TYPO3 Content Blocks
Deep Dive

•••

André Kraus

#cig-structuredcontent

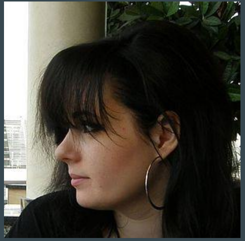
Content Types Team

André Kraus
Team lead, developer

Jonas Eberle
Frontend development

Nikita Horvatov
API development lead, Core Team bridge

Manuel Schnabel
developer, article author, social media



Lidia Demin
Ex-Lead (silent member)

Filippos Karpouchtsis
Frontend development

Philipp Kuhlmay
developer, testing, documentation

TYPO3

Content Blocks - Deep Dive

Best Practices and technical edge cases while creating content elements with the content blocks approach.

Content Types Team 08.08.2025

CLI Commands

Content Blocks comes with a few CLI commands

- kickstart command
- list command
- language generate command
- publish assets command

→ speed up setup process

→ reduce repetitive tasks

Skeleton for CLI Command

Predefined standards for creating a new Content Block

- pre-defined assets, templates and config
- for content elements, page types, record types
- except: language/labels.xlf and config.yaml
- use content-blocks.yaml for config.yaml default values

→ speed up setup process

→ reduce repetitive tasks, reduce bugs

Basics

Reduce redundant YAML code with Basics

- pre-defined Basics from Content Blocks
- custom Basics
- nested Basics
- e.g. header palette, link palette, appearance tab

→ shared code: take care when moving a Content Block

→ reduce complexity

Extending Field Types

You can add complete new field types to your system

- implements Content Blocks FieldTypeInterface
- it is just one single field, not a group of fields
- e.g. field type “Image” instead of type “File” with additional config
- e.g. type “SingleSelect” instead of renderType “selectSingle”

→ provide better fitting defaults

→ improve semantic

Extend TypoScript

Override Content Blocks boilerplate TypoScript

- global Content Blocks TypoScript overrides
- override TypoScript of a single Content Block to add specific definition
- e.g. to add Data processing, Shared Partials, layouts, settings

→ add global definitions or additional information

→ customize a Content Block

Shared partials

Reduce redundant code with shared partials

- use own partials from your site package, e.g. for header or image rendering
- for content elements via TypoScript
- for backend preview via page TSconfig

→ reduce redundant code

→ streamline rendering

Extend TCA

Override Content Blocks boilerplate TCA

- it works exactly like overriding core TCA
- find the correct identifier for the field you want to override
- remember collections creating a new custom table
- (RecordTypes creating a new custom table too)

→ typo3/cms-lowlevel is your friend (Configuration module)

→ extend the TCA in Configuration/TCA/Overrides/

Data Processing

Content Blocks is based on the new Record API in TYPO3 v13

- old way works only for new variables alongside `$processedData['data']`
- use RecordCreationEvent to modify the record object
- page data processing has to be added manually via TypoScript

→ you can use your own models and repositories

→ for low complexity or simple data collection, consider this approach instead of Extbase

Display conditions

Display conditions in Content Blocks is writing TCA in YAML

- it works exactly like in TCA
- you need the correct identifier and value
- fields used in a condition should have the column option “onChange” set to reload

→ typo3/cms-lowlevel is your friend (Configuration module)

→ please also take a look at the official documentation

TYPO3

Handle common fields in CE’s

Fields needed in nearly every content element like header, header_layout, background_color, background_image

- re-defining root fields in every content element (tt_content) can lead to table row size to large issue
- re-use existing fields in content elements
- consider to share common root fields

⟶ Attention: some columns aren’t meant to re-use!

⟶ Some properties aren’t meant to override, e.g. type

Nested Content Elements

It is possible to nest content elements within Content Blocks

- **this can not replace e.g. EXT:container**
- special setting in YAML file
- take care to prevent duplicate rendering in frontend & backend
- this solution does not provide drag and drop or creation of new child elements in the Page module

Link.editRecord ViewHelper

Improve UX in TYPO3 backend while using Content Blocks

- adds the return anchor on custom edit links - jumps back to the previous position on close element
- only in Page layout context
- also works for linking directly to collection items

Extbase Plugin

How to create Extbase plugins via Content Blocks

- faster and easier registration
- easier configuration and maintenance
- TODO: register controller actions manually
- TODO: add custom TypoScript

→ Attention: experimental feature!

Type Overrides

For field type “Collection” and “File”

- reorder fields of the child record
- remove fields from the child record
- add completely new fields to the child record
- override options like “label”, “description”, “renderType”, etc.

→ override the showitem string for specified type

→ can be used to re-define file definitions

File Types

Override file types globally

- re-define file types
- e.g. add new custom fields
- use carefully, since this changes the arrangement of fields globally

→ a Basic could be a better way to do so

→ consider using Type Overrides instead

Restrict Collection child types

Improved control over collection children

- e.g. restrict tt_content element types within a slider or an accordion
- easy configuration via YAML

→ improve user experience

→ avoid rendering issues

Content Blocks Documentation

Go ahead and find further information:
https://docs.typo3.org/p/friendsoftypo3/content-blocks/main/en-us/Index.html

Content Types Team 08.08.2025

JSON Schema for IDEs

currently available via:
https://github.com/nhovratov/content-blocks-json-schema

These slides available here

Link to the slides:
https://github.com/krausandre/cb-demo-25

Coming soon:

Moving forward to migrate to TYPO3 core

Content Blocks GUI

EXT:content_blocks_styled_content

JSON Schema moving to public API

Try it out now!

friendsoftypo3/content-blocks

slack

#cig-structuredcontent