

Content Editing Unlocked

Multisite · Visual Editor · CKEditor

...maintainable, predictable, and pleasant

Thanks for having us, **T3DD 2026!**



Ondřej Chrastina

Developer Advocate



<https://ondrej.chrastina.dev>



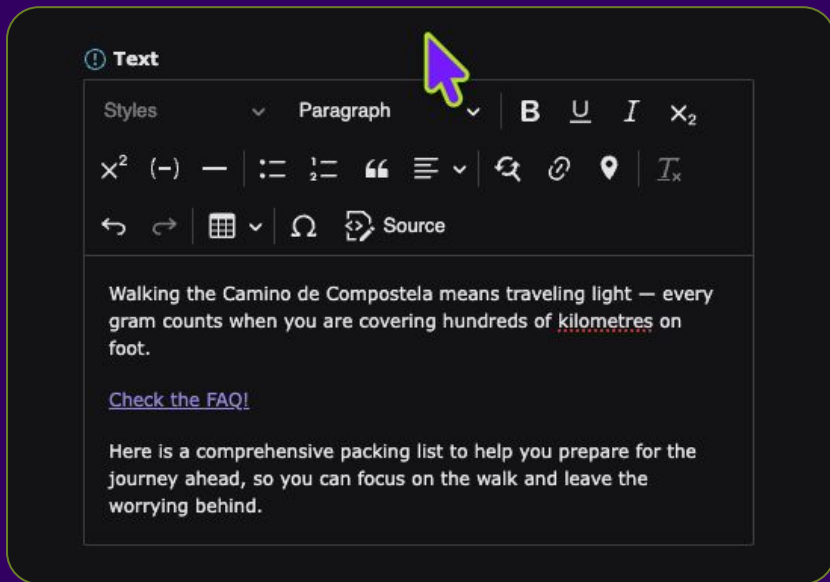
Jürgen Pietschmann

CKEditor & AI Consultant

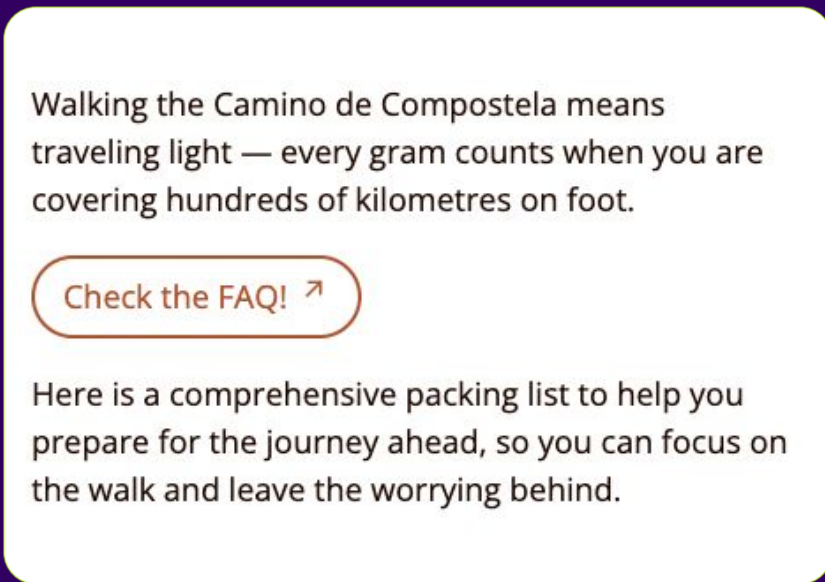
(Unable to attend due to illness)



How much power to give to editors?



The RTE preview



The live site, same content

Same content. Every bit of power you hand over is also a new way for the editor and the site to disagree.

How much power to give to editors?

How much WYSIWYG power?



Packing List

for the Camino de Compostela

Walking the Camino de Compostela means traveling light — every gram counts when you are covering hundreds of kilometres on foot.

[Check the FAQs!](#)

Here is a comprehensive packing list to help you prepare for the journey ahead, so you can focus on the walk and leave the worrying behind.

CKEditor

Where do you draw the line?

LOCKED DOWN

Structured fields only. Nothing breaks, and nothing new is possible without a developer.

Structured elements only

Curated RTE styles

Components in the text

Visual Editor

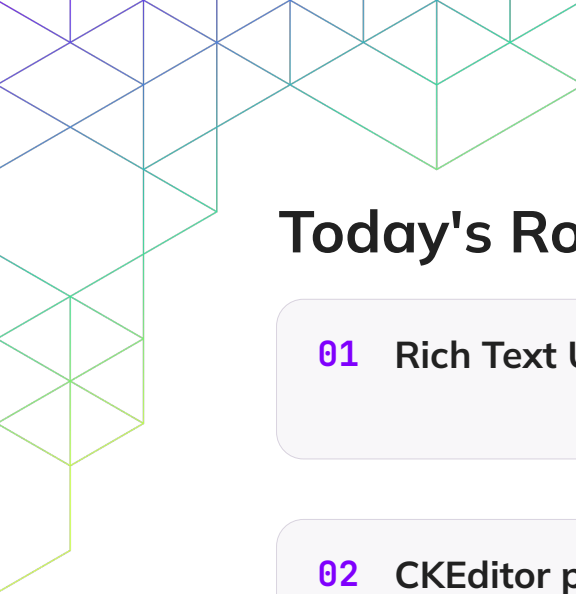
CKEditor plugin pack

Source view

FREE-FOR-ALL

Source view and free HTML. Everything is possible, including breaking the design system.

Today: move the line **A BIT** to the right, without giving up the design system.



