

CMS Certified Integrator 14 LTS

Created on: 08.10.2026

TYPO3 Basics

About TYPO3

TYPO3 is an open-source enterprise content management system (CMS). The system offers a wide range of functions out of the box and can be extended by extensions. The software license (GPL) lets everyone access, read, copy, distribute, and modify the source code which is freely available. TYPO3 is well documented at <https://docs.typo3.org>.

Maintainer: TYPO3 (Torben Freudewald)

TYPO3 Versions

TYPO3 uses Semantic Versioning (semver.org) for the version numbers. Each major version usually gets 3 years of free bug fixes and security updates. After that, support can be extended with ELTS versions for a fee via TYPO3 GmbH.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Project

The development of the TYPO3 project began in 1997. Since then, in addition to the TYPO3 CMS, other products such as Surf, Fluid, Neos have been developed as open source software by the TYPO3 community and later by TYPO3 GmbH, which was founded by the TYPO3 Association. The TYPO3 Association is the owner of the TYPO3 brand and provides the TYPO3 style guide.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Community

The TYPO3 community consists of everyone who uses TYPO3 as an editor or develops it further as a developer, integrator or agency. The community organizes events like Usergroup meetings and TYPO3Camps. Everyone can participate in different teams to bring the TYPO3 project forward.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Association

The TYPO3 Association builds the groundwork for the TYPO3 project and empowers the TYPO3 community with infrastructure and guidance.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Handling

Page Tree

The structure of the web pages managed in TYPO3 is easily accessible via the page tree. It is possible to manage pages that are visible in the frontend as well as associated data, which are then installed at the appropriate place in the frontend. With a click on the page icon the context menu appears with additional options.

Maintainer: TYPO3 (Torben Freudewald)

Admin Panel Basics

Once enabled, the admin panel can be accessed by a click of a button in the frontend. This opens a bar at the bottom the screen that shows a summary of information about the currently accessed page, including several TypoScript messages, page load time, and possibly errors and warnings. The main purpose of the admin panel is general system debugging but it contains toggles and switches to control the frontend output as well.

Maintainer: TYPO3 (Torben Freudewald)

TYPO3 CLI (command line interface) commands

TYPO3 provides a command line interface (CLI) for running scripts directly from the terminal. CLI commands can be run locally, on remote servers via SSH, inside Docker containers, and within CI/CD pipelines.

Maintainer: TYPO3 (Oliver Thiele)

Admin Panel

The "TYPO3 Admin Panel" provides a deeper insight into the internal processes of TYPO3 at run-time. Once it has been activated, backend users can access performance and cache statistics, settings of a specific page, etc. while accessing the frontend of the website.

Maintainer: TYPO3 (Oliver Thiele)

Dashboard Basics

The backend module "Dashboard" allows backend users to create, select, and configure dashboards and widgets. These widgets typically provide details and status information about the TYPO3 instance.

Maintainer: TYPO3 (Torben Freudewald)

Reference Index

The reference index in TYPO3 ensures data consistency by managing and updating references between records. It can be accessed through the TYPO3 backend to check and rebuild references, helping to resolve missing and broken references.

Maintainer: TYPO3 (Oliver Thiele)

Fluid CLI Tools

TYPO3 ships four CLI commands under the ``fluid:`` namespace that help integrators analyse template quality, manage the Fluid cache, inspect ViewHelper namespaces, and generate XSD schema files for IDE support. These commands are particularly useful during development, after deployments, and when debugging Fluid-related issues in a project.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript

TypoScript Conditions

Using TypoScript conditions to influence frontend rendering based on various factors like the application context, page properties or user groups.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript TLOs and Reserved Names

Top-Level Object (TLO) overview, commonly used configuration directives in regard to TLOs.

Maintainer: TYPO3 (Oliver Thiele)

Page TSconfig

Page TSconfig configures settings and backend behavior at the page level. It allows field visibility, backend modules, and default values to be customized for specific pages and page trees.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript cObject FLUIDTEMPLATE

Understanding how to configure TypoScript FLUIDTEMPLATE content objects so that Fluid templates can be rendered in the frontend. This includes setting template files, template paths, and passing variables and settings from TypoScript into the Fluid view.

