TYPO3 Version 9.4 – What's New

Summary of the new features, changes and improvements

Created by:
Michael Schams

12/September/2018

Creative Commons BY-NC-SA 3.0

TYPO3

TYPO3 Version 9.4 – 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 9.4 – The Facts

- Release date: 04 September 2018
- Release type: Sprint Release

Introduction

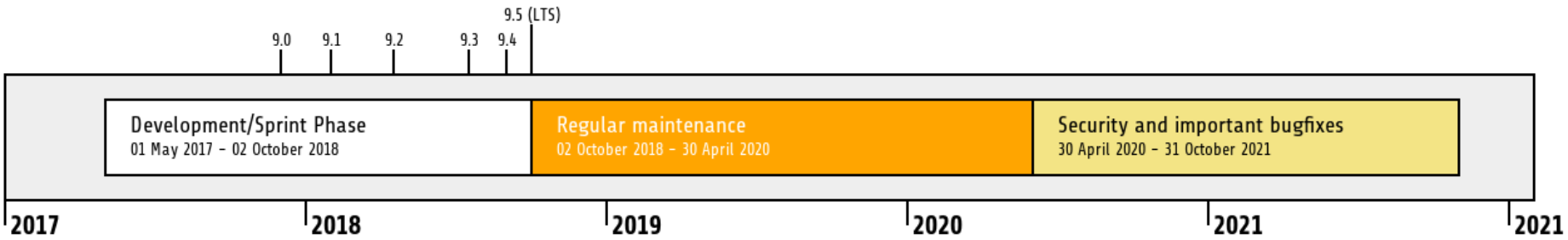
System Requirements

- PHP version 7.2 or higher
- 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:

Introduction

Development, Release and Maintenance Timeline

TYPO3 v9



Extended Support

The TYPO3 GmbH offers further support options for TYPO3 v9 LTS even after 31 October 2021 for up to two additional years.

Introduction

TYPO3 v9 Roadmap

Release dates and their primary focus:

- v9.0 12/Dec/2017 Install Tool and Page Tree Refactoring, Unified Page Translations
- v9.1 30/Jan/2018 Redirect Handling
- v9.2 10/Apr/2018 Site Handling
- v9.3 12/Jun/2018 SEO and URL Routing Preparations
- v9.4 04/Sep/2018 URL Routing for Pages
- v9.5 02/Oct/2018 LTS Release

https://typo3.org/article/typo3-v9-roadmap/

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

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/9.4
$ tar xzf typo3_src-9.4.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-9.4.0 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

