](#)
- `typo3/sysexst/core/Documentation/Changelog/10.4/*`

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 Team:

Paul Blondiaux, Pierrick Caillon, Andreas Fießer,
Richard Haeser, Jigal van Hemert, Henrietta Kucsovan, Corina Miron,
Sinisa Mitrovic, Jaume Presas i Puig,
Michael Schams y Roberto Torresani

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

Licensed under Creative Commons BY-NC-SA 3.0