Maintainer: TYPO3 (Oliver Thiele)

Knowing TYPO3 Data Types

TypoScript includes simple data types such as string, integer and boolean, as well as TYPO3-specific data types such as imgResource.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript top level object Config

This skill focuses on using the TypoScript config object to control the rendering and behavior of the TYPO3 frontend.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript Syntax & Basics

Basic knowledge of TypoScript and its syntax

Maintainer: TYPO3 (Oliver Thiele)

PageTitleProvider

As part of search engine optimization (SEO), the HTML title of pages can be customized by configuring PageTitle Providers in TYPO3. By setting provider priority in TypoScript, you can control which title source takes precedence.

Maintainer: TYPO3 (Oliver Thiele)

Active TypoScript / Included TypoScript

About Active and Included TypoScript.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript cObject PAGEVIEW

The content object PAGEVIEW can be used instead of FLUIDTEMPLATE to configure the HTML body of a page using Fluid templates. This content object has very specific conventions and defaults, that require (and allow) less configuration compared to using FLUIDTEMPLATE.

Maintainer: TYPO3 (Oliver Thiele)

Site Sets

Site sets ship parts of the site configuration as modular pieces. They are intended to provide settings, TypoScript and page TSconfig for the scope of a site.

Maintainer: TYPO3 (Oliver Thiele)

Accessing Data Sources

Using available data sources in TypoScript.

Maintainer: TYPO3 (Oliver Thiele)

TypoScript function parseFunc

ParseFunc is a TypoScript function used to parse and transform text content, such as rendering links, modifying HTML tags (adding/removing CSS classes), filtering unwanted HTML tags, etc. It is frequently applied in content processing for frontend output.

Maintainer: TYPO3 (Oliver Thiele)

Server Management

Server-Side Image Manipulation

Overview of server-side image manipulation libraries used in TYPO3.

Maintainer: TYPO3 (Oliver Thiele)

Knowing the Cron Daemon

Cron is a Unix/Linux daemon for time-based task scheduling. In TYPO3, it is used to trigger the Scheduler regularly, for example for cache warm-up, indexing, or sending emails. Integrators need to be able to read, write, and configure crontab entries on the server.

Maintainer: TYPO3 (Oliver Thiele)

YAML Basics

Understanding YAML configuration files.

Maintainer: TYPO3 (Oliver Thiele)

File Permissions

Understanding UNIX system file permissions.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 File Architecture

Every TYPO3 extension follows a defined directory and file structure that determines where PHP classes, configuration, templates, and public assets are placed. Alongside this, correctly set file and directory permissions at the operating system level are essential for both a functional and secure installation. Integrators need to understand both aspects to structure extensions correctly and harden their TYPO3 deployments.

Maintainer: TYPO3 (Oliver Thiele)

Command Line Basics

Performing basic tasks via the command line.

Maintainer: TYPO3 (Oliver Thiele)

Regular Expression Basics

Identifying and matching patterns in strings with the help of regular expressions (PCRE).

Maintainer: TYPO3 (Oliver Thiele)

IP-Anonymization for Data Records

Using the scheduler to anonymize IP-Adresses that are in persistent storage to increase data protection.

Maintainer: TYPO3 (Oliver Thiele)

HTTP Status Codes

HTTP status codes are part of every web project an integrator handles. In TYPO3, status codes appear in the redirect module, error page configuration, caching behaviour, and when diagnosing broken URLs or misconfigured pages. Knowing the most relevant codes and when TYPO3 generates them is essential for reliable site configuration.

Maintainer: TYPO3 (Oliver Thiele)

Backend Administration

User TSconfig

User TSconfig is a TypoScript-syntax configuration system for the TYPO3 backend, applied to backend users and groups. It controls individual backend behaviour per editor or group, and can override page TSconfig properties.

Maintainer: TYPO3 (Oliver Thiele)

Scheduler

The Scheduler supports one-time or periodic execution of tasks. Internal TYPO3 tasks can be extended by extensions.

Maintainer: TYPO3 (Oliver Thiele)

Using the Drag & Drop Editor for Basic Forms

EXT:form provides a visual form editor for TYPO3 integrators and editors. Forms are built from form elements including basic inputs, special fields, select elements, file uploads, and container structures.