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 CmsBaseDistribution
      ^9
```

- 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

Admin Panel (1)

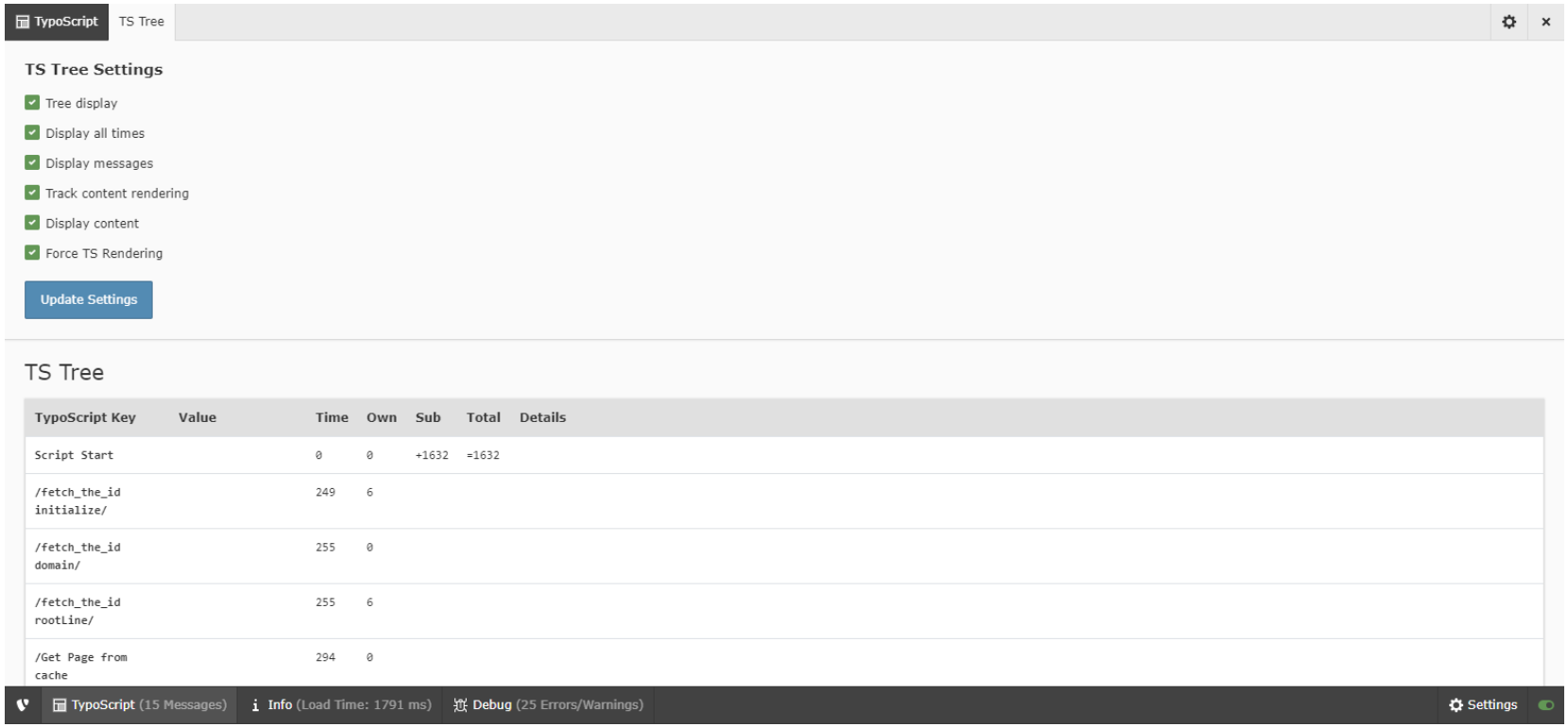
The Admin Panel received a complete overhaul regarding its design as well as the underlying code and architecture.

The Admin Panel is displayed at the bottom of a page in the frontend of TYPO3. The toggle button at the right allows integrators and editors to enable and disable the Admin Panel. The current state shows the *enabled* state.

Backend User Interface

Admin Panel (2)

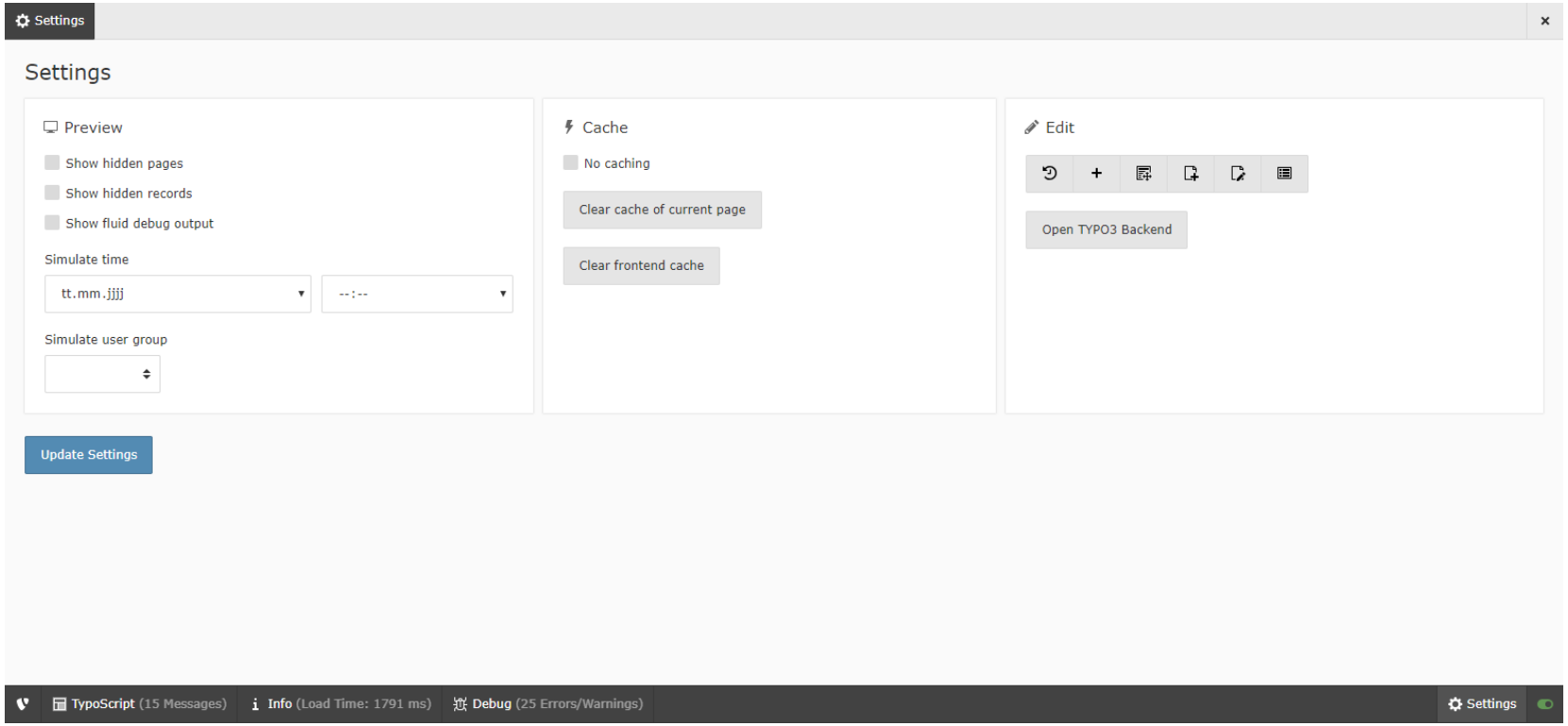
Example screenshot below shows TypoScript options.



Backend User Interface

Admin Panel (3)

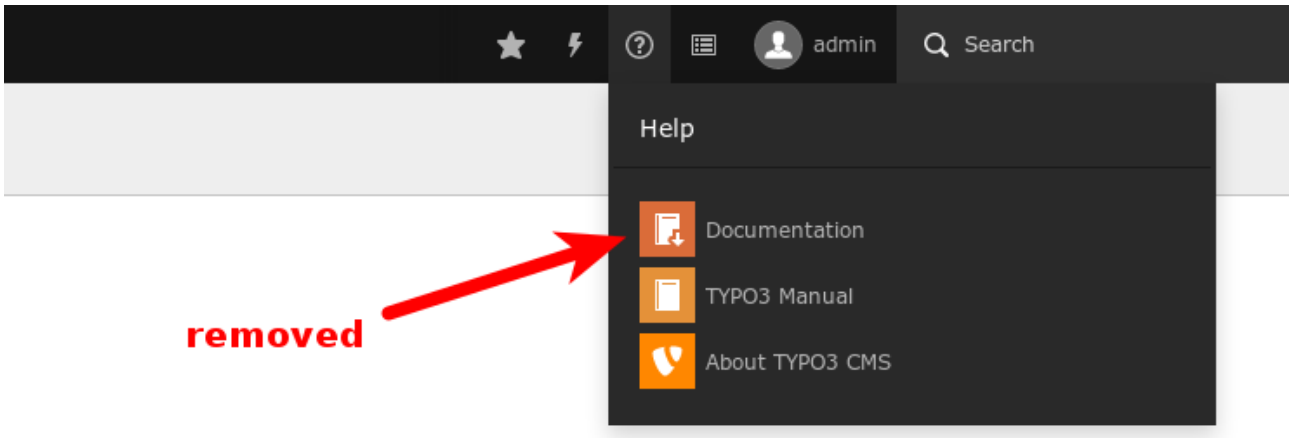
Example screenshot below shows configuration options ("Settings").



