

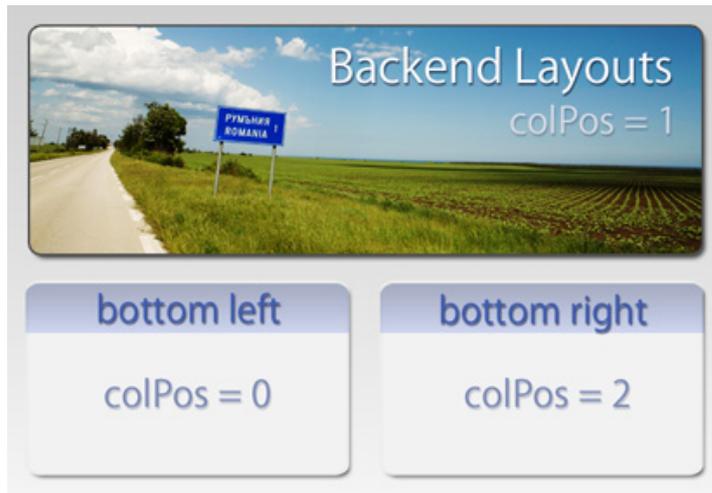
undkonsortenBlog

der Berliner Internet Agentur zu TYPO3, WordPress, SEO



TYPO3-Tutorial: Backend Layout nutzen (TYPO3 4.5)

6. Februar 2011 | 27 Kommentare



Einen Homepage mit einem breiten Teaserbilder oder einer Slideshow oben, darunter ein zwei- oder dreispaltiges Layout – dass kann man jetzt im TYPO3-Backend (ohne TemplaVoilà) einfach darstellen. Denn seit der **TYPO3-Verion 4.5**. läßt sich jetzt jeder Seite die Eigenschaft **Backend Layout** zuweisen.

Wir wollen in unserem TYPO3-Tutorial erklären, wie man diese Backend-Layouts am besten verwendet.

Das Backend-Layout im TYPO3-Backend einrichten

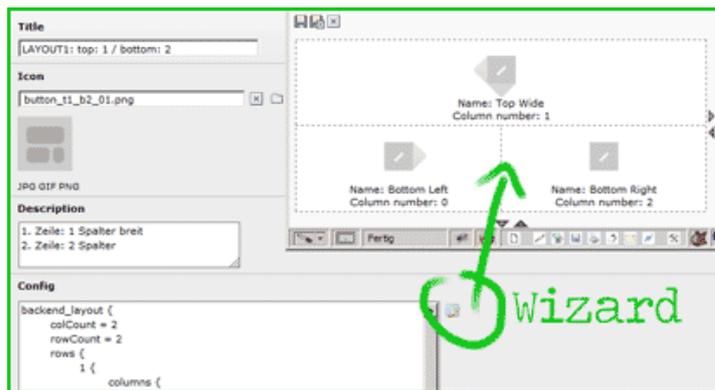
Zuerst einmal müssten wir dazu zwei Backend-Layouts erstellen. Wir gehen in unserem Beispiel von folgenden zwei Varianten aus:



- **LAYOUT 1:**
Erste Zeile Einspalter breit / Zweite Zeile Zweispalter 50-50
- **LAYOUT 2:**
Zweispalter 50/50 (keine erste Zeile)

Dazu legen wir einen Sys-Folder an und legen hier einen neuen

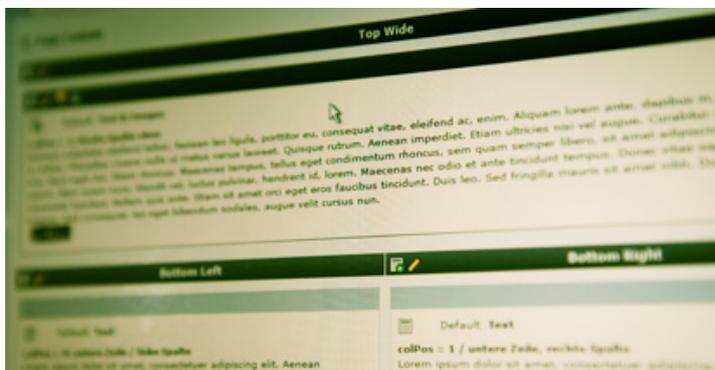
Datensatz vom Typ *Backend Layout* an. Der Wizard hilft dabei, die Config-Datei zu erstellen.



