

Update auf Version 5.3.55

Navbar, Meta-Zeile, Brotrumen, Seitentitel, Jumbotron und Footer lassen sich jetzt auf einer einzelnen Seite abschalten – ohne die Konfiguration der ganzen Site anzufassen.

Dazu zwei neue Gestaltungsoptionen: ein seitenweiter Einzug für die Inhaltselemente, der sich auf Unterseiten vererbt, und ein Überschriftenblock, der sich um 90 Grad drehen lässt.

Und dreizehn behobene Fehler: ein fehlendes Backend-Layout im Workspace, zwei veraltete Integrity-Hashes, die Bootstrap im CDN-Betrieb vollständig blockierten, Tabellen im RTE, die ihre Bootstrap-Klasse verloren, eine Meta-Zeile, die ohne Menü nie erschien, und Rahmenlinien, die in Wrappern nicht zu sehen waren.

VERSION

5.3.55

TYPO3

^14.3

PHP

8.2 – 8.5

Hinweis	Diese Version trägt den vollständigen Inhalt von 5.3.54. Der Tag <code>v5.3.54</code> zeigte bei Packagist auf den Stand von 5.3.53, Composer-Installationen erhielten deshalb den alten Code. Veröffentlichte Versionen sind bei Packagist unveränderlich – der Inhalt erscheint daher unter neuer Nummer. Wer <code>5.3.54</code> per Composer geladen hat, aktualisiert auf <code>5.3.55</code> .
Seitenweise abschalten	Neuer Reiter T3SB in den Seiteneigenschaften: sechs Häkchen für Navbar, Meta-Zeile, Brotrumen, Seitentitel, Jumbotron und Footer.
Achtung	Neun neue Spalten – sieben in <code>pages</code> , zwei in <code>tt_content</code> . Datenbankanalyse erforderlich , danach Cache leeren und den Konfigurations-Datensatz einmal speichern.
Alle drei aus	Reiter, Einzug und senkrechte Überschrift erscheinen erst, wenn du <code>pageOptOut</code> , <code>contentIndent</code> oder <code>headerVertical</code> in der Extension-Konfiguration einschaltest.
Farben im RTE	Die sechs Bootstrap-Textfarben sind ein eigenes Toolbar-Dropdown statt zwölf Styles-Einträgen – und im Editor jetzt auch zu sehen.
Einzug pro Seite	Eine Seite rückt ihre Inhaltselemente um 1 bis 6 Raster-spalten ein, ab <code>lg</code> und nur im einspaltigen Layout. Unterseiten erben den Wert – abschaltbar über <code>contentIndentInherit</code> .
Senkrechte Überschrift	Der Überschriftenblock des Hintergrund-Wrappers lässt sich um 90 Grad drehen, nach oben oder unten gelesen – senkrecht mittig, sofern „Header inside“ an ist.
Behoben	Im Workspace fehlte auf Seiten mit Versionsdatensatz das Backend-Layout – auf einer Root-Seite war die Spaltenaufteilung weg.
Behoben	Zwei veraltete SRI-Hashes blockierten Bootstrap-CSS und -JS im CDN-Betrieb vollständig. Vier weitere Konstanten standen an zwei Orten verschieden.
Behoben	Slide-In und Fade am Dropdown liefen nie: das Markup bekam statt <code>dd-animate-1</code> immer <code>removed-animate-1</code> . Und ein Backend-Layout ohne Partial nahm die Seite mit – jetzt <code>OneCol</code> .
Behoben	Eine mit „Table Striped“ formatierte Tabelle sah nach Öffnen und Speichern wieder schmucklos aus – CKEditor verschluckt beim Lesen die Bootstrap-Basisklasse <code>table</code> .
Behoben	Die Rahmen-Option „Linie darüber/darunter“ blieb an Wrappern ohne Wirkung; am Parallax-Wrapper und am Swiper-Container fehlte sie ganz.
Behoben	Die Meta-Zeile blieb leer, sobald nur <code>meta_text</code> gefüllt war und <code>meta_value</code> nicht – <code>config.meta</code> wurde gar nicht gebaut.
Behoben	Fünf Beschriftungen blieben leer – der „Weiterlesen“-Link im Carousel, Platzhalter und Schaltfläche des Suchfelds und zwei Zeilen in der Container-Vorschau.

Behoben	Ein Element mit Überschrift, aber ohne Unterzeile bekam ein leeres <code><p class="subheader"></code> – und ein Element nur mit Unterzeile ein leeres <code><h2></code> .
Behoben	Carousel- und Swiper-Bilder zeigten keinen Copyright-Hinweis. Und „Gallery columns“ stand bei jedem Inhaltselement unter „Extended“, auch ohne Bilder – es heißt jetzt <i>Breite des Bildblocks (Spalten)</i> .
Behoben	<code>content_margin_top</code> in Spalten-Containern, der Seitentitel ohne Jumbotron-Container und der letzte Absatz in Alerts.
Behoben	Ein Backend-Stylesheet der Extension verbog das Karten-Raster in <i>allen</i> Backend-Modulen – betrifft 5.3.30 bis 5.3.53 (Issue #527).
Behoben	Das Offcanvas trug eine Klasse namens <code>undefined</code> ; Suchfeld und Ausrichtung im Panel sind geradegezogen.
Gepflegt	Zwei CDN-Versionen nachgezogen (Baguette Box, GLightbox); die Beschreibungen von Jarallax und Swiper behaupteten, ihr gepinnter Major sei der neueste.