Today's Route

01 Rich Text Unlocked components editors can place freely inside the text

Ondřej

**02 CKEditor pack
& Visual Editor** the same freedom, one level up the page

jürgen Ondřej

03 Wrap up CKEditor 47.x today, and what is coming

Both

ACT 01

Rich Text Unlocked

In four stages

Structured elements only



Components in the text

WHERE THIS PART COMES FROM

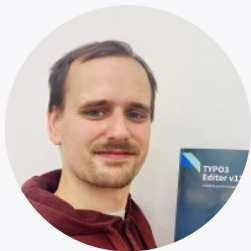
Grounded in Real Projects



Production multisite projects shared editor config, agency reality

Active hearing what integrators and editors actually struggle with

Feel the process a public demo project built for this session - based on Camino



Martin Přebyl

CEO (CZ branch)

.hyperdigital

```
# the demo project  
git clone github.com/Simply007/t3dd-2026-content-editing-unlocked
```

TYPO3 v14 base distribution + Camino theme + one custom extension

STAGE 01

Out of the Box in Camino

A **rich text** element for the copy

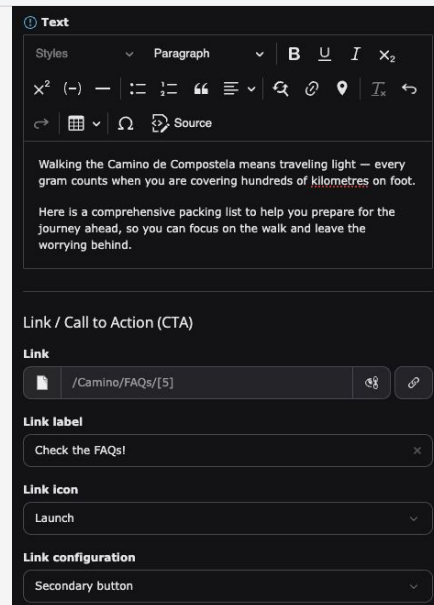
A separate **CTA button** element underneath

Primary and secondary variants, icons included, all rendered by the theme

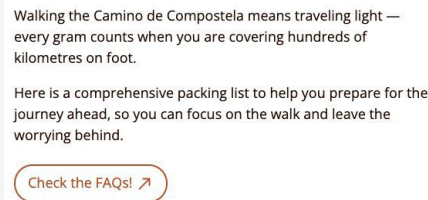
Camino renders (Fluid partial)

```
<a class="button" >
  <span class="button__content">
    <span class="button__text">Plan your route</span>
  </span>
</a>
```

The catch: the CTA is its own content element. It always sits below the text, never inside it.



Backend: a text field, then a separate CTA field group



Live: a designed button with an icon, below the text

A CTA Inside the Text

Editors select a word and hit **Link**, the native TYPO3 dialog

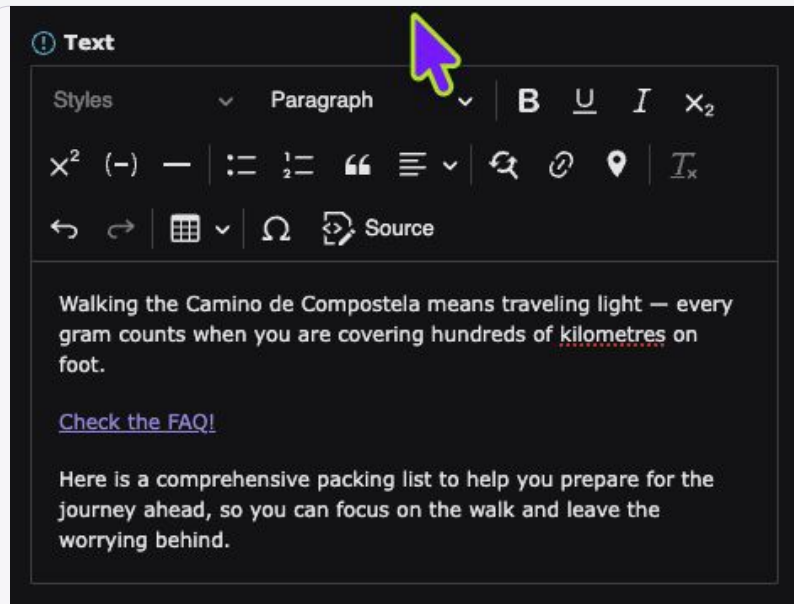
The RTE stores a bare anchor, nothing else

No shape, no icon, nothing like the CTA two paragraphs below

What the RTE stores

```
<p>Ready when you are.  
<a href="/plan" >Plan your route</a>  
</p>
```

The problem: inline, yes, but it is a plain link. Two buttons on one page, two different things.



Walking the Camino de Compostela means traveling light — every gram counts when you are covering hundreds of kilometres on foot.

[Check the FAQ!](#)

Here is a comprehensive packing list to help you prepare for the journey ahead, so you can focus on the walk and leave the worrying behind.

Inside the text, and indistinguishable from any other link

So You Style It. Two Dead Ends.

DEAD END A

Different HTML structure

CTA component produce different HTML than RTE link.

DEAD END B

Duplicate the styles

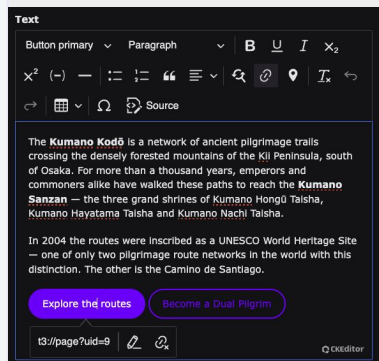
You need to duplicate styles among backend (RTE preview) and live site.

One cause: the editor and the live site disagree about what a button is.