Backend User Interface

Extension "Documentation" Removed

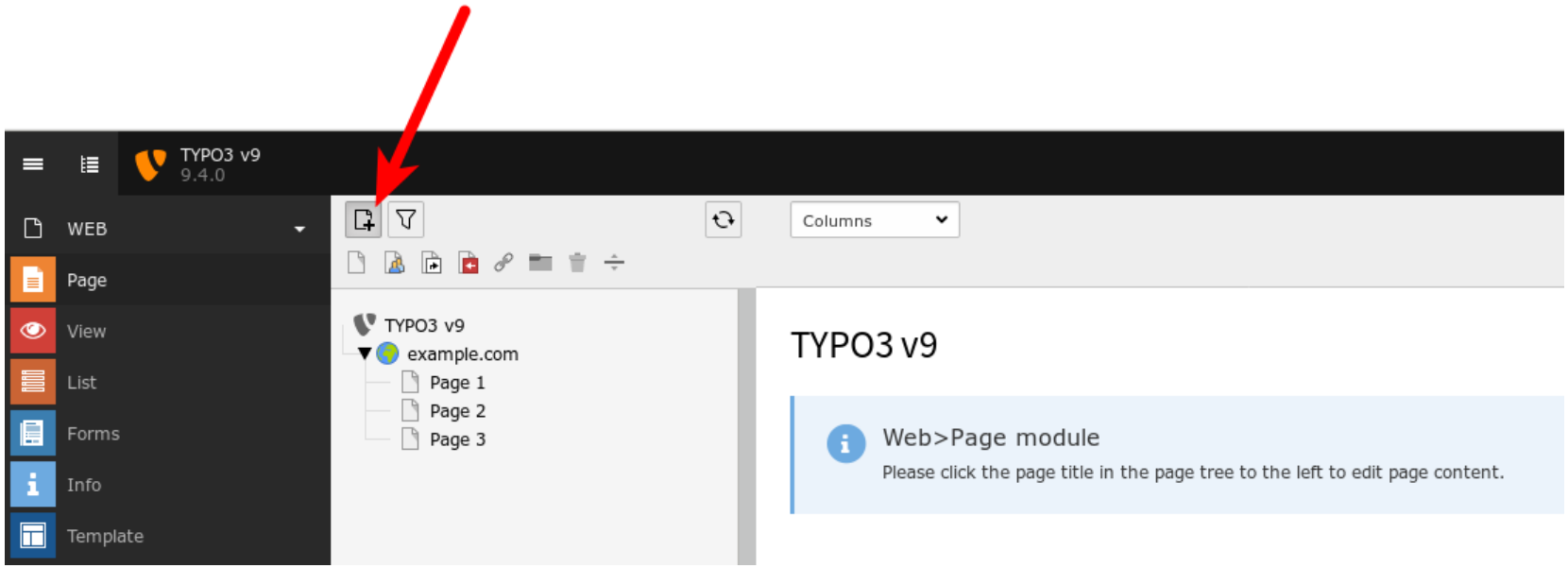
"Documentation" module has been removed from the TYPO3 backend. The module had technical and conceptual flaws and acceptance within the community was not very high.
All documentation remain available at docs.typo3.org.



Backend User Interface

Page Tree toolbar

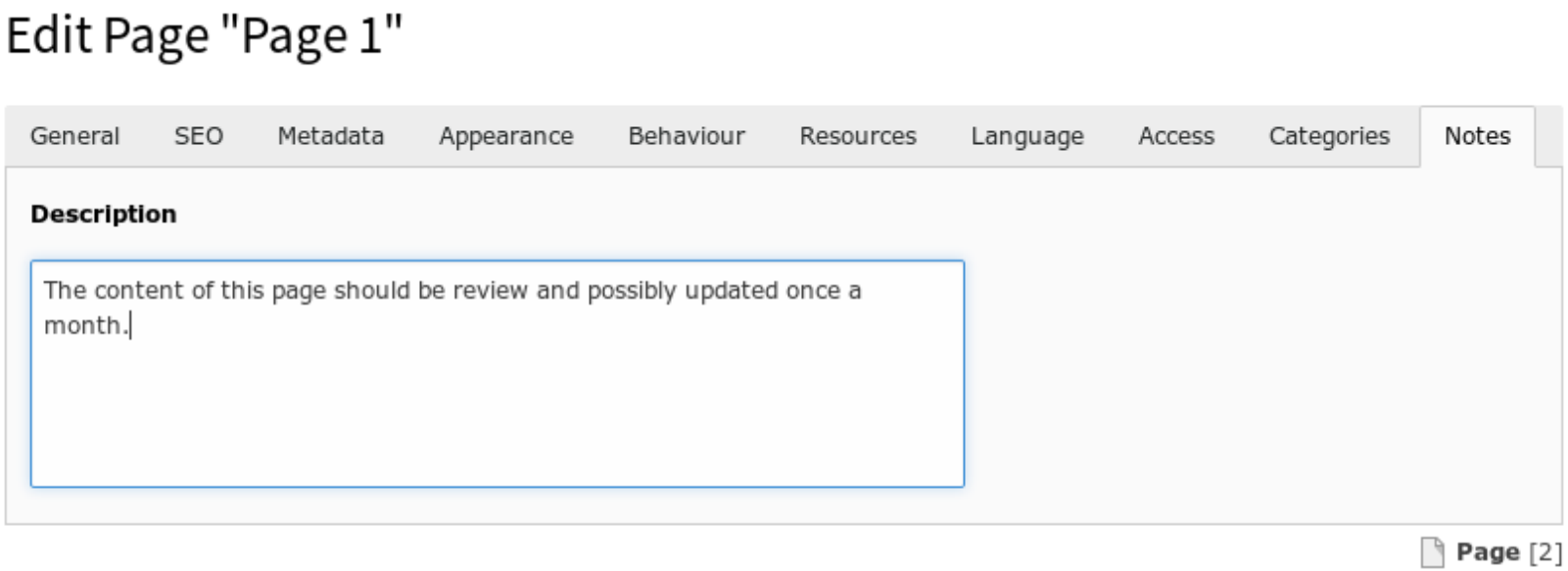
First element of the page tree toolbar is automatically selected on load now.



Backend User Interface

Notes for Pages

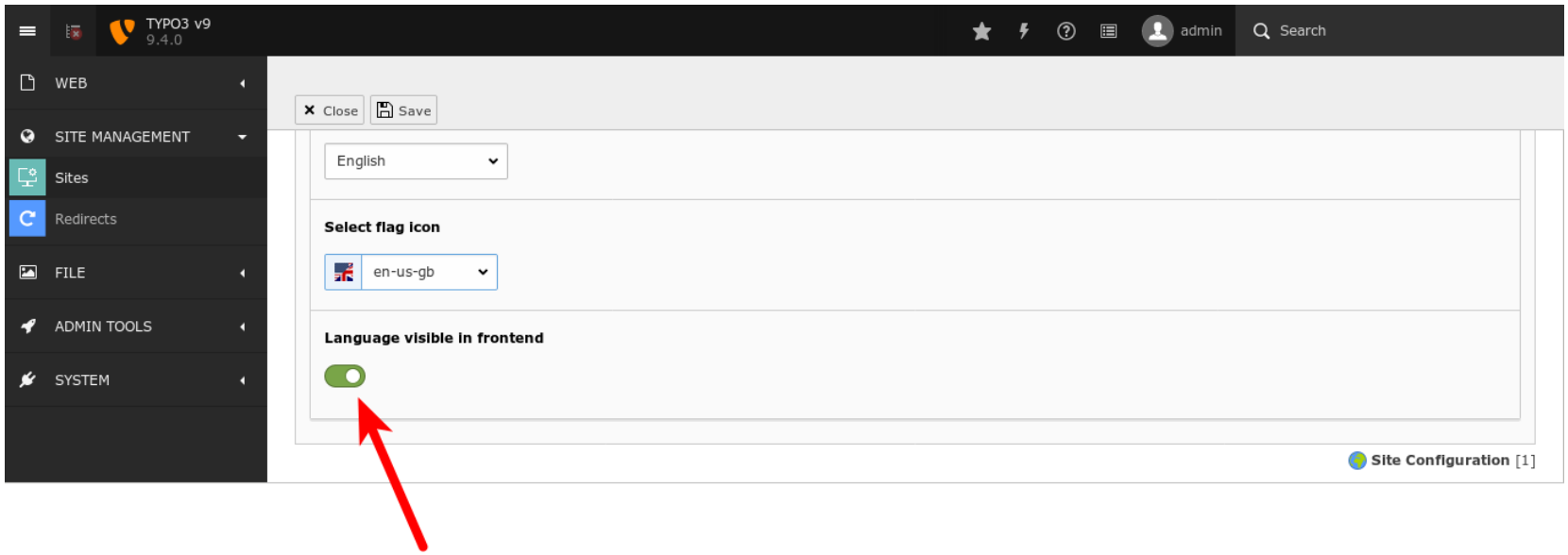
Pages feature a "description" field (under tab "Notes") now, which allows users to add notes. Other backend users can see/edit these.



Backend User Interface

Defined Languages Only

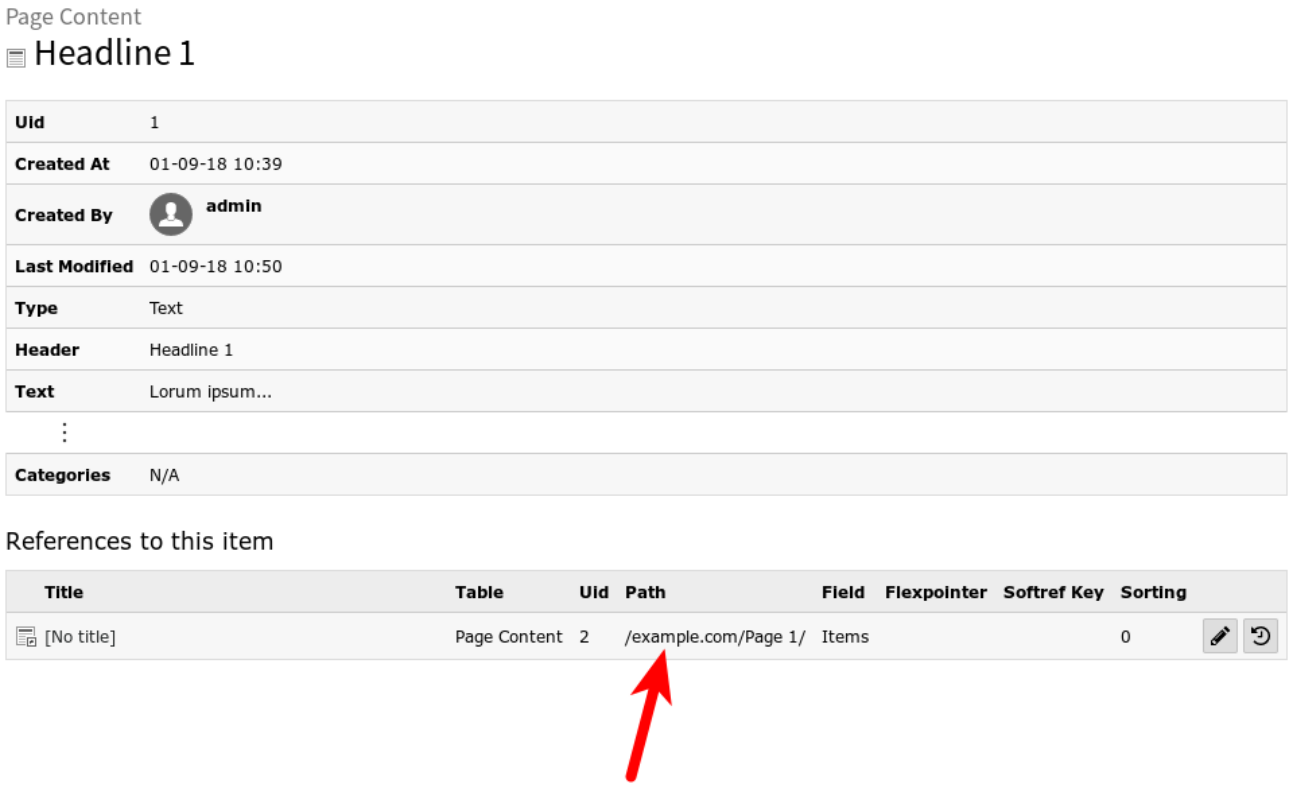