Maintainer: TYPO3 (Oliver Thiele)

Form Localization

EXT:form translates three types of content using XLF files including form element properties, validation messages, and finisher options. There is a consistent fallback chain from most specific to most generic. Translation arguments can inject dynamic values into translated strings.

Maintainer: TYPO3 (Oliver Thiele)

Form Templating

EXT:form uses Fluid for all frontend output - both for rendering form elements in the browser and for generating email content via the EmailFinisher. Integrators can replace the default Bootstrap-based templates by registering custom paths via YAML. The EmailFinisher requires separate HTML and plaintext templates and includes form values and custom variables inside those templates.

Maintainer: TYPO3 (Oliver Thiele)

Import/Export

The system extension “impexp” allows page trees and records to be exported from one TYPO3 instance and then imported into another. Export files are created in XML or custom T3D format and

can include related records, files, and relations. The extension is restricted to admin users by default and has security implications.

Maintainer: TYPO3 (Oliver Thiele)

Root Node

Properties and special features of the root node in TYPO3.

Maintainer: TYPO3 (Oliver Thiele)

Database Mounts

Administrators can configure DB mounts for user groups to control access to page tree sections. This enables user groups to be assigned access to particular areas of the site structure, ensuring more granular permission management.

Maintainer: TYPO3 (Oliver Thiele)

Managing Backend Pages Access

TYPO3's backend access control system combines four mechanisms to define what a backend user can see and do: access lists, DB and file mounts, page permissions and User TSconfig. All mechanisms are defined at group level and merged additively across multiple groups.

Maintainer: TYPO3 (Oliver Thiele)

RTE Configuration

Configuring the Rich Text Editor in TYPO3 involves three independent layers: editor configuration (what the editor shows and allows), RTE transformations (how content is cleaned when saved to or loaded from the database), and frontend output configuration (how stored HTML is rendered in the frontend). x1

Maintainer: TYPO3 (Oliver Thiele)

Accessing the Install Tool/System Maintainer Area

Knowledge about security measures taken to protect the System Maintainer Area (Standalone: Install Tool).

Maintainer: TYPO3 (Oliver Thiele)

Workspace Concepts

TYPO3 Workspaces adds versioning and a structured review process to the TYPO3 backend. It provides a safe editing environment where changes are isolated from the live website until explicitly published, with configurable stages, user roles, and notification workflows.

Maintainer: TYPO3 (Oliver Thiele)

Configuring Workspaces

The configuration and setup of custom editorial workflows.

Maintainer: TYPO3 (Oliver Thiele)

Reports Module

The Reports module contains information and diagnostic data about your TYPO3 CMS installation. This module can list configuration errors, security issues, etc.

Maintainer: TYPO3 (Oliver Thiele)

Managing Backend Users and Backend User Groups

Backend users and groups are stored at the root of the page tree, editable only by admin users. Permissions are primarily defined on groups rather than directly on user records. A user can belong to multiple groups including sub-groups, and all permissions are merged additively. Users and groups can be managed via the Administration > Users module.

Maintainer: TYPO3 (Oliver Thiele)

URL Routing in TYPO3

Handling routing in TYPO3 and understanding the basics of human readable (user friendly) URLs.

Maintainer: TYPO3 (Oliver Thiele)

Configure Installation-wide Options in the System Maintainer Area / Install Tool

The Install Tool is a system-level configuration area, accessible only to administrators with the correct password. It provides access to installation-wide settings that cannot normally be configured in the backend. This includes image processing, session handling, email setup, connection encryption, and Content Security Policy.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 URL Redirects (Configuration)

Redirecting URLs in TYPO3, related options and their implications.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Status Module

The Status backend module displays a variety of information related to the pages in your web site.

Maintainer: TYPO3 (Oliver Thiele)

Recycler backend module

The Recycler backend module provides the option to clear the garbage collection and to restore data.

Maintainer: TYPO3 (Oliver Thiele)

Linkvalidator

Detect broken internal and external links using the TYPO3 core extension Linkvalidator.

Maintainer: TYPO3 (Oliver Thiele)

Dashboard Configuration

Knowledge of dashboard configuration options in the TYPO3 backend.