1

Reiter „T3SB“ in den Seiteneigenschaften

Bisher galt die T3SB-Konfiguration für den ganzen Seitenbaum. Wer auf einer einzelnen Landingpage die Navbar loswerden wollte, brauchte einen eigenen Konfigurations-Datensatz. Jetzt reichen sechs Häkchen:

```
tx_t3sbootstrap_navbar_disable    tx_t3sbootstrap_jumbotron_disable
tx_t3sbootstrap_meta_disable     tx_t3sbootstrap_pagetitle_disable
tx_t3sbootstrap_breadcrumb_disable tx_t3sbootstrap_footer_disable
```

Der Reiter ist **standardmäßig aus**. Er erscheint erst, wenn du in der Extension-Konfiguration `pageOptOut` einschaltest. Und ein einzelnes Häkchen zeigt sich nur, solange der zugehörige Bereich im T3SB-Konfigurations-Datensatz aktiv ist – wer die Navbar site-weit abgeschaltet hat, bekommt dafür kein Häkchen zu sehen.

`ConfigProcessor::applyPageOptOut()` setzt die betroffenen Werte auf genau das, was der Konfigurations-Datensatz speichern würde – Leerstring bei Auswahlfeldern, `0` bei Häkchen –, und zwar bevor irgendein Bereich berechnet wird. Damit braucht kein Template eine zweite Prüfung.

EINE AUSNAHME: DIE META-ZEILE

Sie liest `settings.config.metaEnable` direkt aus dem TypoScript und bekommt deshalb einen eigenen Schlüssel `config.pageOptOut.meta`.

Schaltest du `pageOptOut` wieder aus, verschwindet nur der Reiter. Gespeicherte Häkchen bleiben wirkungslos liegen, bis er wieder eingeschaltet wird.

Warum der Reiter erst nach der TCA-Kompilierung kommt

`addToAllTCAtypes()` in den TCA-Overrides erreicht nur die Seitentypen, die in diesem Moment schon existieren. Eine Extension, die später einen eigenen Typ mit vollständigem `showitem` definiert, wirft den Reiter wieder hinaus. Deshalb hängt `TcaCompilation::addPageOptOutTab()` ihn erst nach der Kompilierung an, wenn alle Typen bekannt sind.

Seitentypen ohne eigene Ausgabe bleiben außen vor: Verweis (3), Shortcut (4), Trenner (199), Ordner (254) und Papierkorb (255). Ein Häkchen zum Verbergen der Navbar hätte dort keinen Sinn.

2 Textfarben als eigenes Dropdown im RTE

Die sechs Bootstrap-Textfarben standen bisher zwölfmal in der Styles-Auswahl: einmal als **Text Color *** für die Markierung, einmal als **Block Color *** für den Absatz. Sie haben jetzt ein eigenes Werkzeug in der Toolbar, so wie die Badges seit 5.3.51. Die Styles-Auswahl schrumpft dadurch von 41 auf 29 Einträge.

Das Dropdown führt beide Gruppen, getrennt durch einen Strich: oben **Text Primary** bis **Text Warning** für die Markierung, darunter **Block Primary** bis **Block Warning** für jeden berührten Absatz. **No color** räumt beides ab. Jede Zeile trägt einen Farbpunkt – rund für die Markierung, eckig für den Absatz.

KEINE MIGRATION NÖTIG

Das Markup ist Zeichen für Zeichen dasselbe wie vorher, `<span class="text-primary">` und `<p class="text-primary">`. Bestehende Inhalte werden beim Öffnen erkannt, lassen sich über das Dropdown ändern und gehen unverändert zurück.

Die Einstellung heißt jetzt **rteColor** statt **rteStyleColors**, und der **RteColorSettingUpgradeWizard** zieht einen gespeicherten Altwert um. Wer die Farben abgeschaltet hatte, behält sie abgeschaltet – auch ohne den Wizard, solange der alte Schlüssel in der Konfiguration steht.

Und man sieht sie jetzt auch

rte.css bekommt die sechs Farbgelb. Bootstrap wird nicht in den Editor geladen, deshalb blieb eine gesetzte Textfarbe dort bisher schwarz – man sah erst im Frontend, was man getan hatte. Das galt schon für die alten Styles-Einträge.

3 Einzug für die Inhaltselemente einer Seite

Eine Seite kann ihre Inhaltselemente einrücken, gemessen in Rasterspalten von 1 bis 6. Das Feld steht im Reiter **T3SB** der Seiteneigenschaften. Der Einzug greift ab dem Breakpoint **lg** und nur im einspaltigen Backend-Layout – darunter und in mehrspaltigen Layouts bleibt alles, wie es war.

