TYPO3 Version 10.1 – What’s New

Summary of the new features, changes and improvements

Created by:
Michael Schams

08/October/2019

TYPO3 Version 10.1 - What's New

Chapter Overview

Introduction

Backend User Interface

Changes for Integrators

Changes for Developers

Deprecated/Removed Functions

Miscellaneous

Sources and Authors

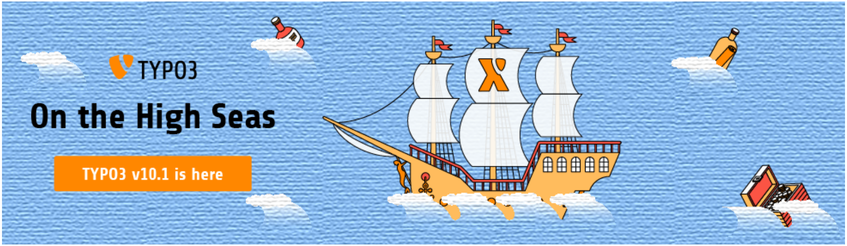
Introduction

The Facts

Introduction

TYPO3 Version 10.1 - The Facts

- Release date: 01 October 2019
- Release type: Sprint Release



Introduction

Executive Summary

TYPO3 version 10.1 is the second sprint release on the way to the LTS-version (long-term support) in 2020.

The new release received more than 240 Git commits (reviewed, tested and approved source code changes) since its predecessor version 10.0 published ten weeks ago.

Although backend users won't see many obvious changes or major new functions as such, TYPO3 version 10.1 received a number of improvements under the hood.

Introduction

System Requirements

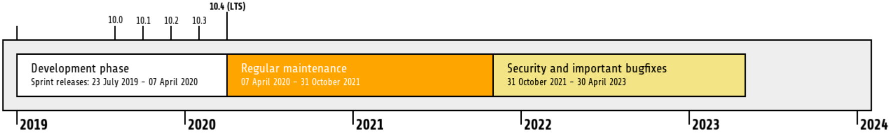
- PHP version 7.2 or 7.3
- PHP settings:
  - `memory_limit` >= 256M
  - `max_execution_time` >= 240s
  - `max_input_vars` >= 1500
  - compilation option `--disable-ipv6` must not be used
- Most database servers supported by **Doctrine DBAL** also work with TYPO3. Tested DB engines are for example:

MySQL® MariaDB PostgreSQL Microsoft® SQL Server™ SQLite

Introduction

Development, Release and Maintenance Timeline

TYPO3 v10



Extended Support

The TYPO3 GmbH offers further support options for TYPO3 v10 LTS even after 30 April 2023 for up to two additional years.

Introduction

TYPO3 v10 Roadmap

Release dates and their primary focus:

v10.0	23/July/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	04/Feb/2020	Feature Freeze
v10.4	07/Apr/2020	LTS Release (Long-term Support)

https://typo3.org/article/typo3-v10-roadmap/

https://typo3.org/article/typo3-v10-safe-and-sound/

Introduction

Installation

- 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/10.1
$ tar xzf typo3_src-10.1.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-10.1.0 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

Introduction

Installation Using composer

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

```
$ cd /var/www/site/
$ composer create-project typo3/cms-base-distribution typo3v10 ^10.1
```

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

```
$ composer install
```

Further details and examples for `composer.json` files are available at:
https://composer.typo3.org

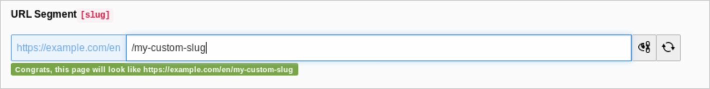
Chapter 1:

Backend User Interface

Backend User Interface

Slug Updates and Redirects (1)

- 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.



URL Segment [slug]

https://example.com/en /my-custom-slug

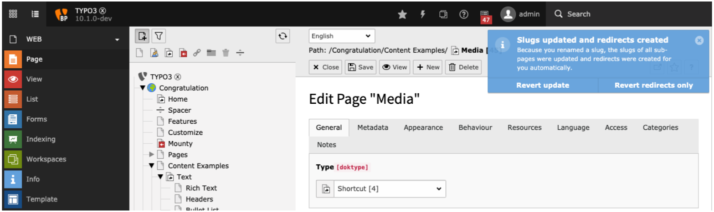
Congrats, this page will look like https://example.com/en/my-custom-slug

- Since TYPO3 v10.1, two actions prevent this from happening:
  - 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 (2)

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



Slugs updated and redirects created

Because you renamed a slug, the slugs of all sub-pages were updated and redirects were created for you automatically.

