

Update auf Version 5.3.50

Alle generierten Dateien liegen jetzt unter `typo3temp/` –
EXT:t3sb_package wird nicht mehr benötigt. Vom neuen
Ablageort bis zur Checkliste für das Update.

VERSION

5.3.50

TYPO3

^14.3

PHP

8.2 – 8.5

i

Auf einen Blick

Neuer Ablageort	Alles Generierte liegt unter <code>typo3temp/assets/t3sbootstrap/</code> statt im Sitepackage.
t3sb_package entfällt	Die Abhängigkeit ist raus. Auch eine Installation ohne Composer braucht das Paket nicht mehr.
Selbstheilend	Fehlendes TypoScript und SCSS wird beim nächsten Frontend-Aufruf aus der Datenbank neu geschrieben.
Upgrade-Wizard	Vorhandene Dateien werden einmalig an den neuen Ort kopiert. Gelöscht wird nichts.
RTE: Alert-Dropdown	Alle acht Bootstrap-5-Varianten, Wechsel der Variante und Aufheben der Box inklusive.
Konfiguration übertragen	Export und Import der Konfiguration sitzen jetzt direkt im T3SB-Config-Modul.
Neue Einstellung	<code>flexformDir</code> – das Verzeichnis für eigene FlexForm-Erweiterungen ist frei wählbar.

1

Generierte Dateien liegen jetzt in typo3temp

Heruntergeladene CSS/JS, die Bootstrap-Quellen, das custom SCSS und das ausgelagerte TypoScript wurden bisher nach `EXT:t3sb_package/Resources/Public/` geschrieben – also in ein Composer-Paket, das bei jedem `composer update` ersetzt wird. Neuer Ort:

```
typo3temp/assets/t3sbootstrap/  
  TypoScript/      t3sbconstants.typoscript, t3sbsetup.typoscript  
  T3SB-SCSS/       custom-variables-<uid>.scss, custom-<uid>.scss  
  T3SB-CSS/        heruntergeladenes CSS  
  T3SB-JS/         heruntergeladenes JS  
  T3SB-Bootstrap/  Bootstrap-Quellen aus dem Release-Zip  
  css/             kompiliertes CSS
```

TypoScript und SCSS sind Ableitungen aus dem Konfigurations-Datensatz in der Datenbank, das kompilierte CSS ist eine Ableitung davon, und die Downloads holt `t3sbootstrap:cdnToLocal` zurück. Nichts davon ist eine Quelle, alles ist ein Cache – und Caches gehören nach `typo3temp/`.

EXT:t3sb_package wird nicht mehr benötigt

Die Abhängigkeit ist aus `composer.json` und `ext_emconf.php` entfernt. Bestehende Installationen können das Paket weiter installiert lassen; es wird nur nicht mehr beschrieben.

Fehlende Dateien werden selbst wiederhergestellt

Eine neue Middleware schreibt TypoScript und SCSS aus dem Konfigurations-Datensatz neu, sobald sie fehlen – nach einem Deployment, nach „Remove Temporary Assets“ oder auf einer frischen Installation. Sie läuft vor der TypoScript-Auflösung, weil TYPO3 fehlende Dateien aus `includeCSS` / `includeJS` ohne Fehlermeldung verwirft und das Ergebnis in den Page-Cache schreibt.

Upgrade-Wizard

Ein neuer Wizard im Install Tool kopiert vorhandene Dateien aus `EXT:t3sb_package/Resources/Public/` einmalig an den neuen Ort. Er greift auch automatisch beim ersten Frontend-Request, falls das Install Tool nicht geöffnet wird. Gelöscht wird nichts.

⚠ WICHTIG BEIM DEPLOYMENT

Die per `t3sbootstrap:cdnToLocal` geladenen CSS- und JS-Dateien sind die einzigen, die sich **nicht** aus der Datenbank ableiten lassen. Wer `typo3temp/` nicht zwischen Releases übernimmt, muss den Task nach jedem Deployment erneut ausführen. Das Backend-Modul weist darauf hin, wenn lokale Auslieferung konfiguriert ist und die Dateien fehlen.