Unterseiten erben den Wert, bis eine von ihnen einen eigenen setzt. Leer heißt erben, **0** heißt ausdrücklich keiner – damit lässt sich ein geerbter Einzug auf einem Zweig gezielt wieder abschalten.

Startseite	3	→	drei Spalten
Leistungen	(leer)	→	erbt die drei Spalten
Beratung	1	→	eine Spalte
Kontakt	0	→	kein Einzug

BootstrapProcessor::resolveContentIndent() läuft dafür die Rootline von der aktuellen Seite nach oben und nimmt den ersten gefüllten Wert.

Wer das nicht will, schaltet **contentIndentInherit** aus: Dann zählt allein die Seite, auf der der Wert steht, und Unterseiten bekommen nichts. Die Auswahlliste verliert dabei den Eintrag *Von der übergeordneten Seite erben* – es gibt nichts mehr zu erben –, und der leere Wert heißt dort schlicht *Kein Einzug*. Eine gespeicherte **0** bedeutet dasselbe; der Schalter lässt sich also hin und her stellen, ohne dass ein Wert umgesetzt werden muss.

Ein einzelnes Inhaltselement nimmt sich im Reiter „Erscheinungsbild“ über `tx_t3sbootstrap_no_indent` aus. Vier Element-Typen sind ohnehin ausgenommen, weil sie über die ganze Breite gehören: Hintergrund- und Parallax-Wrapper sowie Carousel- und Swiper-Container. Die übrigen Wrapper rücken mit ein.

DIE KLASSE SITZT AM ELEMENT MIT DER ID

`t3sb-indent-1` bis `t3sb-indent-6` hängt an `div#c{uid}`, damit die Überschrift über dem Block mit umzieht. Bei einem Container druckt das Layout sie als eigenes Attribut neben die ID, bei einer Button-Gruppe am Gruppen-Element.

Der Schalter heißt `contentIndent` und ist **aus**, `contentIndentInherit` steuert die Vererbung und ist **an**. Den Einzug abzuschalten versteckt nur die Felder; gespeicherte Werte bleiben stehen.

4 Überschrift senkrecht ausgeben

Der Überschriftenblock eines Inhaltselements – Überschrift, Unterzeile und Überzeile – lässt sich um 90 Grad drehen, nach oben oder nach unten gelesen. Das Feld „Vertical header“ steht im Reiter „Kopfzeilen“. Die Drehung gilt ab `lg`; darunter steht der Text wieder normal.

`writing-mode: vertical-rl` erledigt die Drehung, `transform: rotate(180deg)` die Leserichtung nach oben.

Liegt die Überschrift über „Header inside“ **im** Hintergrund-Wrapper, wird sie dort senkrecht mittig gesetzt: `position: absolute` mit `top: 50%` und einem Ausgleich über `translateY(-50%)`.

DER WRAPPER MUSS DER BEZUGSRAHMEN SEIN

Ohne positionierten Vorfahren bezieht sich `top: 50%` auf das nächste äußere Element – die Überschrift landete dann im Element unter dem Wrapper. `BackgroundWrapper` setzt in genau diesem Fall `position: relative`.

DAS FELD HÄNGT AN „HEADER INSIDE“ – IN BEIDE RICHTUNGEN

Es erscheint im Formular nur im Hintergrund-Wrapper und nur, solange die Flexform-Option *Header inside* an ist (`TcaMatcher::headerInside()` als `displayCond`) – und gerendert wird die Drehung auch nur dort. Ohne Wrapper gäbe es nichts, worin der herausgelöste Block zentriert werden könnte; ein Feld, das sichtbar ist und nichts bewirkt, wäre die schlechtere Hälfte.

Die Bedingung liest den Wert aus dem Datensatz, den FormEngine übergibt, nicht aus der Datenbank:

`headerInside` hat jetzt `onChange = reload`, und dieser Neuaufbau trägt die aktuellen Formularwerte, ohne zu speichern – das Häkchen wirkt damit sofort.

Der Schalter heißt `headerVertical` und ist **aus**. Ihn abzuschalten versteckt nur das Feld; ein gespeicherter Wert wirkt wieder, sobald „Header inside“ erneut an ist.

5 Behoben: kein Backend-Layout im Workspace

Auf einer Seite mit Versionsdatensatz fehlte das Backend-Layout; auf einer Root-Seite war die Spaltenaufteilung damit komplett weg.

FormEngine bearbeitet im Workspace den Versionsdatensatz und reicht genau dessen `uid` an die Conditions weiter: `DatabaseEditRow` liest ihn über seine eigene uid, ohne Workspace-Overlay, und

`BackendLayoutView::determinePageId()` gibt diese uid durch. Der Konfigurations-Datensatz hängt aber an der **Live**-Seite und hat nie eine eigene Version – die Tabelle ist nicht workspace-fähig. Die Abfrage mit der Versions-uid fand deshalb nichts, jedes `colPosList()` lieferte `false`.