Website languages in the backend are now restricted to the languages defined under "Site Management → Site Configuration → Languages". Each language can be enabled/disabled.



Backend User Interface

Page Path in Record Info

Details about the reference of records now include the path in the pagetree.



Backend User Interface

Page-based URL Handling

TYPO3 supports page-based URL handling out-of-the-box now.

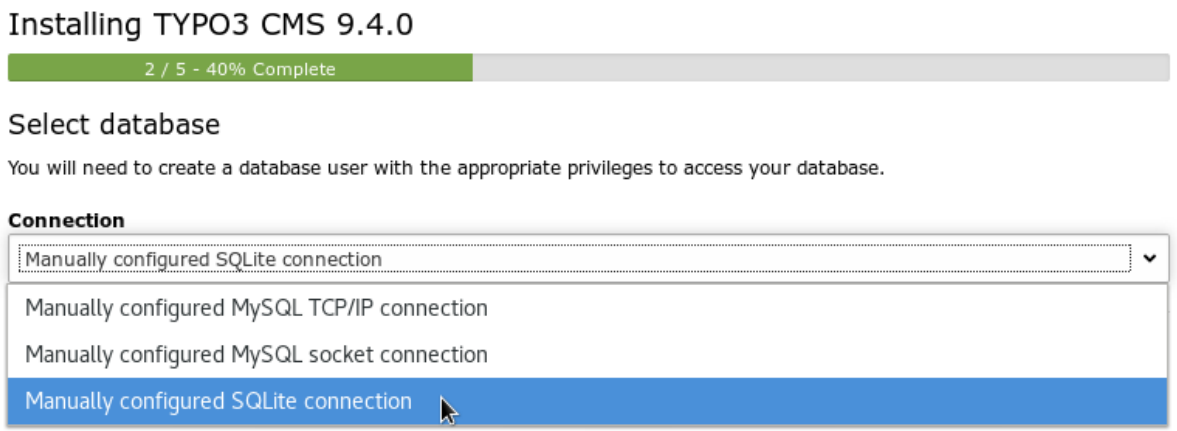
Chapter 2:

Changes for Integrators

Changes for Integrators

Install TYPO3 on SQLite (1)

- TYPO3 now supports SQLite, a self-contained, lightweight open source SQL database engine
- SQLite can be selected during the web-based installation process, if PHP module "pdo_sqlite" is installed and enabled:



Changes for Integrators

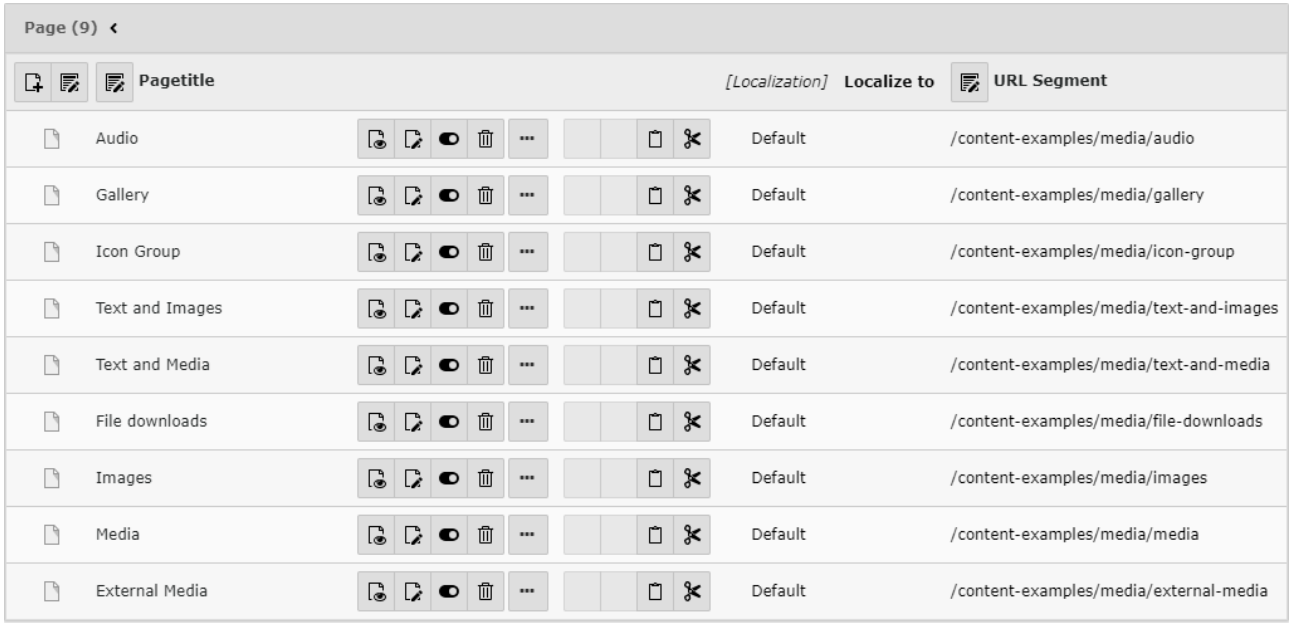