Maintainer: TYPO3 (Oliver Thiele)

Application Context

The application context can be used to switch between configurations for different environments (e.g. development, staging and live system).

Maintainer: TYPO3 (Oliver Thiele)

Internal Notes

The "sys_note" system extension allows backend users to create internal notes.

Maintainer: TYPO3 (Oliver Thiele)

Using Reactions

The system extension "Reactions" handles incoming webhooks to TYPO3. It also provides a corresponding backend module to manage reaction records.

Maintainer: TYPO3 (Oliver Thiele)

Using Webhooks

Webhooks are a useful way to integrate real-time information between different applications or services. They allow a service to "call" or notify another application or service when a specific event occurs. This is done by sending an HTTP request to a specific URL provided by the receiving application.

Maintainer: TYPO3 (Oliver Thiele)

Language Configuration

This skill covers setting up TYPO3 to support multiple languages for frontend content delivery.

Maintainer: TYPO3 (Oliver Thiele)

Configuring Backend Entry Points

The default entry point for the TYPO3 backend is /typo3, but it can be set to any other path or even to a different domain or subdomain.

Maintainer: TYPO3 (Oliver Thiele)

Site Configuration (YAML)

Advanced setup of the site configuration in the file config/sites/[identifier]/config.yaml.

Maintainer: TYPO3 (Oliver Thiele)

Site Configuration (Backend Module)

Setting up the domain and request related settings for a website.

Maintainer: TYPO3 (Oliver Thiele)

Reusing Basic Forms

Transferring basic forms from one installation to another.

Maintainer: TYPO3 (Oliver Thiele)

Form Definition Structure

A form definition is a YAML file that describes a specific form including its pages, form elements, validators, and finishers. It is distinct from the form configuration, which defines what is available to the system. A form definition specifies what a particular form looks like and how it behaves. Form definitions are typically created via the backend form editor.

Maintainer: TYPO3 (Oliver Thiele)

Form Configuration

Form configuration covers the general behaviour of the EXT:form extension - which form elements, validators, and finishers are available, how they are pre-configured, and how they appear in the frontend/backend. Form configuration is defined in YAML and organised in *form sets*. Configuration is defined separately from specific form definitions.

Maintainer: TYPO3 (Oliver Thiele)

Form Concepts

EXT:form is TYPO3's built-in form framework. It covers: creating a form in the backend, embedding it on a page, processing submitted data, and defining what happens after submission. As an integrator, the key task is to configure the framework to control what editors can see and do, and to define how forms behave in the frontend.

Maintainer: TYPO3 (Oliver Thiele)

Media Management

Charsets

Definition of charsets, common usage, differences (advantages/disadvantages) of specific charsets.

Maintainer: TYPO3 (Oliver Thiele)

CSV - Character Separated Values

CSV files are used to exchange data between different computer systems. The editor must know how these files are structured so that they can be exported and additionally how they can be processed by TYPO3 without errors.

Maintainer: TYPO3 (Torben Freudewald)

Copyright and data protection

By using external files, for example images or videos, the copyright must be considered. When handling with information relating to an identifiable person the (country specific) legal situation must be considered.

Maintainer: TYPO3 (Torben Freudewald)

Software Development Concepts

What is MVC?

The Model–View–Controller (MVC) pattern separates application logic into three components: Model, View, and Controller, dividing data handling, presentation, and user interaction.

Maintainer: TYPO3 (Oliver Thiele)

What is a Templating Engine?

Introduction to the basic concept of a templating engine/processor.

Maintainer: TYPO3 (Oliver Thiele)

Templating and Other Outputs

What is Fluid?

Definition of Fluid and its use in the context of the Model View Controller pattern.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelpers

ViewHelpers transform data and dynamically output it into templates. They adapt raw data to meet specific presentation requirements.

Maintainer: TYPO3 (Oliver Thiele)

Templates, Layouts, Partials

Fluid uses three types of template files: Templates, Layouts, and Partials. Templates are always the starting point. Layouts wrap reusable HTML markup around multiple templates, such as header and footer. Partials are reusable code snippets that require variables to be passed in explicitly. Together they separate structure, shared markup, and reusable fragments into maintainable units.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Comment

Including comments in Fluid templates.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: If/Else

Creating if/else statements within Fluid templates.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: For-Loop

Iterating through a list with the for ViewHelper.