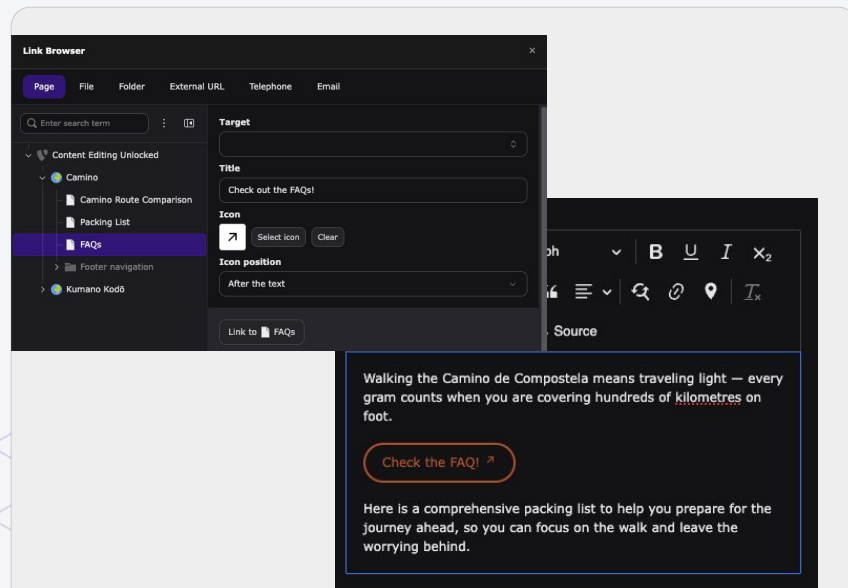
STAGE 04

The Bar We Set

- 1 Looks **exactly like the CTA**, in the editor and on the live site
- 2 Supports **icons**
- 3 Editable in the **native TYPO3 link dialog**, no custom form
- 4 Defined **once**, active on every site in the installation



Respecting site theme



Icon and icon position, in the native link browser

Walking the Camino de Compostela means traveling light — every gram counts when you are covering hundreds of kilometres on foot.

[Check the FAQ! ↗](#)

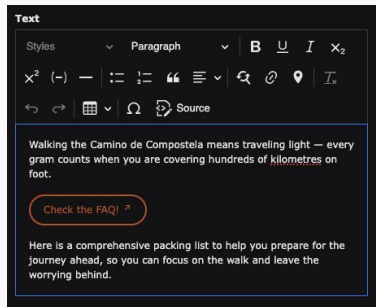
Here is a comprehensive packing list to help you prepare for the journey ahead, so you can focus on the walk and leave the worrying behind.

And the live site renders exactly that

It Was Never Just About the Button

Custom CSS in RTE

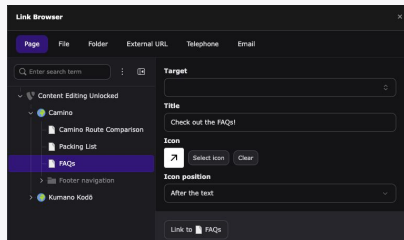
- Styles in backed and live site from same source
- Controlled by source set



.rte-button

TYPO3 Dialog

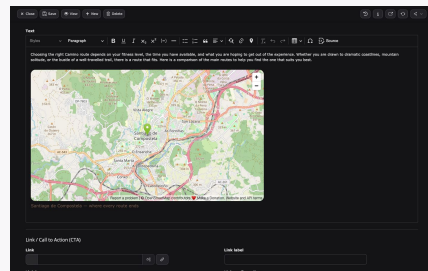
- TYPO3 dialog extension



data-icon

Custom HTML by RTE

- Customizing the RTE HTML structure



caminoMap

The pattern: pick any component and decide, deliberately, that editors may place it freely inside the editable area.

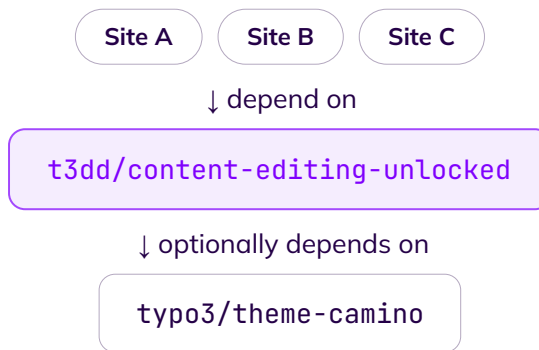
Package extension as an answer to multisite



Declare once: theme, RTE preset, and shared content styles live in one set

Opt in by dependency: sites pull the set instead of copy-pasting config

The multisite win: ten sites, one preset definition, zero drift



One Preset, Every Site

1 Configuration/Sets/Base/config.yaml

```
name: t3dd/content-editing-unlocked
label: 'Content Editing Unlocked'
optionalDependencies:
  - typo3/theme-camino
```

2 ext_localconf.php

```
$GLOBALS['TYPO3_CONF_VARS']['RTE']['Presets']['editing-unlocked' ] =
'EXT:content_editing_unlocked/Configuration/RTE/Styled.yaml';
```

3 Configuration/Sets/Base/settings.definitions.yaml

```
contentEditingUnlocked.buttonClass:
  default: 'rte-button'
```

The Two-Button Problem

What Camino renders (Fluid partial)

```
<a class="button" >
  <span class="button__content" >
    <span class="button__text" >Label</span>
  </span>
</a>
```

`_button.css` sets the text color on the inner span only

What the RTE stores

```
<p>
  <a class="button" href="/get-started" >
    Get started
  </a>
</p>
```

Bare anchor: the global `a` rule from `_base.css` wins, and `--link-color` is the primary color

Decision: RTE buttons are their own component, `.rte-button`. Same look, honest markup, no cascade fights.

