TYPO3 CMS 7.1 – What’s New
Summary of the new features, changes and improvements

Created by:
Patrick Lobacher and Michael Schams

26/February/2015
Creative Commons BY-NC-SA 3.0
Chapter Overview

Introduction
BackendUI
TSconfig & TypoScript

In-Depth Changes
Extbase & Fluid
Deprecated/Removed Functions
Sources and Authors
The Facts
Introduction

TYPO3 CMS 7.1 - The Facts

- Release date: 24 February 2015
- Release type: "Sprint Release"
- Vision: Embrace, Innovate, Deliver
- Primary focus: Core Cleanup and Streamlining
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 >= 128M
  - max_execution_time >= 240s
  - compilation option –disable-ipv6 must not be used
- **Backend requires IE >= 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 → 10/Mar/2015  Frontend
- v7.3 → 21/Apr/2015  Composer Ecosystem
- v7.4 → 09/Jan/2015  Backend Overhaul Vol 2
- v7.5 → 28/Jul/2015  (to be determined...)
- v7.6 → 13/Oct/2015 pre-LTS inferno
- v7.7 → xx/xx/2015  TYPO3 CMS 7 LTS (Long Term Release)

https://typo3.org/typo3-cms/roadmap/
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/7.1
  $ tar xzf typo3_src-7.1.0.tar.gz
  $ cd htdocs
  $ ln -s ../typo3_src-7.1.0 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 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_7.1](http://wiki.typo3.org/Upgrading_to_7.1)
- Official TYPO3 guide "TYPO3 Installation and Upgrading":
- 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
BackendUI

Look & Feel: Date/Time Picker

Date/Time Picker has been replaced with a Bootstrap alternative
"Create Pages" and "Sort Pages" moved to: WEB => Functions
(in TYPO3 CMS < 7.1, they were located under "WEB => Functions => Wizards")
BackendUI

Look & Feel: Access Module

Module Web => Access allows to leave owner/group unchanged when overwriting permissions

Permissions: EDIT

Owner
- leave unchanged -

Group
- leave unchanged -

<table>
<thead>
<tr>
<th>Role</th>
<th>Show page</th>
<th>Edit content</th>
<th>Edit page</th>
<th>Delete page</th>
<th>New pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Group</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Everybody</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Save  Abort
Look & Feel: Icons in List Module

Icons ("action buttons") in List module divided into two groups (primary actions first (read, update, delete), followed by secondary actions)
Chapter 2:

TSconfig & TypoScript
**TSconfig & TypoScript**

**StdWrap for page.headTag**

- **TypoScript setting** `page.headTag` has `stdWrap` functionality now

```typescript
page = PAGE
page.headTag = <head>
page.headTag.override = <head class="special">
page.headTag.override.if {
  isInList.field = uid
  value = 24
}
```
Include JavaScript files asynchronously

- JavaScript files can be loaded asynchronously

```typescript
page {
    includeJS {
        jsFile = /path/to/file.js
        jsFile.async = 1
    }
}
```

- This affects:
  - `includeJSlibs` / `includeJSLibs`
  - `includeJSFooterlibs`
  - `includeJS`
  - `includeJSFooter`
TSconfig & TypoScript

**HMENU item selection via `additionalWhere`**

- TypoScript cObject `HMENU` features a new property `additionalWhere`
- This allows for a more specific database query (e.g. filtering)

**Example:**

```plaintext
lib.authormenu = HMENU
lib.authormenu.1 = TMENU
lib.authormenu.1.additionalWhere = AND author!=""
```
Additional properties for **HMENU** browse menus

Two new properties for cObject **HMENU** (option "special=browse") to select menu items more fine-grained:

- excludeNoSearchPages
- includeNotInMenu

**Example:**

```plaintext```
lib.browsemenu = HMENU
lib.browsemenu.special = browse
lib.browsemenu.special.excludeNoSearchPages = 1
lib.browsemenu.includeNotInMenu = 1
```
Multiple HTTP headers