WARUM ES AUF UNTERSEITEN WENIGER AUFFIEL

Dort findet die Suche über die Rootline eine Elternseite. Auf einer Root-Seite sitzt über dem Versionsdatensatz nur uid 0 – und es gibt nichts mehr aufzufangen.

`resolveLivePageId()` nimmt jetzt `t3ver_oid`, sobald einer gesetzt ist. Die Rootline-Suche wurde bei der Gelegenheit vereinfacht und in `getParentPageIds()` ausgelagert.

Dieselbe Stolperstelle traf den Info-Block des Seitenmoduls: `PageInfo` lief die Rootline über `RootlineUtility` ab, und die scheitert an einer Seite, die es nur im Workspace gibt. Dort steht jetzt `BackendUtility::BEgetRootLine()`, das den Workspace kennt.

6

Behoben: zwei veraltete Integrity-Hashes

`bootstrap.cdn.bootstrapCssIntegrity` und `...JsIntegrity` standen in `constants.typoscript` auf den Hashes einer älteren Bootstrap-Fassung, während `settings.definitions.yaml` die richtigen führte. Beide Orte sind redundant: Wer das Set „OTHER SETTINGS“ nicht einbindet, bekommt die Konstanten – und damit zwei falsche Hashes.

DER BROWSER VERWEIGERT BOOTSTRAP DANN VOLLSTÄNDIG

Kein CSS, kein JS, und als einziger Hinweis eine SRI-Meldung in der Konsole. Beide Werte sind gegen `bootstrap-5.3.8.tgz` nachgerechnet und angeglichen; der Bundle-Hash war bereits korrekt.

Vier weitere Werte standen an beiden Orten verschieden und sind auf den Stand der Site-Settings gebracht:

<code>cdn.swiper</code>	11 → 12
<code>image.maxWidthToast</code>	20 → 40
<code>image.minimumWidth</code>	true → 0
<code>backToTopTitle</code>	„to tops“ → „to top“

Zwei CDN-Versionen nachgezogen

`cdn.baguetteBox` steht jetzt auf 1.13.0 statt 1.11.1, `cdn.glightbox` auf 3.3.1 statt 3.2.0. Beide Pfade sind gegen das jeweilige Paket geprüft – cdnjs liefert `baguetteBox.min.js` unter der neuen Versionsnummer, und GLightbox behält `dist/css/` und `dist/js/`.

Die Beschreibungen von `cdn.jarallax` und `cdn.swiper` behaupteten, der gepinnte Major sei zugleich der neueste. Das stimmt bei beiden nicht mehr – Jarallax steht bei 3.x, Swiper bei 14.x. Sie sagen jetzt, was der Pin tut, ohne eine Zahl zu nennen, die beim nächsten Release wieder falsch ist.

GLIGHTBOX WIRKT NUR LOKAL

Der Wert wird leicht überschätzt: Er greift allein im Kommando `t3sbootstrap:cdnToLocal`. Das CDN-Markup lädt GLightbox ohne Versionsangabe, also immer die neueste Fassung. Das steht ab jetzt auch in der Beschreibung des Felds.

Behoben: Dropdown-Animation blieb dauerhaft aus

Slide-In und Fade wurden ignoriert; das Dropdown bekam statt `dd-animate-1` die Klasse `removed-animate-1`.

Die Umbenennung ist gewollt – unterhalb des Navbar-Breakpoints soll keine Animation laufen. Sie griff nur immer. Fehlt im Body das Attribut `data-navbar-breakpointWidth`, fällt `MainAssets.fluid.html` auf `99999` zurück. Für das Offcanvas ist das die sichere Annahme, für die Animation die falsche: `vw < 99999` stimmt auf jedem Bildschirm.

Zwei Stellen sind nachgezogen:

- `OutsourcedFiles::resolveBreakpointWidth()` greift jetzt auf die Bootstrap-Standardbreiten zurück, wenn die Site-Settings keine liefern – das ist der Fall, sobald das Set „OTHER SETTINGS“ fehlt, denn dort sind sie definiert.
- `t3sbDropdownAnimate()` liest einen Breakpoint, den kein Bildschirm erreicht, als „kein Breakpoint“ statt als „immer mobil“.

Behoben: fehlendes Partial riss die Seite ab

Ein Backend-Layout ohne passendes Partial unter `Partials/BackendLayouts/` warf eine `InvalidPartialException` und nahm die komplette Seite mit – ein weißer Bildschirm wegen eines falschen Werts in `pages.backend_layout`.

Das traf jedes selbstgebaute Layout ohne eigenes Partial, und es traf die `_Extra`-Schlüssel älterer Installationen, bei denen der `BackendlayoutUpgradeWizard` nie gelaufen war. `Main.fluid.html` fällt jetzt auf `OneCol` zurück, statt abzubrechen.