THE HONEST CONS

Visual parity with Camino is maintained by hand

No automatic flips on contextual `--bg-*` sections

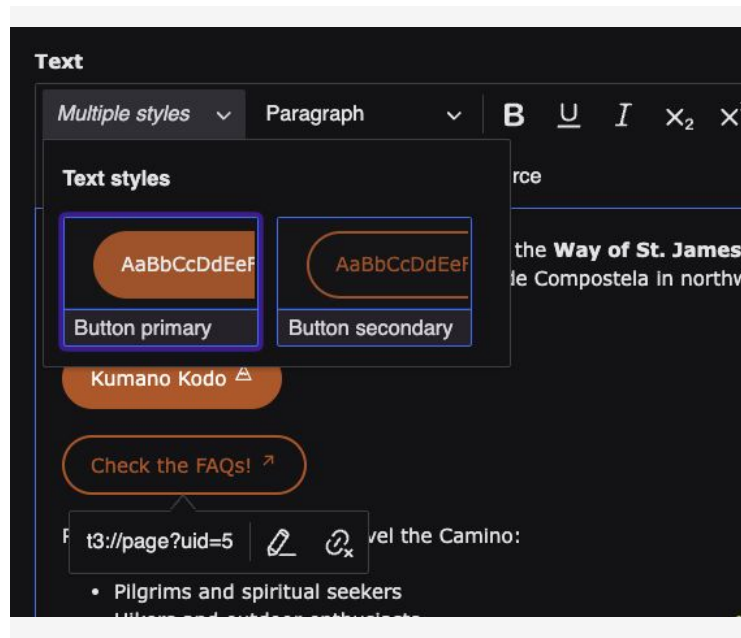
Existing content migrated once, `button` → `rte-button`

Button Styles in the Preset

Configuration/RTE/Styled.yaml

```
style:
  definitions:
    # deliberately NOT Camino's .button classes
    # a separate RTE-owned component, see NOTES.md
    - { name: "Button primary", element: "a" , classes:
      ['rte-button' ] }
    - { name: "Button secondary", element: "a" , classes:
      ['rte-button', 'rte-button--secondary' ] }
    # ...
  contentsCss:
    - "EXT:content_editing_unlocked/.../content-buttons.css"
```

The Style plugin writes classes on the anchor. contentsCss makes the preview honest.



Both buttons previewed in the editor, in the site's active scheme

One Stylesheet, Both Contexts

Resources/Public/css/content-buttons.css

```
a.rte-button {  
    background: var(--color-primary-100, #B45023) ;  
    color: var(--color-neutral-0, #FFFFFF) ;  
}  
/* outranks Camino's global a: hover */  
a.rte-button: hover: not(: disabled) {  
    Background: var(--color-neutral-70, #6C5246) ;  
}
```

The RTE preview

loaded via `contentsCss`

Every frontend

loaded via the Site Set's TypeScript

The contract: palette variables plus the spacing and corner scales, caramel-cream literals as the last resort.

Steal this:

define content styles once, against the theme's variable contract, and load the same file everywhere the content renders.



Under the Hood: Dynamic Colors

1

Detect

the `RteButtons` plugin reads the page ID from the form context



2

Fetch

backend AJAX route `..._rte_colors` with that page ID



3

Resolve

controller reads `camino.colorScheme`, serves Camino's variables CSS, scheme in `X-Color-Scheme`



4

Apply

inject the CSS and set the `theme-*` class on the editor root, like the FE's body class

Hardening next: follow core's own `contentsCss` handling, `adoptedStyleSheets` over a raw style tag, cleanup in `destroy()`, and watch the `:root` leak

Icons

The link browser round-trips **standard attributes only** : href, title, class, target, rel, download. Custom fields are collected and then dropped.

the smuggling

```
// picked in the dialog, encoded into class class=
"rte-button __icon__tabler__arrow-right
  __iconpos__before"
// decoded by the CKEditor plugin, class cleaned
data-icon="tabler:arrow-right"
data-icon-position="before"
```

- 1 Link handler** adds the icon fields, only when the dialog opens from the RTE
- 2 Picker** a searchable grid fed by a backend AJAX route
- 3 Sync** encodes the choice as pseudo-classes in the class field
- 4 Plugin** decodes them into model attributes and strips the class

Re-editing works natively: the link command hands the attributes back, so editors just reopen the dialog.

THE HONEST CON: Icon set for the Camino component and Inline button is different

Rendered by CSS Masks, Zero Frontend JS

link-icons.css (generated)

```
a[data-icon="tabler:arrow-right"]::after {  
  mask-image: url("../Icons/tabler/arrow-right.svg" );  
  background-color: currentColor ;  
}
```

One static file, loaded in the editor preview and on the frontend. Only the icons actually used are fetched, about a kilobyte each.

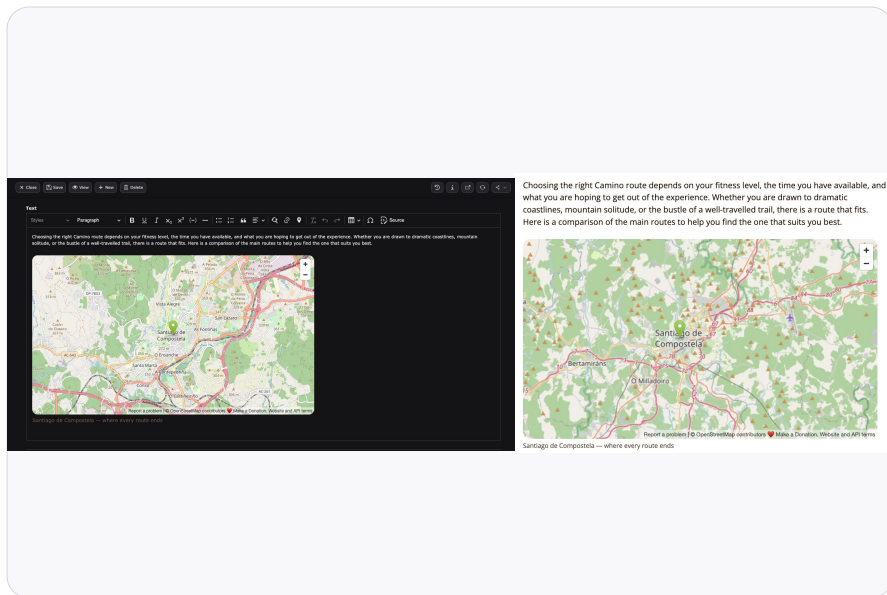
currentColor means the icon inherits the link colour, so it follows the active scheme for free

Tabler icons , MIT, stroke-based on one grid, a curated set vendored with its licence

Steal this: An icon set is content styling. Ship it as CSS, and the live site needs no JavaScript at all.