- HTTP headers can be set as an array (`config.additionalHeaders`)
- This allows for the configuration of multiple headers at the same time

```php
config.additionalHeaders {
    10 {
        # header string
        header = WWW-Authenticate: Negotiate

        # (optional) replace previous headers with the same name (default: 1)
        replace = 0

        # (optional) force HTTP response code
        httpResponseCode = 401
    }
    # set second additional HTTP header
    20.header = Cache-control: Private
}
```
Option "auto" added for config.absRefPrefix

TypoScript setting `config.absRefPrefix` can be used to allow URL rewriting. As an alternative to `config.baseURL` (to configure a specific domain), `absRefPrefix` can detect the site root automatically:

```plaintext
cfg.absRefPrefix = auto
```

```plaintext
# ...instead of:
[ApplicationContext = Production]
cfg.absRefPrefix = /

[ApplicationContext = Testing]
cfg.absRefPrefix = /my_site_root/
```

Note: The new option is also safe for multi-domain environments to avoid duplicate caching mechanism.
The handling of languages is done by records stored in DB table sys_language, which are usually referenced via sys_language_uid.

In TYPO3 CMS 7.1, the ISO 639-1 two-letter code has been introduced:
- New DB field: sys_language.language_isocode
- New TypoScript option: sys_language_isocode

Example:

# Danish by default
config.sys_language_uid = 0
config.sys_language_isocode_default = da

[globalVar = GP:L = 1]
# ISO code stored in table sys_language (uid 1)
config.sys_language_uid = 1
# overwrite ISO code as required
config.sys_language_isocode = fr

[GLOBAL]

page.10 = TEXT
page.10.data = TSFE:sys_language_isocode
page.10.wrap = <div class="main" data-language="|"/>
Custom TypoScript Conditions in Backend

- Support of custom conditions for the **frontend** has been introduced in TYPO3 CMS 7.0 already
- Since TYPO3 CMS 7.1, it is also possible to use custom conditions in the **backend**
- The condition must be derived from AbstractCondition and implement method `matchCondition()`
- Example usage in TypoScript:

  ```
  [BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition]
  [BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = 7]
  [BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = 7, != 6]
  [BigCompanyName\TypoScriptLovePackage\MyCustomTypoScriptCondition = {$mysite.myconstant}]
  ```
Customize icons via PageTSconfig

- Value/label pairs of select fields can be configured by PageTSconfig option `addItems` already.
- It is also possible to influence the icon of these fields now.
  - Option 1: by using `addItems` and sub-property `.icon`
  - Option 2: by using `altIcons` (all items in general)

Example:

```plaintext
TCEFORM.pages.doktype.addItems {
  10 = My Label
  10.icon = EXT:t3skin/icons/gfx/i/pages.gif
}
TCEFORM.pages.doktype.altIcons {
  10 = EXT:myext/icon.gif
}
```
TSconfig & TypoScript

Extend element browser with mount points

- New UserTSconfig option .append allows administrators to add mount points, rather than replacing the configured database mount points of the user
- Example:

  ```
  options.pageTree.altElementBrowserMountPoints = 20,31
  options.pageTree.altElementBrowserMountPoints.append = 1
  ```
TSconfig & TypoScript

Label override of checkboxes and radio buttons

- Labels of radio buttons and checkboxes can be overwritten now
- Example:

  // field with a single checkbox (use "\#default")
  TCEFORM.pages.hidden.altLabels.default = new label
  TCEFORM.pages.hidden.altLabels.default = LLL:path/to/languagefile.xlf:individualLabel

  // field with multiple checkboxes (0, 1, 2, 3...)\n  TCEFORM.pages.l18n_cfg.altLabels.0 = new label of first checkbox
  TCEFORM.pages.l18n_cfg.altLabels.1 = new label of second checkbox
  TCEFORM.pages.l18n_cfg.altLabels.2 = new label of third checkbox
  ...