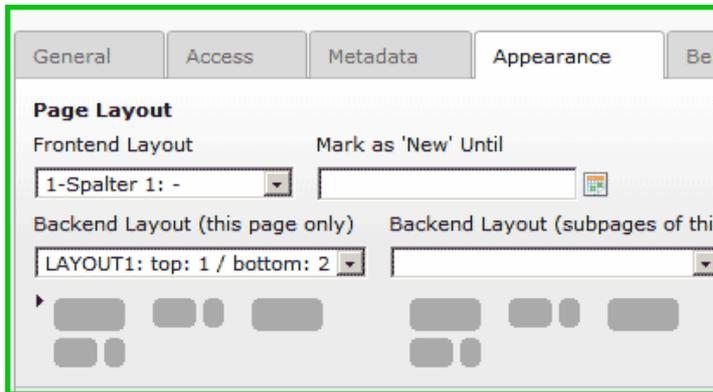
Die sich daraus ergebende *Backend Layout* > *Config* lautet für das Layout 1:

```
backend_layout {
  colCount = 2
  rowCount = 2
  rows {
    1 {
      columns {
        1 {
          name = Top Wide
          colspan = 2
          colPos = 1
        }
      }
    }
    2 {
      columns {
        1 {
          name = Bottom Left
          colPos = 0
        }
        2 {
          name = Bottom Right
          colPos = 2
        }
      }
    }
  }
}
```

Das muss man jetzt noch analog für das Layout 2 wiederholen, wobei man nur die Angaben für die beiden unteren Zeilen braucht.



Nun stehen uns in den Seiteneigenschaften die gerade angelegten *Backend Layouts* unter dem Reiter *Appearance* > *Backend Layout (this page only)* zur Verfügung.



Wir sehen hier auch noch ein Relikt aus alten Zeiten, das *Frontend Layout*, mit dem wir früher versucht haben, unseren Seiten optisches Benehmen beizubringen. Im Prinzip können wir mit der hier vorgestellten Methode auf das gute alte Feld *Frontend Layout* erstmal komplett verzichten.

Die Eigenschaften unseres Seiten-Layout – im Frontend wie im Backend – legen wir jetzt über das Feld *Backend Layout* fest.

Das Backend-Layout für die TYPO3 Frontend-Ausgabe einrichten

So, machen wir uns ans TypoScript für die Frontend-Ausgabe. Was jetzt kommt, hängt natürlich ein bisschen davon ab, wie Ihr Eure Seiten baut. Aber schauen wir uns nochmal das Bild ganz am Anfang des Artikels an und überlegen wir uns was wir an HTML-Elementen brauchen.

Für den *BODY* bietet es sich an, verschiedene Klassen, je nach Layout zu vergeben. Weiterhin umhüllen wir alle drei Content-Boxen mit einem *DIV* der Klassen *col* und dann *col1*, *col2*, *col3*.

```
<body class="x1-1 t1 b1">
  <div class="col col1">Content Oben</div>
  <div class="col col2">Content Unten Links</div>
  <div class="col col3">Content Unten Rechts</div>
</body>
```

Kümmern wir uns erst mal um die Ausgabe **Body-Tag-Klasse**. Das geht z.B. so (TypoScript Setup):

```
# Klassen für BODY-Tag vorbereiten
#####
temp.layoutSwitch = CASE
temp.layoutSwitch {
  key.data = levelfield:-1,backend_layout_next_level,slide
  key.override.field = backend_layout
  1 = TEXT
  1.value = x1-1 t1 b1
  2 = TEXT
  2.value = x1-1 t0 b1
  3 = TEXT
  3.value = t1 b0
}
# BODY-Tag erstellen
#####
temp.bodyTag = COA
temp.bodyTag {
  10 = TEXT
  10.dataWrap = <body id="{TSFE:id}" class="

  # === insert layout switch (page|backend_layout)
  20 < temp.layoutSwitch
  90 = TEXT
  90.value = ">
}
</body>
```

Bleibt zuletzt, col1, col2 und col3 mit den richtigen Inhalten zu füllen.
Schauen wir uns dazu nochmal die Spalten und die dazugehörigen
`pages > colPos`-Angaben an.