ALTERNATIVE: You can render `<svg...` directly to the HTML using JavaScript..

One Level Up: a Block Widget



A live, pannable map inside the rich text field

Steal this: store data, not embeds. The renderer stays swappable and the sanitizer has nothing to strip.

what the database stores

```
<div class="camino-map" data-lat="42.8806" data-lon="-8.5449" data-zoom="13" >  
  <p class="camino-map__caption">Santiago</p>  
</div>
```

No iframe in the database. The editor builds one in the editing downcast, the frontend script builds the same one at render time.

The caption is a nested editable, and `camino-map.css` is loaded in both contexts, exactly like the buttons.

Defaults, Not Assumptions

<code>colorScheme</code>	defer to camino.colorScheme
<code>themeVariablesCss</code>	Camino's <code>_css-variables.css</code>
<code>contentCss</code>	<code>content-buttons.css</code>
<code>buttonClass</code>	<code>rte-button</code>
<code>buttonSecondaryClass</code>	<code>rte-button--secondary</code>

Every setting overridable per site, under `contentEditingUnlocked.*`

The preset YAML is static

A PSR-14 listener rewrites the style classes and the CSS path per site, on the *Before* event so EXT: paths still resolve.

Content Editing Unlocked

Color scheme

Theme class name for the RTE preview. Takes priority over camino.colorScheme; leave empty to defer to the Camino theme.

Theme variables CSS

EXT: path of the stylesheet defining the theme CSS variables, served to the rich text editor.

EXT:theme_camino/Resources/Public/Css/_css-variables.css

Content CSS

EXT: path of the content styles (buttons) loaded into the rich text editor and included on the frontend.

EXT:content_editing_unlocked/Resources/Public/css/content-buttons.css

Primary button class

Class applied by the "Button primary" RTE style definition.

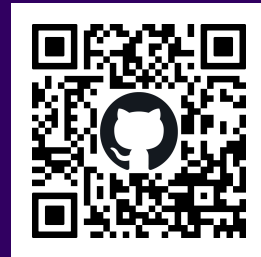
rte-button

Secondary button class

Additional class applied by the "Button secondary" RTE style definition.

rte-button--secondary

Steal this: ship your theme-specific values as defaults, and the extension outlives the theme.

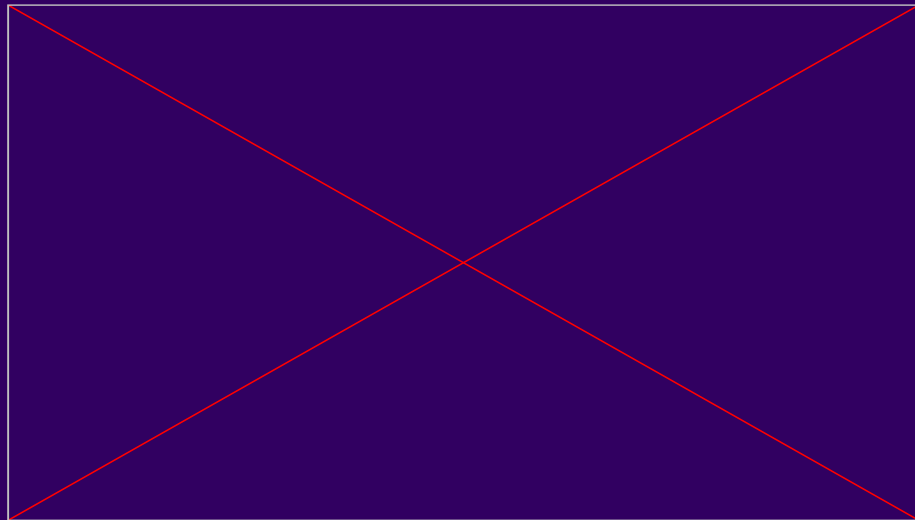


Content Editing Unlocked

Theme colors in the RTE

- 1 Pick a color scheme in Site Management → Settings
- 2 Reopen a content element with a rich text field
- 3 The styles dropdown previews in the site's colors

