

# TYPO3 CMS 7.3 – What's New

Summary of the new features, changes and improvements

Created by:

Patrick Lobacher and Michael Schams

15/June/2015

Creative Commons BY-NC-SA 3.0



# TYPO3 CMS 7.3 - What's New

---

## Chapter Overview

Introduction

Backend User Interface

TSconfig & TypoScript

In-Depth Changes

Extbase & Fluid

Deprecated/Removed Functions

Sources and Authors

## Introduction

## The Facts

# Introduction

---

## TYPO3 CMS 7.3 - The Facts

- Release date: 16 June 2015
- Release type: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Primary focus: Package Ecosystem, Composer and Extension Handling

# Introduction

---

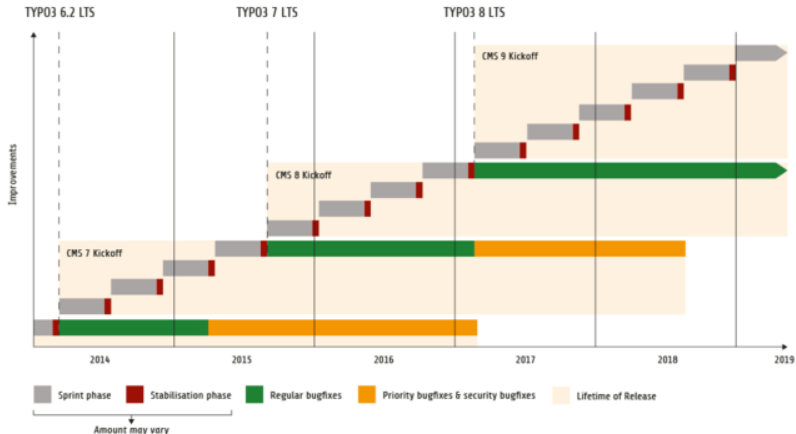
## System Requirements

- PHP\*: v5.5.0 - v5.6.x
- MySQL: v5.5.x - v5.6.x (no strict mode)
- Disk space: min 200 MB
- PHP settings:
  - `memory_limit`  $\geq$  128M
  - `max_execution_time`  $\geq$  240s
  - compilation option `-disable-ipv6` must not be used
- Backend requires IE  $\geq$  9 or any other modern browser

\* ) Further details: [PHP Minimum Requirements for TYPO3 CMS 7](#)

# Introduction

## Development And Release Timeline



# Introduction

---

## TYPO3 CMS Roadmap

Estimated release dates and their primary focus:

- v7.0 02/Dec/2014 Backend Overhaul Vol 1
- v7.1 24/Feb/2015 Core Cleanup & Streamlining
- v7.2 28/Apr/2015 Frontend
- **v7.3 16/Jun/2015 Package Ecosystem, Composer and Extension Handling**
- v7.4 04/Aug/2015 Backend Overhaul Vol 2
- v7.5 29/Sep/2015 *(to be determined...)*
- v7.6 xx/xxx/2015 **TYPO3 CMS 7 LTS** (Long Term Release)

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

<http://typo3.org/news/article/embrace-and-innovate-typo3-cms-7/>

# Introduction

---

## Installation

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

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/7.3
$ tar xzf typo3_src-7.3.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.3.0 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 `mlink` under Windows Vista and Windows 7



# Introduction

---

## Upgrade to TYPO3 CMS 7.x

- Upgrades only possible from TYPO3 CMS 6.2 LTS
- TYPO3 CMS < 6.2 should be updated to TYPO3 CMS 6.2 LTS first
- Upgrade instructions:  
[http://wiki.typo3.org/Upgrade#Upgrading\\_to\\_7.3](http://wiki.typo3.org/Upgrade#Upgrading_to_7.3)
- Official TYPO3 guide "TYPO3 Installation and Upgrading":  
<http://docs.typo3.org/typo3cms/InstallationGuide>
- General approach:
  - Check minimum system requirements (PHP, MySQL, etc.)
  - Review **deprecation\_\*.log** in old TYPO3 instance
  - Update all extensions to the latest version
  - Deploy new sources and run Install Tool → Upgrade Wizard
  - Review startup module for backend users (optionally)