Das erreichen wir mit den folgenden drei Blöcken TypoScript:

```
# COL0 (breite Spalte oben)
page.1000 = COA
page.1000 {
    wrap = <div class="col0 col">|</div>
    # where colPos = 1
    10 < styles.content.get

    # === enable col0 according to page layout
    stdWrap.if {
        value = 1,2
        isInList.data = levelfield:-1,backend_layout_next_level
        isInList.override.data = TSFE:page|backend_layout
    }
}
# COL1 (linke Spalte unten)
page.2000 = COA
page.2000 {
    wrap = <div class="col1 col"> | /div>
    # where colPos = 0
    10 < styles.content.getLeft

    # === enable col1 according to page layout
    stdWrap.if {
        value = 1
        isInList.data = levelfield:-1,backend_layout_next_level
        isInList.override.data = TSFE:page|backend_layout
    }
}
#COL2 (rechte Spalte unten)
page.3000 = COA
page.3000 {
    wrap = <div class="col2 col"> | /div>
    # where colPos = 2
    10 < styles.content.getRight

    # === enable col2 according to page layout
    stdWrap.if {
        value = 1,2
        isInList.data = levelfield:-1,backend_layout_next_level
        isInList.override.data = TSFE:page|backend_layout
    }
}
}
```

(Ich weiß unterm im Code gehen im wrap die DIVs nicht zu – aber liegt an dem mächtigen WordPress-Rich-Text-Editor!)

Damit der Slider für das `backend_layout_next_level` funktioniert müssen wir noch über das Install-Tool folgenden Wert eintragen:

```
$TYPO3_CONF_VARS['FE']['addRootLineFields'] = 'backend_layout_ne
```

So, das sollte eigentlich genügen. Jetzt müsst Ihr nur noch die drei eben erzeugten COLs in Euer PAGE-Objekt kopieren und abhängig von den Einstellungen in *Backend Layout* bzw. *Backend-Layout nächste Ebene* habt Ihr eine modifizierte Frontend-Ausgabe

Dank an Felix für die Inspiration und Lösungsansätze!

Noch Fragen, Inspirationen oder Verbesserungen?

Ähnliche Artikel:

- [Selects bei TYPO3 als Baum darstellen](#)
- [TYPO3: Sprachumschaltung mit Language Selection \(sr_language_menu\)](#)
- [Update: TYPO3-Newsletter mit Direct Mail und Direct Mail Subscription](#)
- [Drucken mit TYPO3](#)
- [TYPO3-Tutorial: RealURL verstehen](#)
- [TYPO3-Tutorial: Faxnummer zu lang? Vordefinierte TCA-Felder ändern](#)
- [TYPO3-Tutorial: FLV Video Gallery – Textplaylist ohne Thumbnails](#)

Meta-Daten

Autor: undkonsorten, Thomas Alboth

Tags+Kategorien: 4.5, Backend-Layout, TYPO3, TYPO3-Tutorial, + CMS, Tutorial, TYPO3

4



27 Kommentare

Von: [t3n-Linktips: Internetzensur, Super Bowl Ads, Android 2.4, TYPO3 4.5 Tutorial und WordPress als Brettspiel » t3n News](#) **1**
7. Februar 2011, 16:52 Uhr

[...] Die neue Backend Layout Eigenschaft von TYPO3 4.5 macht es möglich, eine Homepage oben mit einem breiten Teaserbild auszustatten und direkt darunter ein zwei- oder dreispaltiges Layout zu platzieren. Und das funktioniert auch ohne TemplaVoilà, wie dieses Tutorial zeigt. [...]

Von: [Tweets that mention TYPO3-Tutorial: Backend Layout nutzen \(TYPO3 4.5\) / TYPO3-Agentur Berlin: undkonsorten: -- Topsy.com](#) **2**
8. Februar 2011, 22:32 Uhr

[...] This post was mentioned on Twitter by TYPO3 Agentur Essen and Wolfgang Wagner, Attila Janos. Attila Janos said: RT @wowawebdesign: TYPO3-Tutorial: Backend Layout nutzen (TYPO3 4.5) / TYPO3-Agentur Berlin: undkonsorten: <http://goo.gl/RuwuK> [...]

Von: [Stefano](#) **3**
8. Februar 2011, 23:48 Uhr

Da hat sich ein Fehler im letzten TS Codebeispiel eingeschlichen: In den Grafiken ist die breite Spalte (top width) stets colPos1 in dem TS jedoch ColPos0 und die linke Spalte ist colPos1.

Von: [alBoert](#) **4**
9. Februar 2011, 00:03 Uhr

Hallo zusammen,

vielen Dank für dieses super Tutorial.