`One site setting. Same colors everywhere.`



ACT 02

CKEditor Pack & Visual Editor

RTE YAML definition



WYSIWYG Configuration

CKEditor Pack is an OSS project
launched by



and



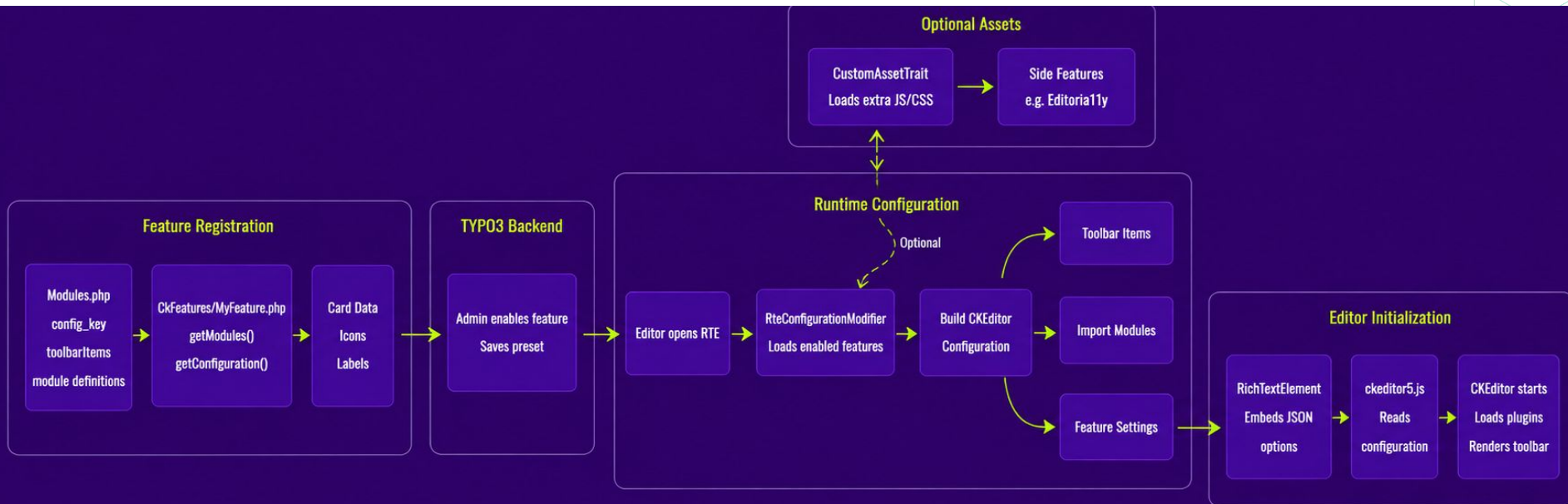
Inspired by CKEditor Plugin Pack in Drupal



A screenshot of the Drupal CKEditor interface. The top navigation bar includes 'Manage', 'Shortcuts', and 'admin'. Below it is a toolbar with icons for Content, Structure, Appearance, Extend, Configuration, People, Reports, and Help. The main editing area has a 'Body' tab and a rich text editor toolbar. The text content reads: 'Analysis of Employment Tenure at Startups', 'This study was designed to determine the effects of startup employment tenure. For the analysis, we only included employees who have received at least two equity awards, such as option grants and SARs. We looked at individual employees from their start date for up to 3 years of employment. Questionnaires were distributed to 20 different company employees through our platform.', 'The section consists of two parts:', a bulleted list with 'Findings and results' and 'Recommendations', 'Findings and results', 'The employees were also asked to choose their age group; all age ranges were represented in the answers. The breakdown consisted of 7.3% under the age of 20, 43.7% between 20 and 2928, 39.4% between the ages of 30 and 39, and 9.6% between 40 and 49.', and 'From the results, we can see that job tenures at startups are way lower than in other industries. The median job tenure, generally for us, is about 3.9 years but for the startup employees, it's only 1.7 years.' On the right side, there are panels for 'Author: admin', 'Create new revision' (checked), 'Revision log message', 'Menu settings' (Not in menu), 'URL alias' (No alias), 'Authoring information' (By admin (1) on 2023-03-02), and 'Promotion options' (Promoted to front page).

A screenshot of the CKEditor 5 toolbar configuration interface. It shows a 'Toolbar configuration' section with instructions: 'Move a button into the Active toolbar to enable it, or into the list of Available buttons to disable it. Buttons may be moved with the mouse or keyboard arrow keys.' Below this, there are two sections: 'Available buttons' and 'Active toolbar'. The 'Available buttons' section shows a list of buttons including undo, redo, style, link, unlink, source, table, tableOfContents, and language. The 'Active toolbar' section shows a list of buttons including bold, italic, strikethrough, subscript, superscript, textColor, textColorBackground, bulletedList, numberedList, decreaseIndent, increaseIndent, quote, insertImage, insertTable, link, unlink, and source. A 'Button divider' is also shown.

Technical Architecture



ACT 02 · RICH TEXT UNLOCKED · VISUAL EDITOR AND CKEDITOR PACK IN REAL WORLD

The screenshot displays the CKEditor Pack interface. On the left is a dark sidebar with navigation links: Dashboard, Content, Editor, Layout, Records, Preview, CKEditor Pack (highlighted), Status, Recycler, Forms, Media, Sites, Setup, and Page TSconfig. The main content area has a top bar with 'Editing-unlocked' and 'Import / Export' dropdowns, and 'Manage Presets' and 'Settings' buttons. Below this is a 'Toolbar' section with tabs for 'Free Packages', 'Productivity', 'Features', and 'Collaboration'. The 'Visible Feature' section contains two panels: 'Toolbar InActive Items' and 'Toolbar Active Items'. The 'InActive' panel lists 'Strike through', 'tableColumn', 'tableRow', 'mergeTableCells', 'TableProperties', 'TableCellProperties', and 'selectAll'. The 'Active' panel lists 'Style', 'Heading', 'Divider (|)', 'Bold', 'Underline', 'Italic', 'Subscript', 'Superscript', 'softyphen', 'Page Break', 'horizontalLine', 'bulletedList', 'numberedList', 'blockquote', 'Alignment', 'Find And Replace', 'link', 'insertimage', 'caminoMap', 'removeFormat', 'undo', 'redo', 'insertTable', 'Special Characters', and 'Source Editing'.

Dashboard

Content

Editor

Layout

Records

Preview

CKEditor Pack

Status

Recycler

Forms

Media

Sites

Setup

Page TSconfig

Editing-unlocked

Import / Export

Manage Presets

Settings

Toolbar

Free Packages

Productivity

Features

Collaboration

Search Toolbar Item

Toggle Label

Toolbar Groups

Advanced Settings

Visible Feature

Toolbar InActive Items

Strike through

tableColumn

tableRow

mergeTableCells

TableProperties

TableCellProperties

selectAll

Toolbar Active Items

Divider (|)

Break (-)

Style

Heading

B Bold

U Underline

I Italic

x₂ Subscript

x² Superscript

(-) softyphen

Page Break

— horizontalLine

bulletedList

numberedList

“ blockQuote

Alignment

Find And Replace

link

insertimage

caminoMap

removeFormat

undo

redo

insertTable

Special Characters

Source Editing

We listen: Key Updates and Improvement Since T3CON 2025

01

Improved YAML preset handling

Load, Sync, and Reset keep the database aligned with YAML. New CLI command (presets:sync) adds additive, ordered, strict, and reset modes.