Maintainer: TYPO3 (Oliver Thiele)

Importing ViewHelpers

Using third-party ViewHelpers in Fluid templates.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Render

Using `<f:render>` to render sections and partials as well as passing arguments.

Maintainer: TYPO3 (Oliver Thiele)

Using the Fluid View

Knowing and understanding the different parts of the Fluid template rendering engine.

Maintainer: TYPO3 (Oliver Thiele)

HTML Basics

HTML (HyperText Markup Language) is the backbone of web development, defining the structure and content of web pages. In TYPO3, HTML plays a crucial role in templates and frontend customization.

Maintainer: TYPO3 (Oliver Thiele)

JavaScript Basics

JavaScript is a versatile programming language that enhances interactivity and functionality of websites. Understanding how it integrates into TYPO3 allows developers to create dynamic and user-friendly frontends.

Maintainer: TYPO3 (Oliver Thiele)

CSS Basics

Cascading Style Sheets (CSS) are a core technology used for describing and controlling the presentation of web pages. TYPO3 integrators need foundational knowledge of CSS to effectively style content elements and templates.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Link, Uri

Using the Fluid ViewHelpers `<f:link.*>` and `<f:uri.*>`.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Format

The `f:format` group contains multiple ViewHelpers for transforming and formatting output in Fluid templates. They cover string manipulation, number and date formatting, HTML processing, and output escaping. Some of them have direct security implications and require careful selection.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Debug

Learn how to use the Debug ViewHelper in TYPO3 to output debugging information in Fluid templates.

Maintainer: TYPO3 (Oliver Thiele)

Data Processors

Data processors transform and enrich raw data before it is passed to Fluid templates. They convert database values and relations into structured objects and arrays that can be used directly in templates. For example, a database field such as `tt_content.assets` only stores file relation information internally, while a data processor resolves it into `FileReference` objects that Fluid can work with directly.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Asset.*

TYPO3 provides dedicated Fluid ViewHelpers for registering frontend assets such as CSS and JavaScript files directly within Fluid templates. Integrators should understand how assets are registered, identified, and rendered consistently across extensions and templates.

Maintainer: TYPO3 (Oliver Thiele)

Managing Resource URIs with Cache Busting

The `f:uri:resources` ViewHelper creates URIs to resources. It can be used for extension resource files.

This ViewHelper automatically appends a timestamp for file versioning. This ensures that browsers fetch updated files when necessary, even if the file name has not changed.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: `render.contentArea`

The `<f:render.contentArea>` ViewHelper, introduced in TYPO3 v14, provides a standardised way to render all content elements within a backend layout column in a Fluid page template. It replaces the previous combination of `<f:cObject>` and `<f:for>` and works with content areas supplied by the page-content data processor.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: translate

The `f:translate` ViewHelper outputs translated strings from XLIFF language files stored in `Resources/Private/Language/`. Labels can be referenced via a short key, an extension name, a translation domain, or a full `LLL:EXT: path`. Dynamic values can be injected into labels using placeholders, and a fallback string can be defined for missing keys.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Page.meta

The `<f:page.meta>` ViewHelper, introduced in TYPO3 14.0, allows integrators to set HTML meta tags directly within Fluid templates. It integrates with TYPO3's MetaTagManager API, which means meta tags defined in templates respect the same priorities and rendering rules as those set via TypoScript or PHP. This removes the need for custom PHP logic or TypoScript configuration in many common scenarios.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Page.headerData, Page.footerData

`f:page.headerData` allows code to be injected into the HTML `<head>` of a rendered page. Its counterpart, `f:page.footerData` injects code right before the closing `</body>` tag. Both ViewHelpers were introduced in TYPO3 14, use the PageRenderer API, and output their content as-is, which means any user-supplied input must be escaped manually to prevent XSS vulnerabilities.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Image and Image.srcset

The `f:image` ViewHelper resizes, crops, or converts an image if required and renders the resulting HTML `<img>` tag. It supports two primary input methods: the `src` argument for extension assets using `EXT: syntax`, and the `image` argument for FAL (File Abstraction Layer) objects such as `File` or `FileReference`. External URLs are not processed. Image conversion format defaults can be configured globally via `$GLOBALS['TYPO3_CONF_VARS']['GFX']['imageFileConversionFormats']`.