Path: /Congratulation/Content Examples/ Media [45]

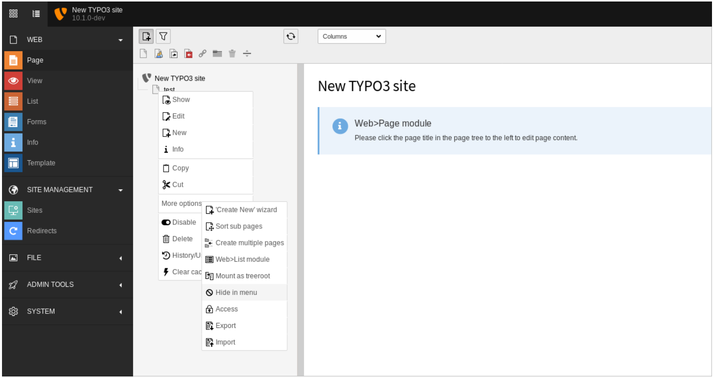
Edit Page "Media"

Type [doktype]: Shortcut [4]

Backend User Interface

Hide/Show in Menu

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



New TYPO3 site

Web>Page module

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

Chapter 2:

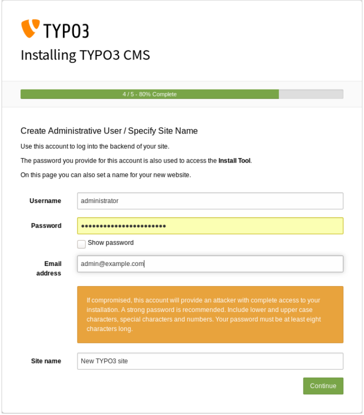
Changes for Integrators

Changes for Integrators

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**.



TYPO3

Installing TYPO3 CMS

Create Administrative User / Specify Site Name

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.

Username	administrator
Password	••••••••••••••••••••••
	□ Show password
Email address	admin@example.com
	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.
Site name	New TYPO3 site

Changes for Integrators

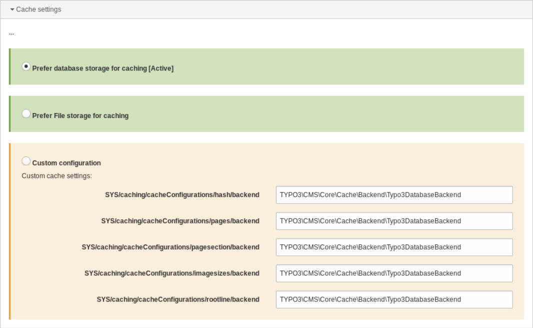
Cache Storage Type (1)

- 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.

Changes for Integrators

Cache Storage Type (2)

- Backend: **ADMIN TOOLS ➟ Settings ➟ Configuration Presets**:



▾ Cache settings

...

● **Prefer database storage for caching [Active]**

○ **Prefer File storage for caching**

○ **Custom configuration**

Custom cache settings:

SYS/caching/cacheConfigurations/hash/backend	TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend
SYS/caching/cacheConfigurations/pages/backend	TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend
SYS/caching/cacheConfigurations/pagesection/backend	TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend
SYS/caching/cacheConfigurations/imagesizes/backend	TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend
SYS/caching/cacheConfigurations/rootline/backend	TYPO3\CMS\Core\Cache\Backend\Typo3DatabaseBackend

Changes for Integrators

Site Configuration

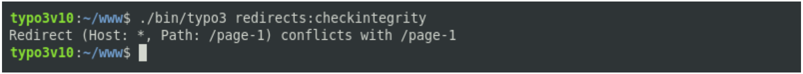
- When a new page is created on the root level, a standard site configuration is automatically generated with it.
- As a result, a basic TYPO3 site can be set up quickly.
- The site configuration features:
  - a pre-defined identifier (e.g. `site-42-a1d0c6e83f`)
  - an entry point (e.g. `https://example.com/site-42`)
  - a default language (e.g. `English`)

Changes for Integrators

Conflicting Redirects (1)

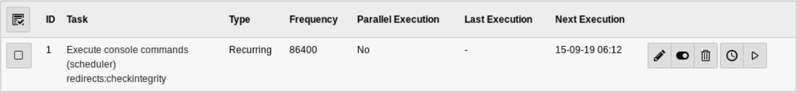
- A new Symfony command has been introduced to detect redirects that conflict with page URLs.
- Execute the command in the CLI:

  (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:

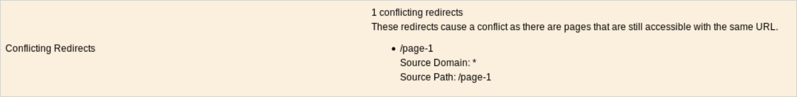


	ID	Task	Type	Frequency	Parallel Execution	Last Execution	Next Execution	
□	1	Execute console commands (scheduler) redirects:checkintegrity	Recurring	86400	No	-	15-09-19 06:12	▷

Changes for Integrators

Conflicting Redirects (2)

- 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.