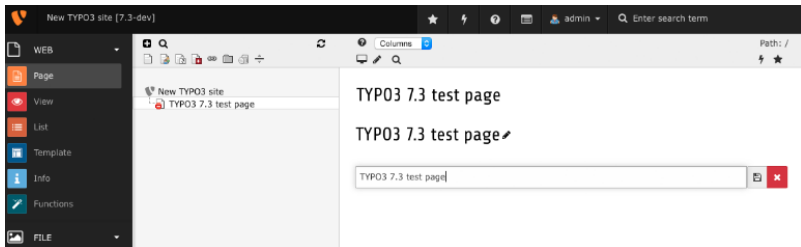
## Chapter 1: Backend User Interface

# Backend User Interface

---

## Page Title in Page- and List-Module

Users can edit page titles in the "Page" and "List" module by double-clicking the page header or the edit icon.



# Backend User Interface

---

## Install Tool: Delete Processed Files

In its "Clean up" section, the Install Tool provides a new function to remove processed files (e.g. image thumbnails) from FAL now.

This is useful if graphic-related settings have been changed or after an update of GraphicsMagick/ImageMagick to force all images to be regenerated.

### Clear processed files

The File Abstraction Layer stores a database record for every file it needs to process. (e.g. image thumbnails) In case you modified some graphics settings (All Configuration [GFX]) and you need all processed files to get regenerated, you can use this tool to remove the existing ones. The new processed files are created once they are needed.

Clear processed files

# Backend User Interface

---

## New Field in FAL Meta Data

The field "**Copyright**" has been added to the meta data of a FAL record (system extension: filemetadata).

Edit File Metadata "test.txt" on root level

The screenshot displays the 'Edit File Metadata' interface for a file named 'test.txt'. The interface is organized into several sections:

- General**: A single text input field for the **Creator**.
- Creator Tool** and **Publisher**: Two text input fields.
- Source** and **Copyright**: Two text input fields, with 'Copyright' being the newly added field.
- Geo Location**: Three text input fields for **Country**, **Region**, and **City**.

At the bottom right of the form, there is a button labeled 'File Metadata [6]'.

## Chapter 2: TSconfig & TypoScript

# TScnfig & TypoScript

---

## New stdWrap Function strtotime

- New TypoScript stdWrap property strtotime allows for conversion of formatted dates to Unix timestamps, e.g. to perform date calculations
- Valid values are 1 or any time string that is used as the first argument of the PHP function strtotime()

```
date_as_timestamp = TEXT
date_as_timestamp {
    value = 2015-04-15
    strtotime = 1
}
```

```
next_weekday = TEXT
next_weekday {
    data = GP:selected_date
    strtotime = + 2 weekdays
    strftime = %Y-%m-%d
}
```

# TScnfig & TypoScript

---

## GPmerged in Conditions

- Using GP in TypoScript conditions only returns the POST variable, if the request contains both, POST and GET variables
- New option GPmerged merges both methods and returns the result

```
[globalVar = GPmerged:tx_demo|foo = 1]
  page.90 = TEXT
  page.90.value = DEMO
[global]
```



# TScnfig & TypoScript

---

## New Options for stdWrap.case

- Options uppercamelcase and lowercamelcase have been added to stdWrap.case
- Example:

```
tt_content = CASE
tt_content {
    key.field = CType
    my_custom_ctype =< lib.userContent
    my_custom_ctype {
        file = EXT:site_base/Resources/Private/Templates/SomeOtherTemplate.html
        settings.extraParam = 1
    }
    default =< lib.userContent
    default {
        file = TEXT
        file.field = CType
        file.stdWrap.case = uppercamelcase
        file.wrap = EXT:site_base/Resources/Private/Templates/|.html
    }
}
```