The `f:image.srcset` ViewHelper, introduced in TYPO3 14.2, generates a list of image URLs with their corresponding `srcset` descriptors for use in the `srcset` attribute of an `<img>` or `<source>` tag. It replaces previous manual approaches using multiple `f:uri.image` calls and generates the required image variants automatically.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Format.date

The `f:format.date` ViewHelper formats any `DateTimeInterface` object or date string into a human-readable output within Fluid templates. It supports PHP date format strings, ICU-based (International Components for Unicode) pattern formatting for locale-aware output, timezone control, and relative date expressions via `strtotime` syntax. Integrators use it whenever dates need to be rendered in a specific format on the frontend.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Asset.styleAttr

The Fluid ViewHelper `f:asset.styleAttr`` is used to generate inline CSS declarations for HTML style

attributes directly in Fluid templates. Integrators should understand how style definitions are generated dynamically and how inline styling can be applied consistently within TYPO3 frontend rendering.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Asset.script

The Fluid ViewHelper `<f:asset.script>` is used to register JavaScript assets directly in Fluid templates. TYPO3 centrally manages registered scripts through the AssetCollector, allowing structured and reusable frontend asset integration without directly embedding script tags in templates.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Asset.css

The Fluid ViewHelper `<f:asset.css>` is used to register CSS assets directly within Fluid templates. TYPO3 collects and renders these assets centrally, helping integrators manage stylesheets consistently across templates, extensions, and frontend output.

Maintainer: TYPO3 (Oliver Thiele)

Outputting and escaping variables in Fluid

Fluid templates receive variables and outputs them using a curly-brace syntax. This skill covers variable access, reserved names, dot notation, automatic escaping, Cross-Site-Scripting (XSS) risks, and the CDATA-based approach introduced in Fluid 5 (TYPO3 v14) to handle syntax collisions with inline CSS and JavaScript.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: Arguments

Fluid templates can define explicit arguments using the `<f:argument>` ViewHelper. Arguments declare which parameters a template, partial or component expects, including their type, default value, and whether they are optional. In the context of Fluid Components, argument definitions are mandatory and form the public API of a component.

Maintainer: TYPO3 (Oliver Thiele)

Fluid Components

Fluid Components are self-contained UI building blocks written entirely in Fluid — no PHP ViewHelper classes are required. Unlike standard partials, they declare an explicit, typed API via `<f:argument>`, making their interface self-documenting and preventing the use of implicit template variables. Child content can be embedded using `<f:slot>`, meaning components can be assembled in a way that standard partials cannot. Components follow Atomic Design methodology - they are organized by complexity into Atoms, Molecules, and Organisms, promoting a structured and scalable template architecture.

Maintainer: TYPO3 (Oliver Thiele)

Data processor: language-menu

The LanguageMenuProcessor is a data processor used within FLUIDTEMPLATE or PAGEVIEW to generate a list of language menu items from the site configuration. It provides all the data an integrator needs to build a functional language switcher in Fluid, including link, availability status, and hreflang attribute for each language.

Maintainer: TYPO3 (Oliver Thiele)

Configuring and Using Crop Variants for Images

This skill covers crop “variants” (different crop settings) in TYPO3’s image manipulation system. It covers how to define aspect ratios, crop areas, focus areas and cover areas. It also covers how to customise crop variants for a content element using `overrideChildTca` and `columnsOverrides`.

Maintainer: TYPO3 (Oliver Thiele)

Backend Layouts

Backend layouts define the content areas shown in the TYPO3 Page module and control frontend rendering at the same time.

They are configured in Page TSconfig as part of a site package, which is the recommended approach; alternatively they can be created directly in the TYPO3 backend and stored as database records, though this is no longer considered best practice.

Each layout consists of a grid of rows and columns; each column represents a named content area with a unique `colPos` value. Using the page-content `DataProcessor` and the `identifier` property, backend layout columns map directly to Fluid template variables. Since TYPO3 14, columns can additionally restrict which content element types are allowed or disallowed within them.