Neue Einstellung: flexformDir

Bisher las die Extension die erweiterten FlexForm-Definitionen fest aus

`EXT:t3sb_package/Configuration/FlexForms/`. Das Verzeichnis ist jetzt in der Extension Configuration einstellbar, zum Beispiel `EXT:mein_sitepackage/Configuration/FlexForms/`.

Bleibt die Einstellung leer, wird weiterhin das Verzeichnis von `EXT:t3sb_package` verwendet, solange diese Extension installiert ist – bestehende Setups ändern sich also nicht. Ist sie nicht installiert und die Einstellung leer, werden die FlexForms nicht erweitert. Voraussetzung bleibt in allen Fällen, dass `flexformExtend` aktiviert ist.

4 RTE: Alert-Box als Auswahlliste

Die Schaltfläche „Insert Box“ im Rich-Text-Editor ist jetzt ein Dropdown mit allen acht Bootstrap-5-Alert-Varianten:



Die Variante einer bestehenden Box lässt sich nachträglich wechseln, ein Eintrag „Remove Box“ hebt die Box wieder auf und behält den Inhalt.

Ein eigenes Stylesheet für den Editorbereich zeigt die Alert-Farben jetzt auch im Backend – bisher war die Box dort unformatiert, weil das Backend kein Bootstrap lädt.

5 Konfiguration übertragen

Die Konfiguration einer Root-Seite lässt sich jetzt exportieren und wieder importieren – als Aufklappbereich „Konfiguration übertragen“ direkt im T3SB-Config-Modul, unter den ausgelagerten Konstanten. Damit wandert eine fertige Konfiguration von einer Installation in die nächste, ohne Datenbank-Dump.

Der Export schreibt die Konfiguration als JSON-Datei – ohne uid und ohne pid, Dateireferenzen als kombinierte Identifier. Der Seitenbaum der Zielinstallation darf also völlig anders aussehen.

IMPORT-MODUS	VERHALTEN
Aktualisieren Voreinstellung	Der vorhandene Datensatz wird überschrieben und behält seine uid. Felder, die der Import nicht enthält, werden auf ihren Standardwert zurückgesetzt – das Ergebnis entspricht also exakt dem Export. Liegen mehrere Konfigurationen auf der Zielseite, wird die wirksame aktualisiert und die übrigen werden entfernt.
Ersetzen	Der vorhandene Datensatz wird gelöscht und ein neuer angelegt. Die uid ändert sich, der alte Datensatz bleibt als gelöschte Zeile zurück.
Hinzufügen	Nur möglich, wenn die Zielseite noch keine Konfiguration hat – sonst bricht der Import mit einer Meldung ab.

Nicht im Export enthalten sind die Site Settings – Bootstrap-Version, Bootswatch-Theme und Google Fonts stehen in der Site-Konfiguration und wandern mit dem Repository, nicht mit der JSON-Datei.

- Nach dem Update

☐ **Caches leeren** – mehrere Konstruktor-Signaturen haben sich geändert, der kompilierte DI-Container muss neu gebaut werden

☐ **Datenbank-Update** anstoßen – `tx_t3sbootstrap_list_item_inline` bekommt eine Spalte `hidden`

☐ Upgrade-Wizard im Install Tool ausführen

- Nach der Erstinstallation

☐ T3SB Config auf der Root-Page anlegen und speichern

☐ `t3sbootstrap:cdnToLocal` – hinzufügen und einmalig ausführen

☐ In den Site Settings *Enable CDN* deaktivieren

☐ Nur mit eigenem SCSS: *Enable Custom SCSS* aktivieren, dann `t3sbootstrap:customScss` mit der Root page ID ausführen

✓ **FERTIG!**

Wenn das Frontend sauber rendert, ist das Update abgeschlossen. TypeScript und custom SCSS holt sich t3sbootstrap ab jetzt selbst aus der Datenbank zurück – nur die CDN-Downloads brauchen den Task.