Für meine letztmalig erstellte Typo3-Seite habe ich mit TemplaVoila gearbeitet. Ich muss zudem sagen, dass das schon vor einiger Zeit geschehen ist und ich mich erst wieder in Typo3 etwas einarbeiten muss...

Besteht irgendwie die Möglichkeit dieses Backend-Layout in Verbindung mit einem HTML-Template, das über Subparts gefüllt wird, zu nutzen?

Ich würde ungern mein Layout direkt in Typo3 zusammenbauen (mit verschachtelten wraps usw.), sondern einmal als Vorlage haben – schon allein um sie dem Kunden präsentieren zu können – und dann einbinden.

Tut mir leid, wenn diese Frage vielleicht banal klingt. Sooo die Erfahrung im Umgang mit Typoscript hab ich leider nicht, ging mit TemplaVoila ja relativ einfach auch so.

Danke schonmal im voraus! 😊

Von: Norman

5

9. Februar 2011, 03:54 Uhr

Alles schön und gut, aber mit TemplaVoila hab ich da weniger Ärger und muss mich mit nicht soviel TS rumschlagen. Trotzdem ein nützliches Tutorial, danke dafür. 😊

Von: undkonsorten, Thomas Alboth

6

9. Februar 2011, 23:34 Uhr

@Stefano: Danke. Fehler behoben!

Kann ich den Kommentar löschen – sonst fangen leute später noch an, den Fehler zu suchen?

Von: undkonsorten, Thomas Alboth

7

9. Februar 2011, 23:44 Uhr

@alBoert:

FRAGE: Besteht irgendwie die Möglichkeit dieses Backend-Layout in Verbindung mit einem HTML-Template, das über Subparts gefüllt wird, zu nutzen?

ANTWORT

Klar geht das.

1: Entweder du machst dein einzubindendes Template-File vom gewählten Backend-Layout abhängig. An der Stelle, wo wir jetzt verschiedene Body-Klassen vergeben, müsstest Du halt abhängig vom Feld `backend_layout` die verschiedenen `temlate.files` einbinden.

2: Oder Du musst halt anfangen in deinen Templates später einen großen Bereich mit dem Marker `###CONTENT###` (und nicht mehr `###CONTENT-LEFT###`, `###CONTENT-RIGHT###` etc. zu nutzen. Dann machst Du aus dem was bei uns `page.1000 = COA` heiß ein `temp.1000 = COA`.

Und den Marker `###CONTENT###` füllst du mit etwas wie

```
page.10.marks.CONTENT = COA
page.10.marks.CONTENT.10 < temp.1000
page.10.marks.CONTENT.11 < temp.2000
```

Sollte auch funktionieren.

Von: *undkonsorten, Thomas AlbothNo*

8

9. Februar 2011, 23:49 Uhr

@Norman: Wir können bei TemplaVoila einfach nur diese Vermischung aus Datenbank+XML-Struktur nicht ab. Das kommt so übergestülpt daher.

Schwer, da mal mit einer SQL-Abfrage was umzuhängen oder Content in nicht Templa-Seiten zu migrieren.

Aber das ist wohl Geschmackssache. Wir können auch mit Flash nicht und andere finden es toll.

Von: *Henrik Ziegenhain*

9

10. Februar 2011, 09:28 Uhr

vielen Dank für das wirklich ausführliche Tutorial zum genialen Grid View.

Wie weiße ich denn den neuen Bodytag meiner Seite zu? page.bodyTag akzeptiert nur Tags, wenn ich dem mein temp.bodyTag zuweise erhalte ich statt dem kompletten Bodytag nur ein COA. Den W3C-Validator freut das sicher nicht 😊

Von: *Henrik Ziegenhain*

10

10. Februar 2011, 09:40 Uhr

page.bodyTag >
page.bodyTagCObject < temp.bodyTag

manchmal muss man einfach erst einen Kommentar posten um direkt im Anschluss selbst über die Lösung zu stolpern 😊

Danke nochmals

Von: *Heike*

11

10. Februar 2011, 18:53 Uhr

Hallo,

völlig begeistert von dem Tipp habe ich heute meine Typo3-Baustelle von Version 4.4.6 auf 4.5 gebracht und ein paar solcher Layouts angelegt.

Ich habe Bilder eingebunden, Namen vergeben, aber ich sehe diese Option weder beim Ändern von vorhandenem Inhalt, noch beim Anlegen neuer Seiteninhalte.