Behoben: Meta-Zeile ohne `meta_value`

Gemeldet für 5.3.53: Ist `meta_value` leer und `meta_text` gefüllt, erschien die Meta-Zeile nicht. Die Bedingung in `Main.fluid.html`

```
<f:if condition="{metaNavigation} || {config.meta.text}">
```

konnte nie wahr werden, weil **beide** Werte fehlten: `metaNavigation` erzeugt der Menü-Prozessor nur aus `meta_value`, und `config.meta` wurde im `ConfigProcessor` überhaupt nicht gebaut. Im `f:debug` tauchte deshalb keiner der beiden auf.

Der Schaden reichte weiter als der Bericht: `Partials/Page/Meta` liest *jeden* seiner Werte aus `config.meta`. Auch mit gefülltem `meta_value` kam dort nichts an – `{config.meta.container}` war leer, blieb damit ungleich `none` und druckte `<div class=" d-flex justify-content-">` um die Zeile.

`ConfigProcessor::processMeta()` baut den Block jetzt. `meta_enable` trägt dabei die Ausrichtung, so wie `navbar_enable` das Farbschema trägt – leer heißt aus, sonst `start`, `end` oder `scroller`. Ein nicht gespeicherter Container wird zu `none` statt zu einem leeren `class`-Attribut, und die eigene Klasse bekommt ihr führendes Leerzeichen, weil das Partial sie als `nav-meta{class}` ausgibt.

EIN LEERES FELD IST IN DEN KONSTANTEN EINE „0“

`OutsourcedFiles` schreibt ein leeres Feld als `0`, in `settings.config` kommt also die Zeichenkette `"0"` an, nicht `""`. Mit `??` wäre daraus `nav-meta 0` geworden – der Block prüft deshalb mit `?:`.

Der seitenweise Opt-out wirkt unverändert: `applyPageOptOut()` leert `metaEnable`, bevor der Block gebaut wird.

10 Behoben: fünf leere Beschriftungen

`f:translate` gibt für einen Schlüssel, den es nicht gibt, den Leerstring aus – man sieht also nichts und bekommt auch keinen Hinweis. Fünf Stellen traf das:

Carousel, Link unter dem Text `locallang.xlf:read-more` fehlte → *Read more / Weiterlesen*

Suchfeld, Platzhalter `locallang.xlf:form.search` fehlte → *Search term / Suchbegriff*

Suchfeld, Schaltfläche `locallang.xlf:form.submit` fehlte → *Search / Suchen*

Container-Vorschau, 2 und 4 Spalten griff auf `flexform.cw.equalWidth`, jetzt auf `flexform.equalWidth`

Toast-Vorschau, Platzierung suchte in `locallang_be.xlf`, jetzt in `locallang_db.xlf`

Die unteren beiden waren keine fehlenden Texte, sondern falsche Referenzen – die Schlüssel gab es längst, nur unter anderem Namen beziehungsweise in der anderen Datei.

Dazu zwei Einträge, die nur auf Englisch vorlagen: *Expanded Content Top* und *Background Images*. Beide Sprachdateien tragen jetzt dieselben 841 Schlüssel.

11 Behoben: leeres Markup im Kopfblock

Ein Inhaltselement mit Überschrift, aber ohne Unterzeile bekam trotzdem ein `<p class="subheader">` – leer. Umgekehrt stand über einem Element, das nur eine Unterzeile, ein Datum oder eine Überzeile trägt, ein leeres `<h2>`. Am auffälligsten im Hintergrund-Wrapper, wo der Kopfblock über oder auf dem Bild sitzt.

Gemessen gibt `Partials/Content/Field` bei leerem Feld nicht den Leerstring zurück, sondern **ein Byte**: `\n`. Fluid entfernt beim Einlesen eines Partials die umschließenden `<html>`-Tags, behält aber, was dahinter steht – und das ist der Zeilenumbruch am Dateiende. In Fluid ist eine Zeichenkette aus Leerraum `TRUE`:

```
{subheader}           → "\n"    → f:if TRUE
{subheader -> f:format.trim()} → ""    → f:if FALSE
```

`Header/All` prüft längst mit der getrimmten Fassung, reicht aber die ungetrimmte an `Header/SubHeader` weiter, und dort stand `<f:if condition="{subheader}">`. Dieselbe Falle hatte 5.3.52 schon beim Supraheader behoben – der Kommentar in `Content/Field` warnt vor einem Umbruch vor dem `<f:if>`, nicht vor dem am Dateiende.

WARUM NICHT EINFACH ÜBERALL TRIMMEN?

`f:format.trim` escaped seine Ausgabe. Ein `<br>` in der Überschrift käme damit als Text heraus. Getrimmt wird deshalb nur für die Bedingung, ausgegeben weiter der ungetrimmte Wert.

Beides ist jetzt zu: `Content/Field` endet ohne abschließenden Zeilenumbruch, und die beiden Partials prüfen auf den getrimmten Wert – damit ein Editor, der den Umbruch wieder einfügt, nichts kaputt macht.

`Header/Header` hatte zusätzlich gar keine Prüfung: Es druckte das `<hN>` immer, sobald `Header/All` den Block rendert, und das tut es, sobald *irgendeines* der vier Felder gefüllt ist. Ein Icon ohne Text zählt dabei als Überschrift und wird weiter ausgegeben.