Install TYPO3 on SQLite (2)

- Database is stored in a single file, which means, TYPO3 instances can now run natively in PHP, including the data storage
- Using SQLite makes sense for relatively small TYPO3 sites or for test and development instances for example
- System administrators should take appropriate actions to protect the `*.sqlite` file from unauthorized access, if the file is stored inside the web container (depends on type of setup)

Changes for Integrators

Page-based URL Handling

- All links generated in the backend and frontend use this field, if set
- Page-based URL Handling requires a Site Configuration



Changes for Integrators

Page-based URL Handling

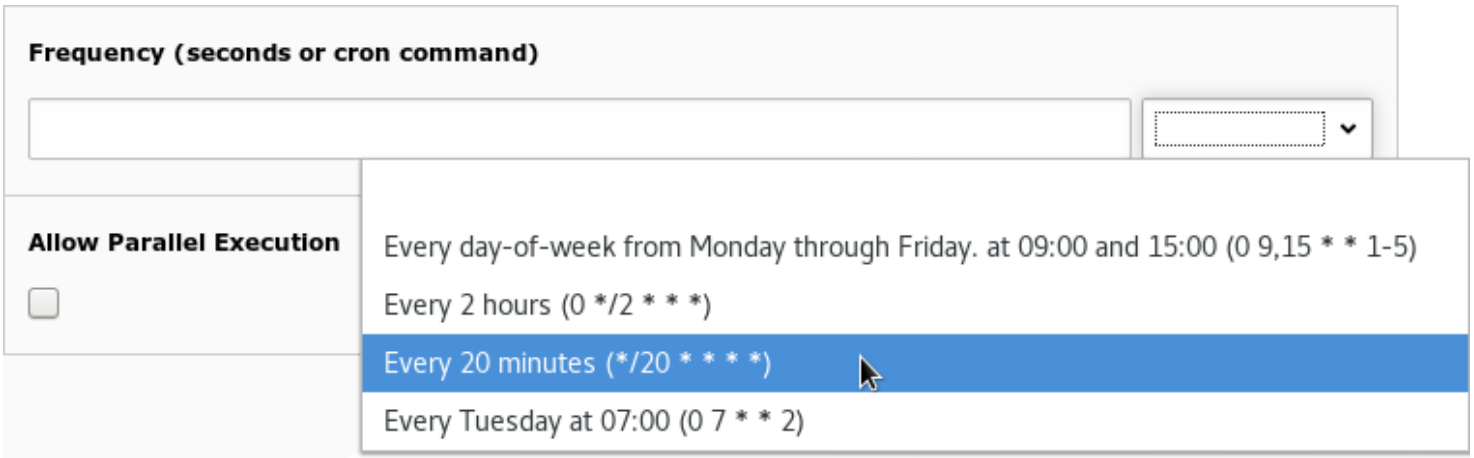
- New TCA field type `slug` has been added
- Define parts of a URL path to generate and resolve URLs