Muss ich das noch irgendwo aktivieren?

Gruß
Heike

Von: *Heike*

12

11. Februar 2011, 14:20 Uhr

Bitte den letzten Kommentar löschen.
War etwas anderes.
Die Lösung läuft und ist genial.
Genau das, was ich gesucht habe.

Danke an die Entwickler 😊)

Heike

Von: *undkonsorten, Jochen Biedermann*

13

14. Februar 2011, 11:58 Uhr

ist weg 😊

Von: *Heike*

14

14. Februar 2011, 12:20 Uhr

Dankeschön 😊

Ein Frage habe ich noch...

Ich kann in meinem Backend zwar die Layouts auswählen, allerdings verweigert mir mein Typo3 diese praktische Anzeige im Backend, wenn ich Inhalte einfüge, dass die entsprechenden Bereiche an der passenden Stelle sichtbar sind...

Gibt es dafür einen Trick?

Brauche ich hierfür noch eine andere Einstellung?

Wenn ich einen neuen Inhalt anlege muss ich noch immer wählen, in welche Spalte er soll. Ich kann dann zwar die Position auswählen, die ich im Layout definiert habe, aber irgendwie habe ich das Gefühl, dass mir da etwas noch nicht rund läuft...

Gruß

Heike

Von: *sun*

15

17. Februar 2011, 22:54 Uhr

Danke für das Tut und Danke an die Entwickler. Endlich können die Redakteure im Backend das Layout und die Spalten sehen, wie es im Frontend angezeigt wird – und das sogar mit Backend-Icons.

Ich arbeite auch lieber mit einem HTML-Template. Dazu hier ein Beispiel, wie man das Template-TS-Setup inklusive levelfield/slide beeinflussen kann:

```
page.10 = TEMPLATE
page.10 {
  template = FILE
  template.file = fileadmin/tmpl/html/index.html
  workOnSubpart.cObject = CASE
  workOnSubpart.cObject {
    key.data = levelfield:-1, backend_layout_next_level, slide
    key.override.field = backend_layout
    default = TEXT
    default.value = LAYOUT1
    2 = TEXT
    2.value = LAYOUT2
    3 = TEXT
    3.value = LAYOUT3
  }
}
```

Das könnte man auch direkt auf "page.10.template" anwenden
(Hauptsache man hat ein cObject für das CASE):

```
page.10 = TEMPLATE
page.10 {
template.cObject = CASE
template.cObject {
key.data = levelfield:-1, backend_layout_next_level, slide
key.override.field = backend_layout
default = TEXT
default.value = fileadmin/tmpl/html/layout0.html
2 = TEXT
2.value = fileadmin/tmpl/html/layout1.html
3 = TEXT
3.value = fileadmin/tmpl/html/layout2.html
}
workOnSubpart = DOCUMENT
}
```

Ohne levelfield/slide geht es auch wesentlich einfacher:

```
page.10 = TEMPLATE
page.10.template = FILE
page.10.template.file = fileadmin/tmpl/html/index.html
page.10.workOnSubpart.cObject = TEXT
page.10.workOnSubpart.cObject.dataWrap =
LAYOUT{field:backend_layout}
```

Oder:

```
page.10 = TEMPLATE
page.10.template = FILE
page.10.template.file = fileadmin/tmpl/html/index.html
page.10.workOnSubpart = LAYOUT1
```

```
[globalVar = TSFE:page|backend_layout = 2]
page.10.workOnSubpart = LAYOUT2
[globalVar = TSFE:page|backend_layout = 3]
page.10.workOnSubpart = LAYOUT3
[global]
```

(Weiß wer, wie man in eine Condition "levelfield:-1, _field_,slide"
unterbringt? Geht das irgendwie? Nundenn, ist ein anderes Thema.)

Von: Anonymous **16**
24. Februar 2011, 16:37 Uhr

[...] [...]

Von: Thomas **17**
2. März 2011, 12:07 Uhr

Danke für das Tutorial. Echt genial! Da ich leider Anfänger
bin komm ich mit dem letzten Zusatz:

"Jetzt müsst Ihr nur noch die drei eben erzeugten COLs in Euer
PAGE-Objekt kopieren"

nicht klar. Das ist bestimmt ne total blöde Frage, aber die Antwort
würde mich weiterbringen. Vielen Dank!