12 Behoben: kein Copyright in Carousel und Swiper

Ein Bild im Carousel zeigte seinen Copyright-Hinweis nicht; dasselbe Bild als normales Inhaltselement schon. Im Swiper fehlte er ebenso.

Die Verwechslung liegt nahe, denn `Partials/Content/Media/Carousel.fluid.html` hatte eine Section „Copyright“, und die wurde dort auch gerendert – nur ist dieses Partial für die Galerie-Variante „Carousel“ in Textmedia, Mediaobject und Card zuständig. Das Inhaltselement „Carousel“ im Carousel-Container rendert

`Templates/Content/Carousel.fluid.html`, und dort fehlte die Ausgabe vollständig. In den drei Swiper-Partials ebenso.

Statt den Block nun viermal zu wiederholen, steht er in einem eigenen Partial `Media/CopyrightNote`. Es erwartet `{file}` und wird von diesen fünf Stellen eingebunden:

Templates/Content/	<code>Carousel.fluid.html</code> – das Inhaltselement <code>arguments="{file: files.0}"</code>
---------------------------	---

Partials/Content/Media/	<code>Carousel.fluid.html</code> – die Galerie-Variante <code>arguments="{file: media}"</code>
--------------------------------	---

Partials/Content/Swiper/	<code>Default</code> , <code>CustomOne</code> und <code>CustomTwo</code> <code>arguments="{file: file}"</code>
---------------------------------	---

Der Hinweis steht jeweils im `figure` hinter dem Bild und vor einer Bildunterschrift – derselben Stelle wie bei `Media/Type/Image`. Dort muss er stehen, denn `.img-copyright-default` zieht sich mit einem negativen Margin über die Unterkante des Bildes und braucht das Bild als vorangehendes Geschwister.

KEINE TOAST-VARIANTE IN SLIDERN

Die Einstellung `imgCopyright = 2` zeigt den Hinweis sonst erst auf Klick, in einem Toast. Dessen Skript hängt an `.image .img-copyright`, und ein geöffneter Toast würde beim nächsten Slide-Wechsel mitsamt seinem Slide weggeschoben. In Carousel und Swiper erscheint deshalb immer der eingebettete Hinweis.

Das Galerie-Partial prüfte außerdem allein `tx_t3sbootstrap_copyright`. Ein Bild, bei dem nur das FAL-Feld `copyright` gefüllt ist, blieb dort ohne Hinweis, während es als normales Bild einen bekam. Das neue Partial behandelt die zwei Felder überall gleich – das FAL-Feld gewinnt.

Eine Besonderheit in `Swiper/Default`: Dort liegt das ganze `figure` in einem `f:link.typolink`. Ein Klick auf den Copyright-Hinweis folgt also dem Bildlink.

13 Behoben: „Gallery columns“ bei jedem Element

`tx_t3sbootstrap_inTextImgColumns` hatte als einziges Feld der Palette „T3SB Image Settings“ keine `displayCond`. Bei Elementen ohne Bilder fielen die übrigen Felder weg, und übrig blieb ein nackter Kasten unter „Extended“ – an einer Stelle, an der das Feld ohnehin wirkungslos ist.

Es erscheint jetzt nur noch bei den sechs Bild-CTypes. Beschriftung und Beschreibung kommen aus der Sprachdatei statt hartkodiert aus der TCA und heißen jetzt **Breite des Bildblocks (Spalten)**.

WARUM DER NAME SICH ÄNDERT

„Gallery columns“ legte eine Verwandtschaft zu „Number of Columns“ (`imagecols`) nahe, die es nicht gibt. Das Feld bestimmt die Breite des umflossenen Bildblocks in Rasterzellen und wirkt ausschließlich bei den Ausrichtungen *im Text, rechts* und *im Text, links*.

14 Behoben: Tabellen im RTE verloren ihre Klasse

Eine mit „Table Striped“ formatierte Tabelle sah nach Öffnen und Speichern wieder schmucklos aus, und die Styles-Auswahl zeigte den Eintrag nicht mehr als aktiv. Dasselbe bei „Table Hover“ und „Table“.

Der Grund steckt in CKEditor selbst. Dessen General-HTML-Support-Integration für Tabellen verbraucht beim Lesen eines `<table>` zuerst die Klasse `table` und sammelt erst danach die übrigen ein – eine verbrauchte Klasse wird nicht mehr aufgenommen:

```
o.consumable.consume(s, { classes: "table" });
n(s, "htmlTableAttributes");
```

CKEditor darf das, weil sein eigenes Tabellen-Markup `<table class="table">` als Hülle benutzt und den Namen deshalb als intern behandelt. Für Bootstrap ist genau diese Klasse aber die tragende. Der Weg durch den Editor sah so aus:

```
vorher    <table class="table table-striped">
Modell     htmlTableAttributes = { classes: ['table-striped'] }
nachher    <table class="table-striped">
```

Ohne `.table` greift keine der Bootstrap-Regeln, und `hasAllClasses` findet nicht mehr beide Klassen des Styles-Eintrags.