TYPO3 CMS 7.1 - What’s New
TSconfig & TypoScript

Miscellaneous (1)

- Width and height of the Element Browser can be configured using UserTSconfig:
  
  ```
  options.popupWindowSize = 400x900
  options.RTE.popupWindowSize = 200x200
  ```

- PageTSconfig: new RTE configuration property can be used to configure a default target for links of a given type:
  
  ```
  buttons.link.[type].properties.target.default
  ```

Where `[type]` can be page, file, url, mail or spec (extensions may provide further types)
Section headlines of indexed search results are links by default. It is now possible to disable these links and display sections as simple texts:

```
plugin.tx_indexedsearch.linkSectionTitles = 0
```

getData can access field data now (not only arrays such as GPVar and TSFE):

```
10 = TEXT
10.data = field:fieldname|level1|level2
```

TypoScript setting `config.pageTitle` has `stdWrap` functionality now:

```
# make value of `<title>` upper case
page = PAGE
page.config.pageTitle.case = upper
```
Chapter 3:

In-Depth Changes
In-Depth Changes

TCA: Maximum chars in text element

- TCA type text now supports the HTML5 attribute `maxlength` to restrict the length of a text (note: line breaks are usually counted as two characters)

```php
'teaser' => array(
    'label' => 'Teaser',
    'config' => array(
        'type' => 'text',
        'cols' => 60,
        'rows' => 2,
        'max' => '30' // <-- maxlength
    )
),
```

Please note, that not every browser supports this attribute. See [Browser Support List](#) for details.
New SplFileInfo implementation

- New class: TYP03\CMS\Core\Type\File\FileInfo
- This class extends class SplFileInfo, which allows fetching meta information from files

```php
$fileIdentifier = '/tmp/foo.html';
$fileInfo = GeneralUtility::makeInstance(
    TYP03\CMS\Core\Type\File\FileInfo::class,
    $fileIdentifier
);
echo $fileInfo->getMimeType(); // output: text/html
```

- Custom implementations can use the following hook:

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['TYP03\CMS\Core\Type\File\FileInfo::class']['mimeTypeGuessers']
```
In-Depth Changes

UserFunc in TCA Display Condition

- userFunc displayCondition makes it possible to check on any imaginable condition or state
- If a situation cannot be evaluated with any of the existing checks, developers can develop their own user function (return TRUE/FALSE to show/hide appropriate TCA field)

````
$GLOBALS['TCA']['tt_content']['columns']['bodytext']['displayCond'] = 'USER:Vendor\Example\User\ElementConditionMatcher->checkHeaderGiven:any:more:information';
```
In-Depth Changes

API for Twitter Bootstrap modals (1)

- Two new API methods to create/remove modal popups:
  - TYPO3.Modal.confirm(title, content, severity, buttons)
  - TYPO3.Modal.dismiss()

- Options title and content are required

- Options buttons.text and buttons.trigger are also required, if buttons is used

- Example 1:

  TYPO3.Modal.confirm(
    'The title of the modal', // title
    'This the the body of the modal', // content
    TYPO3.Severity.warning    // severity
  );
API for Twitter Bootstrap modals (2)

Example 2:

```javascript
TYPO3.Modal.confirm('Warning', 'You may break the internet!',
    TYPO3.Severity.warning,
    [
        {
            text: 'Break it',
            active: true,
            trigger: function() { ... }
        },
        {
            text: 'Abort!,'
            trigger: function() {
                TYPO3.Modal.dismiss();
            }
        }
    ]
);
In-Depth Changes

JavaScript Storage API (1)

- Accessing the BE user configuration ($BE_USER->uc) can be handled in JavaScript by using simple key-value pairs.
- Additionally, HTML5’s localStorage can be used to store data in the user’s browser (client-side).
- Two new global TYPO3 objects:
  - top.TYPO3.Storage.Client
  - top.TYPO3.Storage.Persistent
- Each object has the following API methods:
  - get(key): fetch data
  - set(key, value): write data
  - isset(key): check, if key is already used
  - clear(): empty all storage data
In-Depth Changes

JavaScript Storage API (2)

Example:

```javascript
// get value of key 'startModule'
var value = top.TYPO3.Storage.Persistent.get('startModule');

// write value 'web_info' as key 'start_module'
top.TYPO3.Storage.Persistent.set('startModule', 'web_info');
```
Inline Rendering of Checkboxes