# TScnfig & TypoScript

---

## Property integrity Added for JavaScript Files (1)

- Property integrity has been added to JavaScript file inclusions in order to specify a SRI hash to allow resource verification (SRI: Sub-Resource Integrity, see next slide)
- This affects the TypoScript PAGE properties `page.includeJSLibs`, `page.includeJSFooterlibs`, `includeJS` and `includeJSFooter`
- Example:

```
page {
  includeJS {
    jQuery = https://code.jquery.com/jquery-1.11.3.min.js
    jQuery.external = 1
    jQuery.disableCompression = 1
    jQuery.excludeFromConcatenation = 1
    jQuery.integrity = sha256-7LkWEzqTdpEfELxcZZ1S6wAx5Ff13zZ831Y02/ujj7g=
  }
}
```

# TSconfig & TypeScript

---

## Property integrity Added for JavaScript Files (2)

- SRI is a W3C specification that allows web developers to ensure that resources hosted on third-party servers have not been tampered with
- Generate integrity hashes:
  - Option 1: <https://srihash.org>
  - Option 2: use the following shell command

```
cat FILENAME.js | openssl dgst -sha256 -binary | openssl enc -base64 -A
```

- Read more:
  - <http://www.w3.org/TR/SRI/>

## Chapter 3: In-Depth Changes

# In-Depth Changes

---

## Symfony/Console Integration into CommandController (1)

The CommandController now makes use of Symfony/Console internally and provides various methods:

- TableHelper

- `outputTable($rows, $headers = NULL)`

- DialogHelper

- `select($question, $choices, $default = NULL, $multiSelect = false, $attempts = FALSE)`
  - `ask($question, $default = NULL, array $autocomplete = array())`
  - `askConfirmation($question, $default = TRUE)`
  - `askHiddenResponse($question, $fallback = TRUE)`
  - `askAndValidate($question, $validator, $attempts = FALSE, $default = NULL, array $autocomplete = NULL)`
  - `askHiddenResponseAndValidate($question, $validator, $attempts = FALSE, $fallback = TRUE)`

# In-Depth Changes

---

## Symfony/Console Integration into CommandController (2)

- ProgressHelper
  - `progressStart($max = NULL)`
  - `progressSet($current)`
  - `progressAdvance($step = 1)`
  - `progressFinish()`

(see next slides for code examples)

# In-Depth Changes

---

## Symfony/Console Integration into CommandController (3)

```
<?php
namespace Acme\Demo\Command;
use TYPO3\CMS\Extbase\Mvc\Controller\CommandController;

class MyCommandController extends CommandController {
    public function myCommand() {

        // render a table
        $this->output->outputTable(array(
            array('Bob', 34, 'm'),
            array('Sally', 21, 'f'),
            array('Blake', 56, 'm')
        ),
        array('Name', 'Age', 'Gender'));

        // select
        $colors = array('red', 'blue', 'yellow');
        $selectedColorIndex = $this->output->select('Please select one color', $colors, 'red');
        $this->outputLine('You choose the color %s.', array($colors[$selectedColorIndex]));

        [...]
    }
}
```

# In-Depth Changes

---

## Symfony/Console Integration into CommandController (4)

```
[...]  
// ask  
$name = $this->output->ask('What is your name?' . PHP_EOL, 'Bob', array('Bob', 'Sally', 'Blake'));  
$this->outputLine('Hello %s.', array($name));  
  
// prompt  
$likesDogs = $this->output->askConfirmation('Do you like dogs?');  
if ($likesDogs) {  
    $this->outputLine('You do like dogs!');  
}  
  
// progress  
$this->output->progressStart(600);  
for ($i = 0; $i < 300; $i++) {  
    $this->output->progressAdvance();  
    usleep(5000);  
}  
$this->output->progressFinish();  
}  
}  
?>
```