Das neue `table-base-plugin.js` hängt einen Upcast-Konverter mit `priority: 'lowest'` hinter den von CKEditor (`'low'`) und legt die Basisklasse zurück ins Attribut; von dort schreibt der vorhandene Downcast sie wieder an das `<table>`. Die Styles-Einträge bleiben unverändert, ein Toolbar-Element kommt nicht hinzu.

BESTEHENDE TABELLEN RÄUMEN SICH SELBST AUF

Erkennt der Konverter eine `table-*`-Variante ohne Basisklasse, ergänzt er sie ebenfalls. Schon beschädigte Inhalte sind damit beim nächsten Speichern wieder in Ordnung – der Style muss nicht neu zugewiesen werden.

15 Behoben: „Linie darüber/darunter“ in Wrappern

Die Rahmen-Option eines Elements („Erscheinungsbild“ → Rahmen) blieb an einem Wrapper ohne Wirkung. Beim Hintergrund-Wrapper war die Linie nicht zu sehen, beim Parallax-Wrapper und beim Swiper-Container fehlte sie ganz.

`Layouts/Content/Default` druckt bei einem Container **kein** `classAttr` an das `div#c{uid}`; die Klassen gehen über `{class}` an das jeweilige Template und landen dort an einem inneren Element. Für `frame-ruler-before` heißt das: Das Pseudoelement wird innerhalb des Wrappers gezeichnet, beim Hintergrundbild also hinter dem Bild.

Und `ParallaxWrapper.fluid.html` wie `SwiperContainer.fluid.html` geben `{class}` überhaupt nicht aus – dort fiel die Klasse vollständig weg.

Klassen, die das Element als Ganzes beschreiben, gehören an das Element mit der ID: die Linien, weil sie Pseudoelemente im Fluss sind, der Einzug, weil er die Überschrift mitnehmen muss. `BootstrapProcessor` sammelt beide jetzt in `outerAttr`, löst `frame-*` bei Containern aus `class` heraus, und das Layout druckt sie neben die ID.

Ausgenommen ist das Accordion: Das Layout rendert es ohne dieses `div`, dort bleibt die Klasse, wo sie war.

16 Behoben: Abstände und der Seitentitel

- **content_margin_top in Spalten-Containern.** Der automatische Abstand galt auch für Inhaltselemente in zwei-, drei-, vier- und sechsspaltigen Containern. Dort regelt die Rinne des Rasters den Abstand; ein Margin darüber schiebt die Spalten gegeneinander aus der Flucht, weil jede Spalte mit einem anderen Element beginnt. `DefaultHelper` nimmt jetzt beide Container-Listen aus.
- **Seitentitel ohne Jumbotron-Container.** `Jumbotron.fluid.html` setzte `pageTitleSkipContainer` unbesehen. Ist für den Jumbotron kein Container eingestellt, hat er auch keinen – und der Seitentitel lief dann über die volle Fensterbreite, während alles andere eingerückt stand. Die Variable wird jetzt nur noch gesetzt, wenn `{jumbotronContainer}` belegt ist.
- **Letzter Absatz in Alerts.** Die Regel `div.alert > *:last-child { margin-bottom: 0 }` erreichte Absätze nicht, die im RTE-Feld `bodytext` stecken. Sie gilt jetzt auch dort.

17 Behoben: Karten-Raster im Backend

Gemeldet als Issue #527, betrifft 5.3.30 bis 5.3.53. Bei `preview = 1` lädt die Extension `bestyles.css` als Backend-Stylesheet. Darin stand eine Regel ohne jede Eingrenzung: `.card-size-small` bekam per `!important` eine feste Breite. Sie traf jedes Modul, das die Karten des Cores benutzt – unter „Hilfe → Über“ standen statt vier nur noch zwei Karten je Reihe, und das `!important` schlug auch `.w-100` fremder Module.

Der Ansatz war dazu von Grund auf falsch: Diese Karten sind Grid-Items, und der Core bemisst sie über `grid-template-columns` am `.card-container`, nicht über `width`. Eine feste Breite ändert die Spaltenaufteilung nicht, sie lässt die Karte aus ihrer Zelle laufen. Die Klasse kommt in der gesamten Extension sonst nirgends vor – die Regel konnte also nur fremde Oberflächen treffen und ist ersatzlos entfernt. Im selben Zug gelten die beiden Regeln für `.settings-category-*` nur noch im Site-Settings-Editor und kommen ohne `!important` aus.

18 Behoben: Offcanvas-Menü

Im Markup des Offcanvas stand eine Klasse namens `undefined`. Sie kommt nicht aus dem HTML, sondern aus `t3sb0ffcanvas()`: Die Funktion setzt zwei Farbklassen, der `ConfigProcessor` liefert nur eine – und `classList.add(undefined)` schreibt die Zeichenkette ins DOM. Sie arbeitet jetzt über die gefilterte Liste und setzt so viele Klassen, wie ankommen.