VG Thomas

Hallo

Aufgrund euer Informationen habe ich mal ein Template mit Typoscript für Fluid gebaut.

Hier könnt Ihr ein Rahmenlayout und ein subTemplate verwenden!
Das Rahmenlayout ist immer gleich und die Subtemplates sind mit dem neuen Backendlayout verknüpft.

```
lib.subTemplateVars {
  lorem = TEXT
  lorem.value = ipsum

  content < styles.content.get
}

lib.subTemplate1 = FLUIDTEMPLATE
lib.subTemplate1 {
  file = {$plugin.tx_dftemplate.resPathPrivate}Templates/Page/Standard.html
  variables < lib.subTemplateVars
}
lib.subTemplate2 = FLUIDTEMPLATE
lib.subTemplate2 {
  file = {$plugin.tx_dftemplate.resPathPrivate}Templates/Page/Margin.html
  variables < lib.subTemplateVars
}

lib.layoutTemplate = FLUIDTEMPLATE
lib.layoutTemplate {
  file = Layout.html
  file.wrap = {$plugin.tx_dftemplate.resPathPrivate}Templates/Page/ |
  variables {
    title = TEXT
    title.value = Some Title
    imagePath = TEXT
    imagePath.value = {$plugin.tx_dftemplate.resPathPublic}Images/
    navTop < lib.navTop
    address = TEXT
    address.value = xxxxxx
    copyright = TEXT
    copyright.value = © xxx xxxx 2011

    subTemplate = CASE
    subTemplate {
      key.data = levelfield:-1,backend_layout_next_level,slide
      key.override.field = backend_layout
      1 < lib.subTemplate1
      2 < lib.subTemplate2
    }
  }
}
```

Das neue BE-Layout gefällt mir super. Das mit FLUIDTEMPLATE von KimPixel umso mehr. Allerdings hat das Ganze ein Nachteil:

Eine mehrsprachige One-Tree-Lösung

Ich hatte in relativ kurzer Zeit 4 verschiedene BE-Layouts nach Designvorlage Einsatzbereit. Mir fiel auf, dass unter dem PAGE-Modul in der Language-Ansicht zwar die Aufteilung so angezeigt werden, wie im BE-Layout eingerichtet war.

Nur hat der Redakteur hier immer noch nur die Möglichkeit alle Default-Inhaltselemente auf einmal in die andere Sprache zu kopieren und zu bearbeiten. An dieser Stelle bietet TV einfach eine komfortablere Lösung.

Ich muss mich Aktuell entscheiden, welchen Ansatz ich meinem Kunden anbieten soll. TV mit 4 verschiedenen Template-Objekten und einfachere mehrsprachige Verwaltung oder einfache Template-Verwaltung und etwas "kompleziertere" mehrsprachige Verwaltung.

Was meint ihr?

Von: *undkonsorten, Jochen Biedermann*

20

31. März 2011, 16:29 Uhr

Es ist ja kein Geheimnis, dass wir undkonsorten keine großen Fans von TV sind und es aus unserer Sicht nur bei wenigen Einsatzzwecken wirklich Sinn ergibt.

Vielleicht verstehe ich daher grade auch nicht wirklich, wo TV aus Deiner Sicht bei einem One-Tree-Baum genau Vorteile bringt...

Von: *Urs Bräm*

21

17. April 2011, 05:59 Uhr

Eine Frage zu Kommentar 15 von sun: Die Lösung mit dem Template ist super.

Muss man aber zusätzlich zur bedingten Wahl des Templates auch noch die Zuweisung der Inhalte zu den Markern mit `marks.INHALTSMARKER < styles.content.get` an die Gegebenheiten des Templates anpassen?

D.h. hier nochmal analog mit `if` oder `case` arbeiten?

Oder reicht es, wenn in einem allgemeinen Marks-Block wie

```
marks {
  COL0 < styles.content.get
  COL0.select.where = colPos = 0
  COL1 < styles.content.get
  COL1.select.where = colPos = 1
  COL2 < styles.content.get
  COL2.select.where = colPos = 2
  (etc...)
}
```

einfach alle Marker befüllt werden, wobei dann halt nicht alle in dem jeweiligen Template vorkommen?

Oder ist das ein Performancebremser, weil unnötiger Inhalt ausgelesen, aber nicht zugewiesen wird?

Danke!