# In-Depth Changes

---

## Backend Login API (1)

- Backend login has been completely refactored and a new API has been introduced
- The OpenID form has been extracted and is now using the new API (makes it independent of the central Core classes)
- The concept of the new backend login is based on "login providers", which can be registered in file `ext_localconf.php` as follows:

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['backend']['loginProviders'][1433416020] = [  
    'provider' => \TYPO3\CMS\Backend>LoginProvider\UsernamePasswordLoginProvider::class,  
    'sorting' => 50,  
    'icon-class' => 'fa-key',  
    'label' => 'LLL:EXT:backend/Resources/Private/Language/locallang.xlf:login.link'  
];
```

# In-Depth Changes

---

## Backend Login API (2)

- Options are defined as follows:
  - provider:  
login provider class name, which must implement  
`TYPO3\CMS\Backend\LoginProvider\LoginProviderInterface`
  - sorting:  
ordering of the links to the possible login providers on the login screen
  - icon-class:  
font-awesome icon name for the link on the login screen
  - label:  
label for the login provider link on the login screen

# In-Depth Changes

---

## Backend Login API (3)

- The `LoginProviderInterface` only contains method 

```
public function render(StandaloneView $view, PageRenderer $pageRenderer, LoginController $loginController);
```
- Parameters are defined as follows:
  - `$view`:  
Fluid `StandaloneView` which renders the login screen. You have to set the template file and you may add variables to the view according to your needs.
  - `$pageRenderer`:  
`PageRenderer` instance provides possibility to add necessary JavaScript resources.
  - `$loginController`:  
`LoginController` instance.

# In-Depth Changes

---

## Backend Login API (4)

- Template must contain `<f:layout name="Login">` and `<f:section name="loginFormFields">` (for form fields):

```
<f:layout name="Login" />
<f:section name="loginFormFields">
  <div class="form-group t3js-login-openid-section" id="t3-login-openid_url-section">
    <div class="input-group">
      <input type="text" id="openid_url"
        name="openid_url"
        value="{presetOpenId}"
        autofocus="autofocus"
        placeholder="{f:translate(key: 'openid', extensionName: 'openid')}}"
        class="form-control input-login t3js-clearable t3js-login-openid-field" />
      <div class="input-group-addon">
        <span class="fa fa-openid"></span>
      </div>
    </div>
  </div>
</f:section>
```

# In-Depth Changes

---

## CategoryRegistry with New Options

- Method `CategoryRegistry->addTcaColumn` received options to set `l10n_mode` and `l10n_display`:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable(  
    $extensionKey,  
    $tableName,  
    'categories',  
    array(  
        'l10n_mode' => 'string (keyword)',  
        'l10n_display' => 'list of keywords'  
    )  
);
```

# In-Depth Changes

---

## Sprites in Backend Modules

- Backend modules (main modules such as "Web" as well as submodules such as "Filelist") can now use sprites as icons (only sprite icons known to TYPO3 are available!)
- Example:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::addModule(  
    'web',  
    'layout',  
    'top',  
    \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extPath($_EXTKEY) . 'Modules/Layout/',  
    array(  
        'script' => '_DISPATCH',  
        'access' => 'user,group',  
        'name' => 'web_layout',  
        'configuration' => array('icon' => 'module-web'),  
        'labels' => array(  
            'll_ref' => 'LLL:EXT:cms/layout/locallang_mod.xlf',  
        ),  
    ),  
);
```

# In-Depth Changes

---

## FormEngine NodeFactory API (1)

- It is now possible to register new nodes and overwriting existing nodes

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['formEngine']['nodeRegistry'][1433196792] = array(  
    'nodeName' => 'input',  
    'priority' => 40,  
    'class' => \MyVendor\MyExtension\Form\Element\T3editorElement::class  
);
```