Changes for Integrators

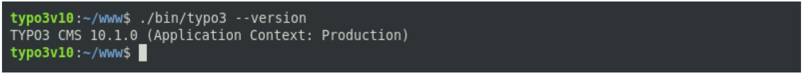
Distribution Packages

- 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.

Changes for Integrators

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$ 
```

Changes for Integrators

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 TypoScript:

    ```
    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}"
    />
    ```

Changes for Integrators

File Uploads

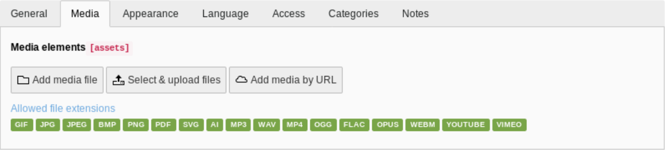
- It is now possible to configure the default action when uploading files in the file list module using drag'n drop.
- User TSConfig:

```
# 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
```

Changes for Integrators

Media Element Buttons

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



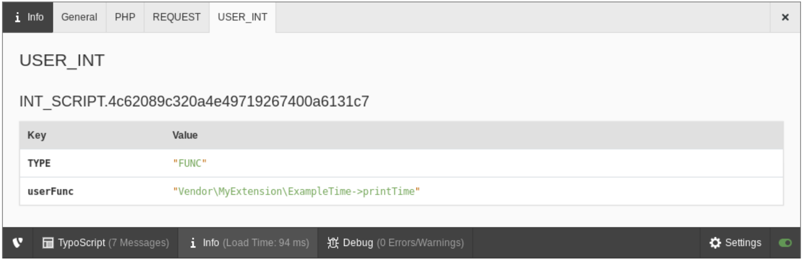
- The example below hides both buttons:

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

Changes for Integrators

Admin Panel

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



USER_INT

INT_SCRIPT.4c62089c320a4e49719267400a6131c7

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

Chapter 3:

Changes for Developers

Changes for Developers

Cache Dependency Injection (1)

- 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’);
  }
}
```

Changes for Developers

Cache Dependency Injection (2)

- Since **TYPO3 v10.1**, 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;
  }
}
```

Changes for Developers

Cache Dependency Injection (3)

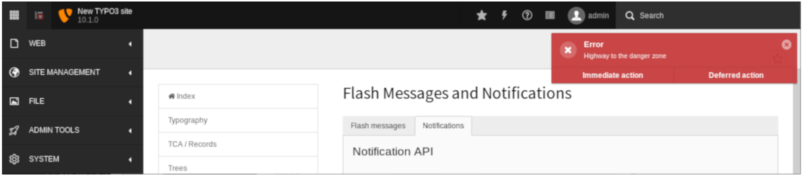
- ...and the following container service configuration is required:

```
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’
```

Changes for Developers

Notification Actions

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



Error
Highway to the danger zone

Flash Messages and Notifications

Notification API

Changes for Developers

Broadcast Channels and Messaging (1)

- 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);
});
```

Changes for Developers

Broadcast Channels and Messaging (2)

- 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.

Changes for Developers

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.

Changes for Developers

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.

Changes for Developers

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();
```

Changes for Developers

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.

Changes for Developers

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’,
);
```

Changes for Developers

Miscellaneous (1)

- 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`

Changes for Developers

Miscellaneous (2)

- 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`)

Chapter 4:

Deprecated/Removed Functions

Deprecated/Removed Functions

JavaScript Deprecations (1)

- 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 `<a href="...">link</a>`
- The JavaScript function `jumpExt()` has been marked as **deprecated**.

Deprecated/Removed Functions

JavaScript Deprecations (2)

- 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

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

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:
  `TYPO3\CMS\Core\Localization\Parser\LocallangXmlParser`

Chapter 5:

Miscellaneous

Miscellaneous

Strengthen Security

- External links generated by TypoLink, or links using `_blank` target, include attribute `rel="noopener noreferrer"` now.
- This aims to strengthen the security of the TYPO3 site:
  - "**noopener**" instructs the browser to open the link without granting the new browsing context access to the document that opened it.
  - "**noreferrer**" prevents the browser, when navigating to another page, to send the page address, or any other value, as referrer via the `Referer:` HTTP header.

Miscellaneous

YAML File Inclusion

- YAML files can already be included by other YAML files using the following syntax:

```
imports:
  - { resource: "EXT:my_extension/Configuration/FooBar/Example.yaml" }

another:
  option: true
```

- This has been extended to import resources relative to the current YAML file:

```
imports:
  - { resource: "subfolder/AnotherExample.yaml" }
  - { resource: "../path/to/configuration/AnotherExample.yaml" }

another:
  option: true
```

Chapter 6:

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_10.1.0
- TYPO3 v10 ChangeLog
- typo3/sysext/core/Documentation/Changelog/10.1/*

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, Sinisa Mitrovic,
Michael Schams and Roberto Torresani

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