- Checkbox setting `inline` for "cols" can be used to render checkboxes directly next to each other to reduce the amount of space used.

```php
'weekdays' => array(
    'label' => 'Weekdays',
    'config' => array(
        'type' => 'check',
        'items' => array(
            array('Mo', ''),
            array('Tu', ''),
            array('We', ''),
            array('Th', ''),
            array('Fr', ''),
            array('Sa', ''),
            array('Su', '')
        ),
        'cols' => 'inline'
    ),
),
...
In-Depth Changes

Content Object Registration

- New global option to register and/or extend/overwrite cObjects such as TEXT has been introduced
- A list of all available cObjects is available as:
  \$GLOBALS['TYPO3_CONF_VARS']['FE']['ContentObjects']

**Example: register a new cObject EXAMPLE**

\$GLOBALS['TYPO3_CONF_VARS']['FE']['ContentObjects']['EXAMPLE'] = Vendor\MyExtension\ContentObject\ExampleContentObject::class;

- The registered class must be a subclass of TYP03\CMS\Frontend\ContentObject\AbstractContentObject
- Store your class in directory typo3conf/myextension/Classes/ContentObject/ to be prepared for future autoload mechanisms
In-Depth Changes

Hooks and Signals (1)

- New hook has been added to the end of `PageRepository->init()`, which allows to influence the visibility of pages.

- Register the hook as follows:

  ```
  $GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS'][
  \\TYPO3\CMS\Frontend\Page\PageRepository::class]['init']
  ```

- The hook class must implement the following interface:

  ```
  \TYPO3\CMS\Frontend\Page\PageRepositoryInitHookInterface
  ```
In-Depth Changes

Hooks and Signals (2)

- New hook has been added to the PageLayoutView to manipulate the rendering of the footer of a content element.

Example:

```php
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS'][
    'cms/layout/class.tx_cms_layout.php'][
    'tt_content_drawFooter'];
```

- The hook class must implement the following interface:

```php
\TYPO3\CMS\Backend\View\PageLayoutView\PageLayoutViewDrawFooterHookInterface
```
New hook has been added as a post processor of BackendUtility::countVersionsOfRecordsOnPage

This can be used to visualize workspace states in the page tree for example

Register the hook as follows:

```
$GLOBALS[‘TYPO3_CONF_VARS’][‘SC_OPTIONS’]['t3lib/class.t3lib_befunc.php']['countVersionsOfRecordsOnPage'][] = 'My\Package\HookClass->hookMethod';
```
In-Depth Changes

Hooks and Signals (4)

- New signal has been added to the end of method `DataPreprocessor::fetchRecord()`

- This can be used to manipulate array `regTableItems_data` for example, in order to display manipulated data in TCEForms

```php
$this->getSignalSlotDispatcher()->dispatch(
    \TYPO3\CMS\Backend\Form\DataPreprocessor::class,
    'fetchRecordPostProcessing',
    array($this)
);
```
New signal has been added, that allows for additional processing upon initialization of a mailer object, e.g. registering a Swift Mailer plugin

```php
$signalSlotDispatcher = \TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance(
    \TYPO3\CMS\Extbase\SignalSlot\Dispatcher::class
);

$signalSlotDispatcher->connect(
    \TYPO3\CMS\Core\Mail\Mailer::class,
    'postInitializeMailer',
    \Vendor\Package\Slots\MailerSlot::class,
    'registerPlugin'
);
```
In-Depth Changes

Multiple UID in PageRepository::getMenu()

- Method PageRepository::getMenu() accepts arrays now, in order to define multiple root pages

```php
$pageRepository = new \TYPO3\CMS\Frontend\Page\PageRepository();
$pageRepository->init(FALSE);
$rows = $pageRepository->getMenu(array(2, 3));
```
Chapter 4:

Extbase & Fluid
Extbase & Fluid

PaginateViewHelper

- Since TYPO3 CMS 7.1, PaginateViewHelper accepts input collections of the following types:
  - QueryResultInterface
  - ObjectStorage
  - ArrayAccess
  - array

- Example:

  ```
  <f:widget.paginate objects="{blogs}" as="paginatedBlogs">
    <f:for each="{paginatedBlogs}" as="blog">
      <h4>{blog.title}</h4>
    </f:for>
  </f:widget.paginate>
  ```
Extbase & Fluid

ContainerViewHelper loads RequireJS modules

- ContainerViewHelper can load RequireJS modules via the includeRequireJsModules attribute

Example:

```xml
<f:be.container pageTitle="Extension Module" loadJQuery="true"
includeRequireJsModules="{
  0:'TYPO3/CMS/Extension/Module1',
  1:'TYPO3/CMS/Extension/Module2',
  2:'TYPO3/CMS/Extension/Module3',
  3:'TYPO3/CMS/Extension/Module4'
}"/>
```
Method `has()` in ObjectAccess

- For the usage in Fluid, `object.property` and `object.isProperty` already support the following methods:
  - `isProperty()`
  - `getProperty()`
- New since TYPO3 CMS 7.1: `hasProperty()`
- This calls method `$object->hasProperty()` if `object.hasProperty` is used in Fluid
Extbase & Fluid

Upload multiple files with FormUpload-ViewHelper

FormUpload-Viewhelper supports new attribute `multiple`, that provides the option to upload multiple files at a time

```xml
<f:form.upload property="files" multiple="multiple" />
```

Note: developers need to prepare the incoming value for the property mapping by writing their own TypeConverter!
Chapter 5:
Deprecated/Removed Functions
Deprecated/Removed Functions

$TYPO3_CONF_VARS[SYS][compat_version]

- Option $TYPO3_CONF_VARS[SYS][compat_version] (which was modified on update in the Install Tool wizard) has been removed
- All checks on GeneralUtility::compat_version are now made against constant TYPO3_branch

Note: TypoScript conditions, which check for older compat_version have a different behaviour now!
Deprecated/Removed Functions

Inline styles of `<blockquote>` tag

- CSS Styled Content renders `<blockquote>` tags using `lib.parseFunc_RTE TypoScript`.
- These lines have been removed without substitution:

  ```
  lib.parseFunc_RTE.externalBlocks.blockquote.callRecursive.tagStdWrap.HTMLparser = 1
  lib.parseFunc_RTE.externalBlocks.blockquote.callRecursive.tagStdWrap.HTMLparser.tags.blockquote.
      overrideAttribs = style="margin-bottom:0;margin-top:0;"
  ```

- As a result, inline styles "margin-bottom:0;margin-top:0;" are removed.

  **Note:** styling of `<blockquote>` tags possibly differ after an upgrade to TYPO3 CMS 7.1.
Deprecation/Removed Functions

Workspaces: field disable_autocreate

- Deprecated field disable_autocreate has been removed from EXT:workspaces
- If a TYPO3 extension relies on this field, a SQL error occurs
Deprecated/Removed Functions

**Functionality `include_once`**

- Functionality to include PHP files within module functions (e.g. Info module) via an `include_once` array has been removed.
- This applies to the following modules:
  - Web => Page
  - Web => Page - New Content Element Wizard
  - Web => Functions
  - Web => Info
  - Web => Template
  - Web => Recycler
  - User => Task Center
  - System => Scheduler
Setting `config.meaningfulTempFilePrefix`

- In TYPO3 CMS < 7.1, file names of images generated by the GIFBUILDER could be influenced by TypoScript option: `config.meaningfulTempFilePrefix` (GIFBUILDER used a hash value as the file name only).
- This option has been removed (names of files in directory `typo3temp/GB/` show the original file name as the first element automatically).
Deprecated/Removed Functions

Removed files

The following files have been removed:

- typo3/file_edit.php
- typo3/file_newfolder.php
- typo3/file_rename.php
- typo3/file_upload.php
- typo3/show_rechis.php
- typo3/listframe_loader.php

Their functionalities have been migrated to backend modules, e.g.

typo3/file_edit.php in BackendUtility::getModuleUrl('file_edit');
Deprecated/Removed Functions

ExtJS DateTimePicker

- ExtJS component Ext.ux.DateTimePicker has been removed and replaced with a Twitter Bootstrap alternative (see chapter "Backend User Interface")

- Affected TYPO3 CMS system extensions are EXT:belog or EXT:scheduler for example

  Note: extensions which rely on the deprecated function Ext.ux.DateTimePicker will likely break!
Deprecated/Removed Functions

Access List Render Mode

- The following variable has been removed:
  $GLOBALS[TYP03_CONF_VARS][BE][accessListRenderMode]

- Corresponding fields in TCA tables be_users and be_groups are set to the default value "checkbox"

- This can be changed in file typo3conf/extTables.php:

```php
$GLOBALS['TCA']['be_users']['columns']['file_permissions']['config']['renderMode'] = 'singlebox';
$GLOBALS['TCA']['be_users']['columns']['userMods']['config']['renderMode'] = 'singlebox';

$GLOBALS['TCA']['be_groups']['columns']['file_permissions']['config']['renderMode'] = 'singlebox';
$GLOBALS['TCA']['be_groups']['columns']['pagetypes_select']['config']['renderMode'] = 'singlebox';
$GLOBALS['TCA']['be_groups']['columns']['tables_select']['config']['renderMode'] = 'singlebox';
$GLOBALS['TCA']['be_groups']['columns']['tables_modify']['config']['renderMode'] = 'singlebox';
$GLOBALS['TCA']['be_groups']['columns']['non_exclude_fields']['config']['renderMode'] = 'singlebox';
$GLOBALS['TCA']['be_groups']['columns']['userMods']['config']['renderMode'] = 'singlebox';
```
Deprecated/Removed Functions

Content element "Mailform"

- Mailform functionality, which provided the cObject FORM, has been removed from the core
- Still available in legacy extension EXT:compatibility6 if required
- The following options have been marked as deprecated:
  
  $TYPO3_CONF_VARS[FE][secureFormmail]
  $TYPO3_CONF_VARS[FE][strictFormmail]
  $TYPO3_CONF_VARS[FE][formmailMaxAttachmentSize]

- The following methods within TypoScriptFrontendController have been removed:
  
  protected checkDataSubmission()
  protected sendFormmail()
  public extractRecipientCopy()
  public codeString()
  protected roundTripCryptString()
Deprecated/Removed Functions

Functionality changed (1)

- EXT:indexed_search is activated as soon as the extension is installed. As a result, appropriate TypoScript options `config.index_enable = 1` and `config.index_externals = 1` become active automatically, too.

- TSconfig `web_func.menu.wiz` changed to `web_func.menu.functions`

- Extensions, which integrate into the upper right toolbar, must implement the new interface: `TYPO3\CMS\Backend\Toolbar\ToolbarItemInterface` and must be registered in: `$GLOBALS['TYPO3_CONF_VARS']['BE']['toolbarItems']`
Deprecated/Removed Functions

Functionality changed (2)

- File `typo3/js/tree.js` has been replaced by `EXT:backend/Resources/Public/JavaScript/LegacyTree.js` (latter is based on jQuery)
- Variable `$GLOBALS['WEBMOUNTS']` has been replaced by `$GLOBALS['BE_USER']->returnWebmounts()`
- Support of `.t3-table` and `.t3-button` has been removed (Twitter Bootstrap classes implement the visual appearance now)
- Country flags (PNG images) have been moved from `typo3/gfx/flags/` and `typo3/sysext/t3skin/images/flags/` to: `typo3/sysext/core/Resources/Public/Icons/flags/`
Deprecated/Removed Functions

Functionality changed (3)

- CSS Styled Content TypoScript templates for TYPO3 CMS 4.5 to 6.1 have been removed.
- The following TypoScript cObjects have been moved to legacy extension `EXT:compatibility6`:
  
  - `SEARCHRESULTS`
  - `COLUMNS`
  - `OTABLE`
  - `CLEARGIF`
  - `IMGTEXT`
  - `CTABLE`
  - `HRULER`
- Content element `search` has been moved to legacy extension `EXT:compatibility6`.
- The following TCA wizard options have been removed:
  
  - `_PADDING`
  - `_VALIGN`
  - `_DISTANCE`
Deprecated/Removed Functions

TypoScript option `andWhere`

- TypoScript option `andWhere` has been marked as deprecated
- Integrators should use properties `where` and `markers` instead:
The following entry points have been classified as deprecated:

- typo3/tce_file.php
- typo3/move_el.php
- typo3/tce_db.php
- typo3/login_frameset.php
- typo3/sysext/cms/layout/db_new_content_el.php
- typo3/sysext/cms/layout/db_layout.php

Instead, use the following:

```php
\TYPO3\CMS\Backend\Utility\BackendUtility::getModuleUrl(''<parameter>''
```

Where `<parameter>` could be:

- tce_file, move_element, tce_db, login_frameset,
- new_content_element, web_layout
Deprecated/Removed Functions

Miscellaneous (1)

- TypoScript option `config.xhtmlDoctype = xhtml_2` has been marked for removal in TYPO3 CMS 8

- The following methods have been marked as deprecated:
  - `TYPO3\CMS\Backend\Utility\BackendUtility::implodeTSParams()`
  - `TYPO3\CMS\Backend\Controller::makeLoginBoxImage()`

- The following method has been marked as deprecated:
  - `LocalImageProcessor::getTemporaryImageWithText()`

  ...and is replaced by:
  - `TYPO3\CMS\Core\Imaging\GraphicalFunctions::getTemporaryImageWithText()`

- StdWrap properties `textStyle` and `tableStyle` have been marked as deprecated
Deprecated/Removed Functions

Miscellaneous (2)

- TypoScript option `page.includeJSlibs` has been renamed to `page.includeJSLibs` (uppercase "L") and old option marked as deprecated
- TypoScript condition `device` has been marked as deprecated
- Method `DocumentTable::table()` has been marked as deprecated (developers should use Fluid for this)
- The following method has been marked as deprecated:
  ```pseudo
  TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController::doXHTML_cleaning()
  ```
  ...as well as the TypoScript option `config.xhtml_cleaning`
- The following hook has been marked as deprecated:
  ```pseudo
  $GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['GLOBAL']['softRefParser_GL']```
 Deprecated/Removed Functions

Miscellaneous (3)

- The following methods have been marked as deprecated:
  
  `TypoScriptTemplateObjectBrowserModuleFunctionController::verify_TSobjects()`  
  `ExtendedTemplateService::ext_getKeyImage()`  
  `ConfigurationForm::ext_getKeyImage()`  

- Executing `contentObject->COBJECT()` has been marked as deprecated  
  (use `$cObj->cObjGetSingle('...', $conf);` instead)

- Direct access to `FormEngine::$renderReadonly` has been marked as deprecated  
  (use `AbstractFormElement::setRenderReadonly(TRUE);` instead)
Deprecated/Removed Functions

Miscellaneous (4)

- The following FormEngine methods are marked as deprecated:
  - FormEngine::insertDefStyle
  - FormEngine::getAvailableLanguages()
  - FormEngine::sL()
  - FormEngine::renderVDEFDiff()
  - FormEngine::getLL()
  - FormEngine::getTSCpid()
  - FormEngine::getSingleField_typeFlex_langMenu()
  - FormEngine::getSingleField_typeFlex_sheetMenu()
  - FormEngine::getSpecConfFromString()
Chapter 6:
Sources and Authors
Sources and Authors

Sources (1)

**TYPO3 News:**
- [http://typo3.org/news](http://typo3.org/news)

**Release Infos:**
- [http://wiki.typo3.org/TYPO3_CMS_7.1.0](http://wiki.typo3.org/TYPO3_CMS_7.1.0)
- `INSTALL.md` and `Changelog`
- `typo3/sysext/core/Documentation/Changelog/7.1/*`

**TYPO3 Bug-/Issuetracker:**

**TYPO3 Git Repositories:**
- [https://git.typo3.org/Packages/TYPO3.CMS.git](https://git.typo3.org/Packages/TYPO3.CMS.git)
- [https://git.typo3.org/Packages/TYPO3.Fluid.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, Paul Blondiaux, Sergio Catala, Ben van’t Ende,
Michel Mix, Sinisa Mitrovic, Nena Jelena Radovic and Roberto Torresani

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

Licensed under Creative Commons BY-NC-SA 3.0