Maintainer: TYPO3 (Oliver Thiele)

ViewHelper: `render.text`

The `<f:render.text>` ViewHelper offers a unified way to output text fields from records in Fluid templates. It reads the TCA configuration of the field and applies the correct formatting automatically, so integrators no longer need to choose between direct output, `<f:format.nl2br>` or `<f:format.html>` themselves.

Maintainer: TYPO3 (Oliver Thiele)

Core Architecture & APIs

File abstraction layer (FAL)

The FAL is an abstraction of physical file asset storage in TYPO3. It allows *Digital Assets Management Systems* to transparently store, manipulate and access assets, such as high availability cloud storage and asset providers (e.g. Dropbox, AWS S3). Assets can be enriched with metadata like description information, author, and copyright. The information is stored in local database tables. All reference to files used in content elements should use the FAL API.

Maintainer: TYPO3 (Oliver Thiele)

Translations with XLIFF 1.2 and 2.x

TYPO3 uses the XLIFF (XML Localisation Interchange File Format) standard for managing translatable labels. This skill covers the usage and structure of XLIFF 1.2 and XLIFF 2.x files, the required elements such as file, unit/trans-unit, segment, source and target, as well as naming conventions for XLIFF IDs and the correct file storage location within an extension.

Maintainer: TYPO3 (Oliver Thiele)

Performance

Caching Basics

Caching overview and purpose.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Caching

Caching basics in TYPO3. Functionality and types of caching as well as an overview of the different caches.

Maintainer: TYPO3 (Oliver Thiele)

Installation

Upgrade Wizards

The Upgrade Wizard is a tool available to system maintainers that checks for and executes pending migration tasks after a TYPO3 Core or extension upgrade. Integrators need to know when and how to run these wizards, what prerequisites must be met beforehand, and how to execute them both via the backend and the CLI.

Maintainer: TYPO3 (Oliver Thiele)

Important Actions in the System Maintainer Area

The system maintainer area groups all administrative backend modules that go beyond content editing and user management. It is only accessible to backend users who have both admin privileges and system maintainer status. If the backend is unavailable, the Install Tool can be accessed directly via URL after filesystem verification. Sometimes it's necessary to clear all TYPO3 caches, including the opcode and dependency injection caches in the system maintainer area/the Install Tool (e.g., after major upgrades).

Maintainer: TYPO3 (Oliver Thiele)

System Requirements

TYPO3 requires a specific PHP configuration and a set of PHP extensions to run correctly. The database user needs defined privileges, and the choice of database system determines which PHP extensions must be installed. Knowing these requirements is essential before setting up or evaluating a TYPO3 environment.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Directory Structure

A Composer based TYPO3 installation uses a clearly defined directory layout that separates the public web root, private configuration, vendor packages, and temporary files. Integrators need to know which directories serve which purpose, which must be publicly accessible, and which must never be exposed via the web server.

Maintainer: TYPO3 (Oliver Thiele)

Log Rotation

RotatingFileWriter configuration variables that set log file size. This enables efficient log rotation and can be set in shared hosting without needing root access.

Maintainer: TYPO3 (Oliver Thiele)

Getting Help

TYPO3 Documentation

I know where to find documentation and support.

Maintainer: TYPO3 (Oliver Thiele)

Security

About the TYPO3 Security Team

How to deal with security issues, both found and reported.

Maintainer: TYPO3 (Oliver Thiele)

Security best practices in TYPO3

As a TYPO3 integrator I know the different types of security vulnerabilities and how to secure a TYPO3 installation.

Maintainer: TYPO3 (Oliver Thiele)

Multi-Factor Authentication (MFA) configuration

MFA can be configured for individual users/groups and extended by extensions.

Maintainer: TYPO3 (Oliver Thiele)

Password Policies

This skill covers the configuration of TYPO3 password policy to bolster security for both backend and frontend users. It includes understanding how to employ built-in validators and awareness of the possibilities for creating custom validators.

Maintainer: TYPO3 (Oliver Thiele)

Understanding Content Security Policy (CSP)

Content Security Policy (CSP) is a security standard introduced to prevent cross-site scripting (XSS), clickjacking and other code injection attacks resulting of malicious content being executed in the trusted web page context.