- Example above registers a new class `MyVendor\MyExtension\Form\Element\T3editorElement` as render class for TCA type `input`, which must implement the interface `TYPO3\CMS\Backend\Form\NodeInterface`
- The array key is the Unix timestamp of the date when a registry element is added

# In-Depth Changes

---

## FormEngine NodeFactory API (2)

- In cases where multiple registry elements have been registered for the same type, the resolver with the highest priority (0 to 100) is used
- A new TCA type can be registered as follows:

TCA

```
'columns' => array(  
    'bodytext' => array(  
        'config' => array(  
            'type' => 'text',  
            'renderType' => '3dCloud',  
        ),  
    ),  
)
```

ext\_localconf.php

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['formEngine']['nodeRegistry'][1433197759] = array(  
    'nodeName' => '3dCloud',  
    'priority' => 40,  
    'class' => \MyVendor\MyExtension\Form\Element\ShowTextAs3dCloudElement::class  
);
```



# In-Depth Changes

---

## Signal `postProcessMirrorUrl`

- Signal `postProcessMirrorUrl` has been moved to a new class

### **BREAKING CHANGE!**

- The following code example takes the TYPO3 version into account:

```
$signalSlotDispatcher->connect(  
    version_compare(TYPO3_version, '7.0', '<')  
    ? 'TYPO3\\CMS\\Lang\\Service\\UpdateTranslationService'  
    : 'TYPO3\\CMS\\Lang\\Service\\TranslationService',  
    'postProcessMirrorUrl',  
    'Vendor\\Extension\\Slots\\CustomMirror',  
    'postProcessMirrorUrl'  
);
```

## Chapter 4: Extbase & Fluid

## ActionMenuItemGroupViewHelper (1)

- Using this ViewHelper, option groups can be used in the backend select field, which controls which action is selected
- Example:

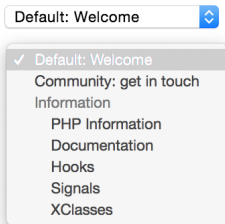
```
<f:be.menus.actionMenu>
  <f:be.menus.actionMenuItem label="Default: Welcome" controller="Default" action="index" />
  <f:be.menus.actionMenuItem label="Community: get in touch" controller="Community"
    action="index" />
  <f:be.menus.actionMenuItemGroup label="Information">
    <f:be.menus.actionMenuItem label="PHP Information" controller="Information"
      action="listPhpInfo" />
    <f:be.menus.actionMenuItem label="Documentation" controller="Information"
      action="documentation" />
    <f:be.menus.actionMenuItem label="Hooks" controller="Information" action="hooks" />
    <f:be.menus.actionMenuItem label="Signals" controller="Information" action="signals" />
    <f:be.menus.actionMenuItem label="XClasses" controller="Information" action="xclass" />
  </f:be.menus.actionMenuItemGroup>
</f:be.menus.actionMenu>
```

# Extbase & Fluid

---

## ActionMenuItemGroupViewHelper (2)

- Example on previous slide results in the following output:



## Template Support for FlashMessagesViewHelper

- The FlashMessagesViewHelper supports templates now
- New attribute `as` allows to specify a variable name, which can be used within the ViewHelper's child elements to access the flash messages
- Example:

```
<f:flashMessages as="flashMessages">
  <ul class="myFlashMessages">
    <f:for each="{flashMessages}" as="flashMessage">
      <li class="alert {flashMessage.class}">
        <h4>{flashMessage.title}</h4>
        <span class="fancy-icon">{flashMessage.message}</span>
      </li>
    </f:for>
  </ul>
</f:flashMessages>
```

- Note: option `renderMode` is deprecated now

# Extbase & Fluid

---

## New Properties of cObject FLUIDTEMPLATE (1)

- cObject FLUIDTEMPLATE has been extended with `templateRootPaths` and `templateName`
- It is possible to set a template name and when rendering the template this name is used together with the set format to find the template in the given `templateRootPaths`
- `templateRootPaths` features the same fallback logic as `layoutRootPath` and `partialRootPath`
  - `templateName`: string/stdWrap
  - `templateRootPaths`: array of file paths with "EXT:" prefix support

## New Properties of cObject FLUIDTEMPLATE (2)

### ■ TypeScript Example:

```
lib.stdContent = FLUIDTEMPLATE
lib.stdContent {
    templateName = TEXT
    templateName.stdWrap {
        cObject = TEXT
        cObject {
            data = levelfield:-2,backend_layout_next_level,slide
            override.field = backend_layout
            split {
                token = frontend__
                1.current = 1
                1.wrap = |
            }
        }
    }
    ifEmpty = Default
}
templateRootPaths {
    10 = EXT:frontend/Resources/Private/Templates
    20 = EXT:sitemodification/Resources/Private/Templates
}
}
```

# Extbase & Fluid

---

## Removal of `xmlns`-Attributes and HTML Tags (1)

- With the introduction of using `xmlns:*` attributes to include ViewHelpers, it is possible to have IDE support for Fluid templates. The Problem is that the `xmlns:*` attributes and the corresponding tag will also be rendered, which is usually not desired.
- The workaround is to use sections, but this solution is counter-intuitive and not available in layouts. It also causes extra processing overhead.
- `xmlns:*` attributes for valid ViewHelper namespaces will now be removed before rendering, if they show the following syntax:  
`http://typo3.org/ns/<phpNamespace>`  
(`xmlns` attributes for non-ViewHelper namespaces are preserved)



# Extbase & Fluid

---

## Removal of xmlns-Attributes and HTML Tags (2)

- Include ViewHelper namespaces within the HTML tag and the data-namespace-typo3-fluid="true" attribute to prevent the rendering of the entire HTML tag

```
<html data-namespace-typo3-fluid="true"
  xmlns:f="http://typo3.org/ns/TYPO3/CMS/Fluid/ViewHelpers"
  xmlns:n="http://typo3.org/ns/GeorgRinger/News/ViewHelpers">

  <f:if condition="{newsItem.title}">
    <f:then>
      <n:titleTag>{newsItem.title}</n:titleTag>
    </f:then>
    <f:else>
      <n:titleTag>News-Detail</n:titleTag>
    </f:else>
  </f:if>

</html>
```

## New Methods in Fluid-StandaloneView

- StandaloneView is extended with `setTemplateRootPaths($templatePaths)` and `setTemplate($templateName, $throwException = TRUE)`
- Same functionality as cObject FLUIDTEMPLATE
- Example (render of an email template):

```
$view = GeneralUtility::makeInstance(StandaloneView::class);  
$view->setLayoutRootPaths(array(GeneralUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Layouts')));  
$view->setPartialRootPaths(array(GeneralUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Partials')));  
$view->setTemplateRootPaths(array(GeneralUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Templates')));  
$view->setTemplate('Email/Notification');  
$emailBody = $view->render();
```

## Data Processing for FLUIDTEMPLATE cObject (1)

- cObject FLUIDTEMPLATE has been extended with `dataProcessing`
- This setting can be used to add one or multiple processors to manipulate the `$data` variable of the currently rendered cObject (e.g. `tt_content` or `page`)
- Processor must implement the interface `FluidTemplateDataProcessorInterface` and contain the following method:

```
function process(array &$data, array $processorConfiguration,  
    array $configuration, StandaloneView $view) {  
    [...]  
}
```

## Data Processing for FLUIDTEMPLATE cObject (2)

### ■ Example:

```
my_custom_ctype = FLUIDTEMPLATE
my_custom_ctype {
    templateRootPaths {
        10 = EXT:your_extension_key/Resources/Private/Templates
    }
    templateName = CustomName
    settings {
        extraParam = 1
    }
    dataProcessing {
        1 = Vendor\YourExtensionKey\DataProcessing\MyFirstCustomProcessor
        2 = AnotherVendor\AnotherExtensionKey\DataProcessing\MySecondCustomProcessor
        2 {
            options {
                myOption = SomeValue
            }
        }
    }
}
```

## Chapter 5: Deprecated/Removed Functions

# Deprecated/Removed Functions

---

## FormEngine Refactoring

### TCA:

- Options `_PADDING`, `_VALIGN` and `DISTANCE` have been removed from `TCA['aTable']['columns']['aField']['config']['wizards']`
- Key `TCA['aTable']['ctrl']['mainPalette']` has been removed

### TScnfig:

- Keys `mod.web_layout.tt_content.fieldOrder` and `TCEFORM.aTable.aField.linkTitleToSelf` have been removed

### Hooks:

- Hooks use key type instead of `form_type` now
- Hook `getSingleFieldClass` has been removed

# Deprecated/Removed Functions

---

## Removal of IdentityMap from Extbase Persistence

- Class IdentityMap has been removed from Extbase persistence (a ReflectionException is thrown if it is still used)
- Accessing the previously existing IdentityMap properties within DataMapper and Repository will fail now and the creation of IdentityMap instances is not possible anymore
- Use "Sessions" persistence instead:

```
$session = GeneralUtility::makeInstance(ObjectManager::class)->get(
    \TYPO3\CMS\Extbase\Persistence\Generic\Session::class
);

$session->registerObject($object, $identifier);

if($session->hasIdentifier($identifier)) {
    $object = $session->getObjectByIdentifier($identifier, $className);
}
```

# Deprecated/Removed Functions

---

## Miscellaneous (1)

- File `typo3conf/extTables.php` is deprecated.  
Use the following file instead:  
`<your_extension>/Configuration/TCA/Overrides/pages.php`
- Configuration `$TYPO3_CONF_VARS[GFX][png_to_gif]` has been removed
- In TYPO3 CMS installations, which do not have extension `rsaauth` installed, BE login passwords are transferred in plain text now (solution: install extension `rsaauth` or use HTTPS for the BE)
- Method `exec_SELECTgetRows()` validates parameter `$uidIndexField` now. If the specified field is not present in the database result, an `InvalidArgumentException` is thrown.



# Deprecated/Removed Functions

---

## Miscellaneous (2)

- DBAL option `config.classFile` has been removed
- Options `iconOnly` and `styleAttributes` of `CshViewHelper` are marked as deprecated now
- TypoScript option `page.bgImg` is deprecated now
- Method `isEnabled()` of class `T3editor` is deprecated now
- Old TYPO3 `ClassLoader` has been removed in favor of `ComposerClassLoader`

## Chapter 6: Sources and Authors

# Sources and Authors

---

## Sources

### TYPO3 News:

- <http://typo3.org/news>

### Release Infos:

- [http://wiki.typo3.org/TYPO3\CMS\\_7.3.0](http://wiki.typo3.org/TYPO3\CMS_7.3.0)
- [INSTALL.md](#) and [Changelog](#)
- `typo3/sysexst/core/Documentation/Changelog/7.3/*`

### TYPO3 Bug-/Issuetracker:

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

### TYPO3 Git Repositories:

- <https://git.typo3.org/Packages/TYPO3.CMS.git>
- <https://git.typo3.org/Packages/TYPO3.Fluid.git>

# Sources and Authors

---

## **TYPO3 CMS What's New Slides:**

Patrick Lobacher

(Research, Information Gathering and German Version)

Michael Schams

(Project Leader and English Version)

### **Translations by:**

Andrey Aksenov, Sergio Catala, Jigal van Hemert, Michel Mix, Sinisa Mitrovic,  
Angeliki Plati, Nena Jelena Radovic, Roberto Torresani

<http://typo3.org/download/release-notes/whats-new>

Licensed under Creative Commons BY-NC-SA 3.0