```
'type' => 'slug',
  'config' => [
    'generatorOptions' => [
      'fields' => ['title', 'nav_title'],
      'fieldSeparator' => '/',
      'prefixParentPageSlug' => true
    ]
    'fallbackCharacter' => '-',
    'eval' => 'uniqueInSite'
  ]
```

Changes for Integrators

Scheduler

- Presets have been added to the Scheduler:
  - `0 9,15 * * 1-5` (Mon to Fri at 9:00 and 15:00)
  - `0 */2 * * *` (every 2 hours)
  - `*/20 * * * *` (every 20 minutes)
  - `0 7 * * 2` (every Tuesday at 7:00)



Changes for Integrators

TypoScript Changes/Improvements (1)

- Attribute `async` is now assigned to the script tag of concatenated JS files, if all files have attribute `async` enabled in TypoScript:

```
config.concatenateJs = 1
page = PAGE
page.includeJSFooter {
  test = path/to/file.js
  test.async = 1
}
```

- It is now possible to read environment variables in TypoScript:

```
# Define default value
myConstant = defaultValue
# Enable overriding by environment variable
myConstant := getEnv(MYCONSTANT)
```

Changes for Integrators

TypoScript Changes/Improvements (2)

- The new Context API (see section "Changes for Developers") allows integrators to also use this in TypoScript
- For example:

```
10 = TEXT
10.data = context:workspace:id
```

- Syntax is: `context:[aspectName]:[propertyName]`
- Arrays are converted to comma-separated lists automatically
  (ideal for reading details on user groups for example)

Changes for Integrators

TypoScript Changes/Improvements (3)

- With new site-based handling, the de-facto standard `GET` parameter "L" became obsolete
- New parameter `typolink.language` has been introduced

```
page.10 = TEXT
page.10.value = Link to the page with the ID in the current
    language
page.10.typolink.parameter = 23
page.20 = TEXT
page.20.value = Link to the page with the ID in the language 3
page.20.typolink.parameter = 23
page.20.typolink.language = 3
```

Changes for Integrators

Simulate User under BE User Settings

- Administrator users had the option to switch to a different backend user ("User Settings → Simulate backend user")
- This function has been removed now

Changes for Integrators

Conditional Variants in `EXT:form` (1)

- New feature for extension "Forms": *conditional variants*
- Variants can contain conditions and allow changing properties of a form element
- This way, it becomes possible to manipulate form element values, validator and finisher options, etc. based on conditions

Changes for Integrators

Conditional Variants in `EXT:form` (2)

- Some typical use cases are for example:
  - Translate form element values depending on the current frontend language
  - Set and remove validators depending on the value of another form element
  - Set finisher values depending on the value of a form element.
  - Hide a form element in certain finishers and on the summary page.
  - Hide entire pages in the workflow depending on the value of a form element.
  - etc.
- Official documentation contains further details and examples

Changes for Integrators

HTML5 validation in Backend Fields

- HTML5 specific field types and attributes are now rendered by the FormEngine in the TYPO3 backend
- This includes email and numbers, incl. range config, for example
- HTML tag attributes are based on the `eval` TCA configuration
- This feature will possibly make custom JavaScript-based processing obsolete in the long term

Changes for Integrators

Workspace CLI Commands

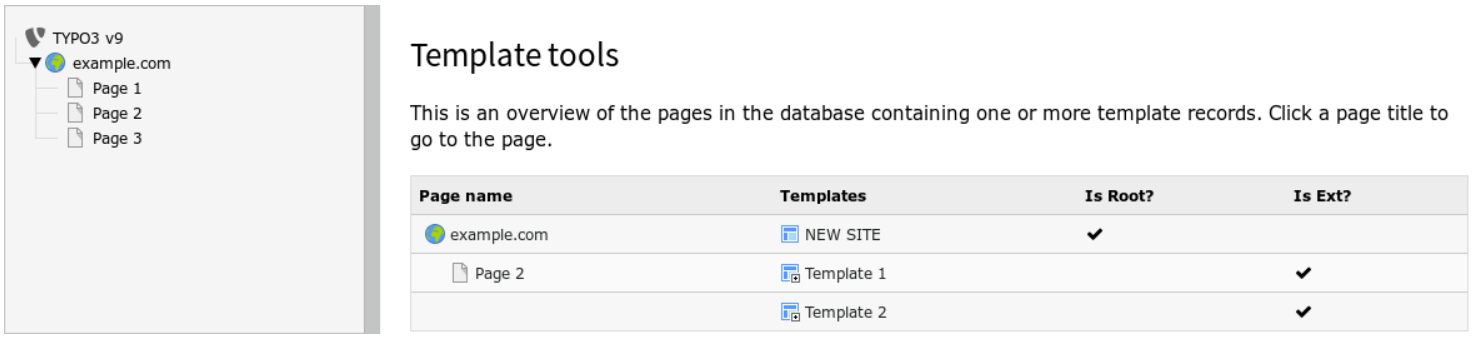
- TYPO3 now supports two new symfony-based CLI commands to trigger regular tasks:
  - `workspace:autopublish`
    Checks for workspaces with auto-publishing configured and does a publishing/swapping process.
  - `cleanup:previewlinks`
    Removes expired previewlinks stored within sys_preview from the database.
- Command line execution, for example:

```
$ typo3/sysext/core/bin/typo3 workspace:autopublish
```

Changes for Integrators

TypoScript Module Information

- Overview of TypoScript templates on root page reworked
- HTML output uses Fluid templates now
- Information shown include page name, template name (with direct link to edit the TypoScript record), state (by icon), is root or extension template



Changes for Integrators

Extension Manager

- Extensions, older than 10 November 2015 (TYPO3 v7 LTS) are excluded from the extension list import
- This reduces the database table size by approx. 75%

Chapter 3:

Changes for Developers

Changes for Developers

Context API

- A new Context API has been introduced in TYPO3 version 9.4
- The main goal of this concept is to centralize global variables
- The API aims to replace globally available objects (e.g. `TSFE`, `sys_page`, `BE_USER`, etc.) and to make them available in a common, structured and logical way
- Instead of exposing a full object (e.g. the `BE_USER` object), "aspects" contain properties, which are relevant and required only
- Extension developers can add aspects to the current context
- See documentation for further details and examples how to use the API

Changes for Developers

Customizing CSV Files

- When exporting database records as CSV, the output can be manipulated before the download starts
- The following two new hooks allow developers to achieve this:
  - `customizeCsvHeader` - to customize the header
  - `customizeCsvRow` - to customize a single row
- Usage example:

```
use \TYPO3\CMS\Recordlist\RecordList\DatabaseRecordList;

$hookName = DatabaseRecordList::class;
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS'][$hookName]['customizeCsvRow'][] =
  \Vendor\ExtName\Hooks\CsvTest::class . '->customizeCsvRow';
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS'][$hookName]['customizeCsvHeader'][] =
  \Vendor\ExtName\Hooks\CsvTest::class . '->customizeCsvHeader';
```

Changes for Developers

Symfony Expression Language

- The Symfony Expression Language has been moved from `EXT:form` into the TYPO3 core
- By moving it into the core, the expression language is now also available in other scopes
- The TYPO3 core features class `DefaultProvider`, which can be used directly (see example below) and custom implementations can extend class `AbstractProvider`

```
use \TYPO3\CMS\Core\ExpressionLanguage\DefaultProvider;
use \TYPO3\CMS\Core\ExpressionLanguage\Resolver;

$provider = GeneralUtility::makeInstance(DefaultProvider::class);
$conditionResolver = GeneralUtility::makeInstance(Resolver::class, $provider);
$conditionResolver->evaluate('1 < 2'); // result is true
```

Changes for Developers

"Currency" ViewHelper

- A dash can be used instead of decimal `00` in the Currency ViewHelper
- Option `useDash="1"` enables this feature
- The following example outputs `123.-`