02

Strict vs. Reset, clarified

Strict copies YAML into the database immediately; Reset clears overrides and falls back to the YAML preset on load.

03

data-* attributes preserved on paste

General HTML Support now retains data-*, id, class, and style attributes when pasting HTML into the RTE; disallowed markup is still stripped.

04

Import & export presets

Share YAML-based presets across environments. Document Outline, Paste Office Enhanced, Editoria11y, and Markdown are excluded from export.

05

Editoria11y accessibility checker

Live, in-editor checks for heading structure, alt text, link text, and contrast — flagged in real time via a toolbar button.

For TYPO3 Integrators & Developers

Core & Integration

- Built on EXT:rte_ckeditor TYPO3 core
- Content Blocks under the hood
- Built-in free plugins
- PSR-14 & DI-compatible
- Compatible with TYPO3 v12-v14

Configuration

- YAML configuration
- Manage presets (UI/UX & Code-level)
- Import/Export presets
- Create, Load from YAML, Reset & Sync presets
- Per-content-type and per-role settings

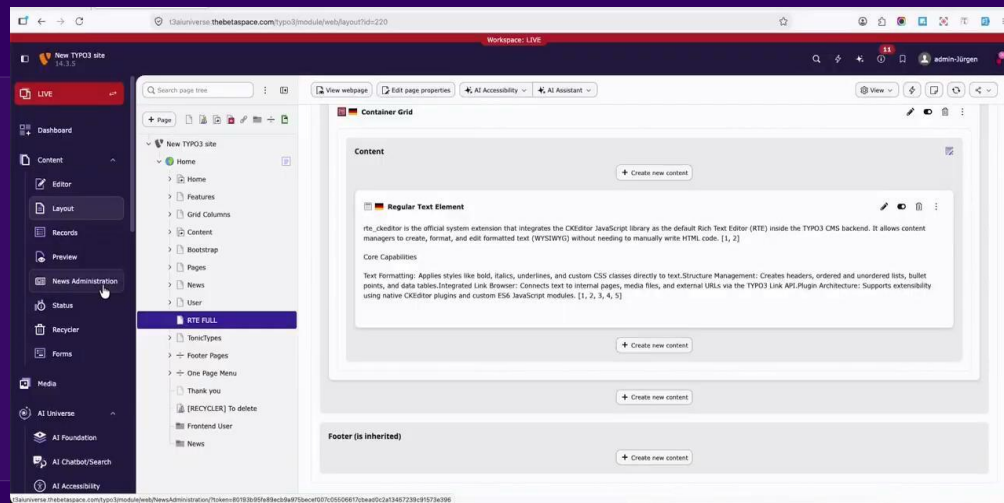
Extensibility

- Support Visual Editor Extension
 **Matthias Vogel**
- Supports custom CKEditor plugins

Case study: Automatenmarkt

Let's take a look at the case study:
AutomatenMarkt

- 1 Backend Module
- 2 CKEditor RTE
- 3 Visual Editor
- 4 AI Integration



What should be CKEditor Pack?

CKEditor Ecosystem with
Foundation





"One ecosystem, not a hundred fragmented extensions."

Our vision is to build a single, unified CKEditor Pack ecosystem for TYPO3 — replacing the scattered landscape of small, independently maintained CKEditor extensions across TER and Packagist.

~50 CKEditors Extensions Active In TYPO3 Community

The Foundation Contributions to CKEditor Pack Ecosystem

TYPO3 CKEditor Ecosystem

For TYPO3 Integrators & Developers

No-Code Setup

Setup your custom CKEditor packages without traditional code steps.

For TYPO3 Editors

Backend CKEditor Builder

Integrate custom CKEditor packages from backend wizard

1000+ NPM Packages of CKEditor can be choose and easily integrate to TYPO3. Of course, your custom one too.

Current state of extending CKEditor

Many steps to integrate your custom CKEditor plugin to TYPO3 core

1 Create Plugin

Create plugin JS

- ES6 / CKEditor 5 plugin class
- e.g. Timestamp

Register toolbar button using componentFactory

2 Register JS Module

Configuration/
JavaScriptModules.php

- Map import path

3 Configure RTE Preset

Add plugin to RTE YAML preset

importModules
Load module +
exports

toolbar.items
Add button
name

4 Register & Use Preset

Register preset
ext_localconf.php

```
$GLOBALS['TYPO3_CONF_VARS']  
['RTE']['Presets']['my_preset'] =  
    ...
```