Maintainer: TYPO3 (Oliver Thiele)

Configuring Content Security Policy (CSP) in TYPO3

TYPO3 includes a backend module for CSP reporting and supports two mechanisms for inline script and style allowlisting: nonce-based and hash-based CSP. Nonces are random per-request tokens and prevent HTTP response caching. Since TYPO3 v14, hash-based CSP is available as an explicit opt-in via `csp.yaml`, allowing responses to be cached by reverse proxies or static file cache extensions while still enforcing a strict Content Security Policy.

Maintainer: TYPO3 (Oliver Thiele)

Extensions

TYPO3 Extension Repository (TER)

Knowledge about the standard extension hub of TYPO3.

Maintainer: TYPO3 (Oliver Thiele)

General Extension Architecture

Composer for TYPO3 Extensions

This skill covers usage of Composer for extensions and integration into TYPO3 projects.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Concepts

TYPO3 Error handling/Exception handling

This skill is about TYPO3-specific ways of handling errors during program execution and finding information about any errors that occurred.

Maintainer: TYPO3 (Andreas Wolf)

System Configuration Files

A Composer-based TYPO3 project uses a small set of well-defined PHP files to control the entire system configuration. Integrators need to know which file stores what, who writes to it, and how the files relate to each other — particularly `settings.php` and `additional.php` in `config/system/`, which together populate the global `$GLOBALS['TYPO3_CONF_VARS']` array.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Coding Guidelines for Integrators

TYPO3 defines coding guidelines (CGL) for the configuration file formats that integrators work with every day: TypoScript, TSconfig, XLIFF, PHP, YAML etc.. The guidelines cover mandatory file extensions, recommended directory structures, and consistent formatting rules. Following them makes code easier to read, review, and maintain across the entire team. An `.editorconfig` file in TYPO3 projects encodes these rules in a tool-agnostic way so that every developer gets the correct formatting applied automatically in their editor, regardless of which IDE they use. Without it, each IDE formats files slightly differently, which leads to unnecessary changes in Git and makes merge conflicts more likely.

Maintainer: TYPO3 (Oliver Thiele)

Package management

Composer

Composer basics and its usage for PHP projects.

Maintainer: HTL Rennweg (Oliver Thiele)

Version Control

Git basics

The most important commands for working with the Git version control system (VCS).

Maintainer: TYPO3 (Oliver Thiele)

Target Group Optimization

SEO Extension

The goals of the SEO extension are to:

- Provide editors with the fields required to set meta tags.
- Provide the rendering of meta tags in the frontend for SEO related information.

Maintainer: TYPO3 (Oliver Thiele)

Page cache

The basic idea of pre-rendering content for speed optimization and the effect it might have on content editing.

Maintainer: TYPO3 (Torben Freudewald)

PHP Framework Basics

Basic PHP skills for integrators

TYPO3 integrators traditionally worked without needing PHP knowledge. That boundary has steadily blurred: configuration files, extension registration, and environment-specific setups all involve PHP that an integrator must be able to read and cautiously edit. This skill covers the foundational PHP concepts that appear most frequently in integrator work — from variables and arrays to conditional logic and functions — so that encountering a PHP file no longer means guesswork.

Maintainer: TYPO3 (Oliver Thiele)

TYPO3 Administration

TYPO3 Major Upgrades

A major TYPO3 upgrade involves three clearly defined phases: preparation before the upgrade, updating the Core via Composer, and completing follow-up tasks in the backend afterwards. Each phase contains specific steps that must be carried out in order to ensure a clean and stable result. The process applies to every major version jump and follows the same general pattern regardless of the versions involved.

Maintainer: TYPO3 (Oliver Thiele)

Installing TYPO3

Installing TYPO3, creating an initial website, and understanding the first configuration steps are essential tasks for TYPO3 integrators. This includes setting up a TYPO3 instance with Composer and DDEV, working with the backend, managing pages and content, and configuring the default site theme and site settings.

Maintainer: TYPO3 (Oliver Thiele)

CMS Basics

Working with translations

TYPO3 offers the function to deliver contents, datasets as well as metadata multilingual. Content elements can be translated via a "free" or "connected" mode. The number of languages that can be editorially recorded is not limited in TYPO3.

Maintainer: TYPO3 (Torben Freudewald)