Dieselbe Datei, zweiter Fund: `t3sbNavbarSearchbox()` griff unterhalb des Breakpoints ohne Prüfung auf das Formular der Navbar zu, während der Zweig darüber brav `if (form)` fragt. Ohne Suchformular brach die Funktion ab.

Panel und Leiste sind dasselbe Markup – unterhalb des Breakpoints ein Panel, darüber die waagerechte Navbar. In CSS war das bisher nicht zu unterscheiden. `t3sb0ffcanvas()` markiert den Zustand jetzt mit `t3sb-offcanvas-panel`. Daran hängen zwei Regeln, die nur im Panel gelten: eine Trennlinie mit Luft über dem Suchformular, und die

Ausrichtungsoptionen der waagerechten Navbar werden neutralisiert – `nav-fill` und `nav-justified` zentrieren dort sonst jede Beschriftung, `ms-auto` schiebt den Block an den Rand.

Das Suchfeld trug in der Variante ohne Schaltfläche ein `me-2` – den Abstand zu einer Schaltfläche, die es dort nicht gibt. Platzhalter und `aria-label` kommen jetzt aus der Sprachdatei.

Dazu entfernt: `Templates/Container/Default.fluid.html` enthielt nichts als ein `<f:debug>` über alle Variablen und wurde von keinem Template referenziert.

19 Checkliste

- Pflicht

- ☐ **Datenbank analysieren** – neun neue Spalten. Sieben in `pages`: `tx_t3sbootstrap_navbar_disable`, `..._meta_disable`, `..._breadcrumb_disable`, `..._pagetitle_disable`, `..._jumbotron_disable`, `..._footer_disable` und `..._indent`. Zwei in `tt_content`: `..._no_indent` und `..._header_vertical`. Ohne die Analyse bleiben der neue Reiter und die neuen Felder leer
- ☐ **Caches leeren** – TCA, Sprachdateien und die kompilierten Fluid-Templates liegen im Cache
- ☐ **Konfigurations-Datensatz einmal speichern** – nur beim Speichern schreibt der Hook `OutsourcedFiles` die Datei `t3sbconstants.typoscript` neu, und erst dann steht `data-navbar-breakpointWidth` im Body

```
vendor/bin/typo3 extension:setup
vendor/bin/typo3 cache:flush
```

- **Nachsehen**

- ☐ **Dropdown-Animation** – wer Slide-In oder Fade eingestellt hatte, sieht sie ab jetzt zum ersten Mal
- ☐ **CDN-Betrieb ohne das Set „OTHER SETTINGS“** – dort lud Bootstrap bisher gar nicht. Prüfen, ob eigene Notbehelfe noch nötig sind
- ☐ **content_margin_top** – wirkt jetzt nicht mehr in Spalten-Containern. Eigene Korrektur-Klassen können weg
- ☐ **Textfarben im RTE** – die sechs Farben stehen jetzt im eigenen Dropdown „Color“ statt in der Styles-Auswahl. Bestehende Inhalte bleiben unverändert, und `rteStyleColors` heißt jetzt `rteColor`
- ☐ **Carousel- und Swiper-Bilder** – ein hinterlegtes Copyright erscheint jetzt im Bild. Eigene Workarounds dafür können raus
- ☐ **Ältere Backend-Layouts** – wer noch `_Extra`-Schlüssel in `pages.backend_layout` hat, sollte den `BackendLayoutUpgradeWizard` laufen lassen; die Seite bricht sonst zwar nicht mehr ab, nutzt aber `OneCol`
- ☐ **Tabellen im RTE** – eine Tabelle mit „Table Striped“ oder „Table Hover“ einmal öffnen und speichern. Die Bootstrap-Klasse steht danach wieder im Markup, auch bei Inhalten, die sie vorher verloren hatten
- ☐ **Rahmenlinien an Wrappern** – wer „Linie darüber/darunter“ an einem Wrapper eingestellt hatte, sieht sie ab jetzt zum ersten Mal. Eigene Notbehelfe über die Extra-Klasse können weg
- ☐ **pageOptOut** – optional. Erst eingeschaltet erscheint der neue Reiter in den Seiteneigenschaften
- ☐ **contentIndent** und **headerVertical** – optional, beide aus. Erst eingeschaltet erscheinen das Einzugsfeld in den Seiteneigenschaften und „Vertical header“ am Inhaltselement. Soll der Einzug nur auf der Seite gelten, auf der er steht, kommt `contentIndentInherit` dazu – aus

✓ **FERTIG!**

Einzug und senkrechte Überschrift sind aus, bis du sie einschaltest – ebenso der neue Reiter. Alles Weitere sind Korrekturen an Stellen, die vorher nicht funktioniert haben; sichtbar wird es nur dort, wo bisher etwas fehlte. Zwei Dinge ändern dabei bestehendes Markup: Eine Rahmenlinie am Wrapper hängt jetzt am äußeren Element, und eine Tabelle im RTE bekommt ihre Bootstrap-Klasse beim nächsten Speichern zurück.