Select preset
Page TSconfig

```
RTE.default.preset =  
my_preset
```

5 Verify

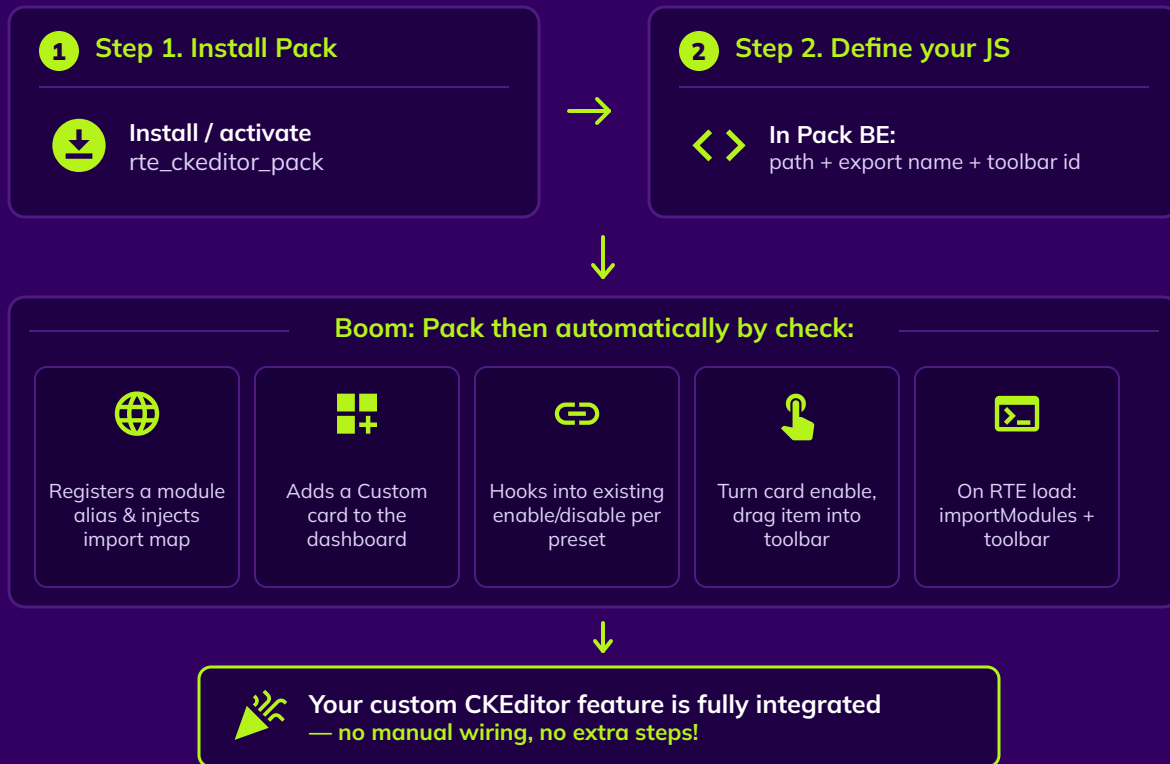
Clear TYPO3 caches

Reopen the RTE field

Verify the plugin works

Idea of CKEditor Pack

Automatically install & configure CKEditor packages to TYPO3 core



ACT 03

Wrap up

CKEditor v48+



TYPO3 15?

THX to the visual editor team!



Packing List

for the Camino de Compostela



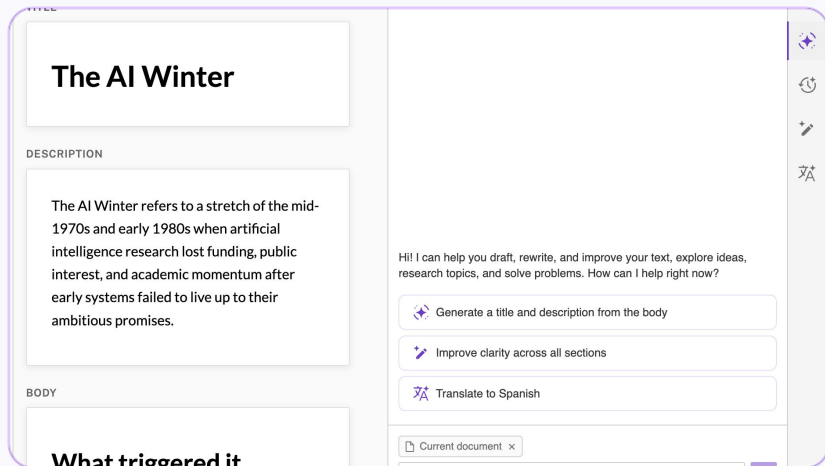
Walking the Camino de Compostela means traveling light — every gram counts when you are covering hundreds of kilometres on foot.

[↗ Check the FAQs!](#)

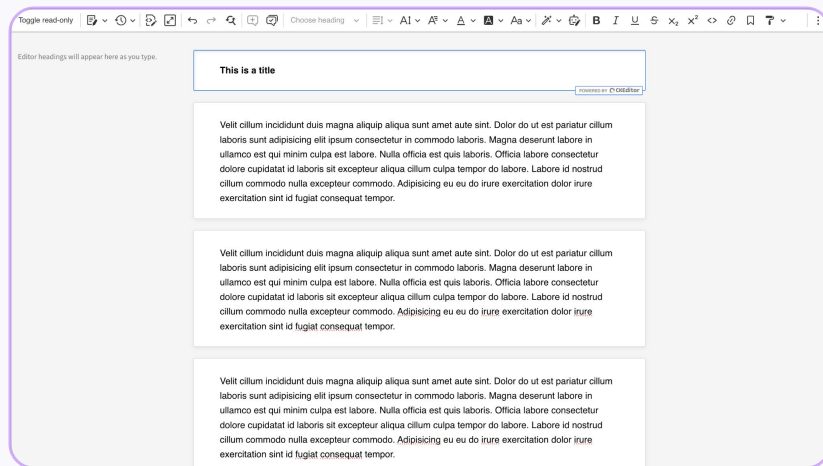
Here is a comprehensive packing list to help you prepare for the journey ahead, so you can focus on the walk and leave the worrying behind.

CKEditor

Sneak peak



Multiple fields sharing the AI context
multi-instance setup



Fields That Behave Like Fields
paragraph-like editor

Not in rte_ckeditor yet. Curious what might land in TYPO3? Come and talk to me!

Takeaways

- 1 Decide how much WYSIWYG power your editors get, then build for that answer
- 2 Share the editor configuration as a Site Set, sites opt in by dependency
- 3 One component, one stylesheet, loaded in the editor and on the live site
- 4 Icons and widgets follow the same rule: clean data attributes, rendered in both contexts
- 5 When the editor and the site disagree, the bug is the contract between them



6 Clone
Sample Project



7 Download
CKEditor Pack



8 Check YOUR
Lanyard