```
<f:format.currency useDash="1">123.00</f:format.currency>
```

Changes for Developers

Changes in `PageRepository` Class

- The following methods have been marked *internal*:
  - `TYPO3\CMS\Frontend\Page\PageRepository::getMovePlaceholder()`
  - `TYPO3\CMS\Frontend\Page\PageRepository::movePlhOL()`
- The following class has been moved from the TYPO3 core to the system extension `lowlevel`:
  - `TYPO3\CMS\Core\Integrity\DatabaseIntegrityCheck`

Changes for Developers

Method `GeneralUtility::explodeUrl2Array()`

- Second argument of method
  `TYPO3\CMS\Core\Utility\GeneralUtility::explodeUrl2Array()`
  should not be used anymore (marked deprecated)
- If set, the string of the 1st argument was parsed into a multi-dimensional array if square brackets are used in variable names
- This can be simplified by using PHP's native `parse_str()` method:

```
$result = [];
parse_str($queryParametersAsString, $result);
```

Changes for Developers

Unit Tests

- TYPO3 core unit tests do not trigger an `E_NOTICE` anymore
- As a consequence, flag `$suppressNotices = true` has been removed and the testing-framework raised to version 4.4.0

Chapter 4:

Deprecated/Removed Functions

Deprecated/Removed Functions

Deprecated Options and Functions (1)

- The following two TypoScript options have been marked deprecated:
  - `config.enableConcatenateFiles`
  - `config.concatenateJsAndCss`

  The latter has been replaced by `concatenateCss` and `concatenateJs` respectively
- The following methods/classes have been marked deprecated:
  - `TYPO3\CMS\Backend\Utility\BackendUtility::shortcutExists()`
  - `TYPO3\CMS\Core\Utility\GeneralUtility::clientInfo()`
  - `TYPO3\CMS\Core\Utility\GeneralUtility::getHostName()`
  - `TYPO3\CMS\Core\Utility\GeneralUtility::unQuoteFilenames()`
  - `TYPO3\CMS\Frontend\Page\PageRepository::getRecordsByField()`
  - `TYPO3\CMS\Frontend\Page\PageRepository::getFileReferences()`
  - `TYPO3\CMS\Core\Database\PdoHelper`

Deprecated/Removed Functions

Deprecated Options and Functions (2)

- The following methods/classes have been marked deprecated:
  - `TYPO3\CMS\Backend\Template\ModuleTemplate::icons()`
  - `TYPO3\CMS\Backend\Template\ModuleTemplate::loadJavascriptLib()`
- The functionality of the following class has been replaced by a request to `index.php?loginRefresh=1` directly (and the class marked deprecated):
  - `TYPO3\CMS\Backend\Controller\LoginFramesetController`

Deprecated/Removed Functions

Deprecated Constants (1)

- The following constants have been deprecated (1/2) and should not be used anymore:
  - `TYPO3_URL_MAILINGLISTS`
  - `TYPO3_URL_DOCUMENTATION`
  - `TYPO3_URL_DOCUMENTATION_TSREF`
  - `TYPO3_URL_DOCUMENTATION_TSCONFIG`
  - `TYPO3_URL_CONSULTANCY`
  - `TYPO3_URL_CONTRIBUTE`
  - `TYPO3_URL_SECURITY`
  - `TYPO3_URL_DOWNLOAD`
  - `TYPO3_URL_SYSTEMREQUIREMENTS`

Deprecated/Removed Functions

Deprecated Constants (2)

- The following constants have been deprecated (2/2) and should not be used anymore:
  - `NUL` (use `"\0"` instead)
  - `TAB` (use `"\t"` instead)
  - `SUB` (use `chr(26)` instead)
  - `PATH_thisScript` (use `Environment::getCurrentScript()` instead)
  - `PATH_site` (use `Environment::getPublicPath().'/'` instead)

Deprecated/Removed Functions

Class `EidUtility` and Methods within `TSFE`

- The following class has been marked deprecated:

  `TYPO3\CMS\Frontend\Utility\EidUtility`

- The following methods have been marked deprecated:

  `TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController`

  - `TypoScriptFrontendController::initFEuser()`
  - `TypoScriptFrontendController::storeSessionData()`
  - `TypoScriptFrontendController::previewInfo()`
  - `TypoScriptFrontendController::hook_eofe()`
  - `TypoScriptFrontendController::addTempContentHttpHeaders()`
  - `TypoScriptFrontendController::sendCacheHeaders()`

- The following hook has been marked deprecated:

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

Deprecated/Removed Functions

Class `ReflectionService`

- The following methods have been marked deprecated:

  `TYPO3\CMS\Extbase\Reflection\ReflectionService`

  - `ReflectionService::getClassTagsValues()`
  - `ReflectionService::getClassTagValues()`
  - `ReflectionService::getClassPropertyNames()`
  - `ReflectionService::hasMethod()`
  - `ReflectionService::getMethodTagsValues()`
  - `ReflectionService::getMethodParameters()`
  - `ReflectionService::getPropertyTagsValues()`
  - `ReflectionService::getPropertyTagValues()`
  - `ReflectionService::isClassTaggedWith()`
  - `ReflectionService::isPropertyTaggedWith()`

Deprecated/Removed Functions

More functions
have been removed or marked deprecated
in TYPO3 version 9.4.

Please see the TYPO3 documentation for further details.

Chapter 5:

Miscellaneous

Miscellaneous

System Extension "Salted Passwords"

- System extension `EXT:saltedpasswords` is a mandatory requirement since TYPO3 version 6.2
- It has been refactored and merged into the TYPO3 core now
- This involved several tasks:
  - Password bulk update task removed from Scheduler
  - Salted passwords configuration options removed
  - Hook in "felogin" removed when user resets password
  - authUserBE/authUserFE moved to default authentication service
  - Password hash configuration in Install Tool reworked
  - Related methods marked deprecated

Chapter 6:

Sources and Authors

Sources and Authors

Sources

TYPO3 News:

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

Release Infos:

- https://get.typo3.org/release-notes/9.x/TYPO3_CMS_9.4.0
- INSTALL.md and ChangeLog
- typo3/sysext/core/Documentation/Changelog/9.4/*

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:

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

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

Licensed under Creative Commons BY-NC-SA 3.0

TYPO3