Urs

Von: *Urs Bräm*

22

17. April 2011, 06:33 Uhr

Noch zum Post von Dennis (Nr. 19)

“Nur hat der Redakteur hier immer noch nur die Möglichkeit alle Default-Inhaltselemente auf einmal in die andere Sprache zu kopieren und zu bearbeiten. “

Könnte es sein, dass du
defLangBinding = 1
nicht einsetzt?

Von: Urs Bräm

23

26. April 2011, 08:03 Uhr

Ich hab's für mich gelöst (denke ich; vgl. Post 21) und poste hier mal meine Anwendung. Diese Funktion ist einfach fantastisch.

```
page.10 = TEMPLATE
page.10 {
// bedingung
stdWrap.if {
value = 1
isInList.data = levelfield:-1,backend_layout_next_level,slide
isInList.override.data = TSFE:page|backend_layout
}
template = FILE
template.file = fileadmin/templates/main/tmpl/start.html
workOnSubpart = DOCUMENT_BODY
marks {
LOGO = TEXT
LOGO.value = Schickeslogo
COL0 < styles.content.get
COL0.select.where = colPos = 0
COL1 < styles.content.get
COL1.select.where = colPos = 1
COL2 < styles.content.get
COL2.select.where = colPos = 2
}
}
```

```
page.20 = TEMPLATE
page.20 {
// bedingung
stdWrap.if {
value = 2
isInList.data = levelfield:-1,backend_layout_next_level,slide
isInList.override.data = TSFE:page|backend_layout
}
template = FILE
template.file = fileadmin/templates/main/tmpl/innen.html
workOnSubpart = DOCUMENT_BODY
marks {
COL0 < styles.content.get
COL0.select.where = colPos = 0
COL1 < styles.content.get
COL1.select.where = colPos = 1
}
}
```

Von: Christian

24

27. April 2011, 23:10 Uhr

Für die Einbindung im TypoSript Template habe ich noch einen etwas anderen Ansatz (eigentlich eine Mischung aus den hier schon aufgeführten):

zunächst definiert man die Inhalte, abhängig des Layouts, z.B.:

```
#1 col
temp.1col = COA
temp.1col {
10 < styles.content.get
wrap = |
}

#2 cols
temp.2cols = COA
temp.2cols {
10 < styles.content.get
10.wrap = |
20 < styles.content.getRight
20.wrap = |
wrap = |
}
```

dann wählt man mit einer Bedingung aus, welches dieser Konstrukte eingebunden werden soll:

```
page {

#1 or 2 cols
10 = CASE
10 {
key.data = levelfield:-1,backend_layout_next_level,slide
key.override.field = backend_layout
1 < temp.1col
2 < temp.2cols
}

}
```

Von: *Dennis*

25

28. April 2011, 09:39 Uhr

Ich habe es so wie oben beschrieben eingebunden. Es funktioniert(e) alles einwandfrei. Bis jetzt geht auch noch alles, nur seitdem ich wohl irgendetwas eingestellt habe, sei es im TS-Setup oder PageTS, kann ich unter Appearance keine meiner 5 angelegten BE-Layouts im gesamten Seitenbaum mehr auswählen. Stattdessen erscheint: [INVALID VALUE("ID-Nummer")]

Seitenbaum:

- > Startseite (Standard) – Hier erscheint schon der Fehler
- > Backend-Layouts (Sys-Folder)

Hat jemand diese Erfahrung gemacht und könnte hier eine Lösung posten?

Von: *Tobias*

26

19. Mai 2011, 07:52 Uhr

Ich habe wie oben beschrieben das Template zusammengestellt. Nur wird das Template vom Typo3 nicht gefunden. Das Drop-Down Menu

für Backend Layouts (in den Seiteneinstellungen) ist leer und enthält nur den Eintrag "none". Hab ich was übersehen?

Von: Urs Bräm

27

19. Mai 2011, 10:40 Uhr

@Tobias (Post 26) hast du in den Seiteneinstellungen bei "Allgemeine Datensatzsammlung" die Seite angegeben, die den Layout-Datensatz enthält?

Von: Julian

28

Achtung: Der Kommentar muss erst noch freigegeben werden.

30. Juni 2011, 08:43 Uhr

Vielen Dank für den guten Artikel!

Hier sieht man mal samt TS und Screens, wie die Backend-Layouts einsetzbar sind – und dass man sich diese vielleicht doch endlich einmal genauer anschauen sollte... :-\

Auch mal Kommentieren:

Name (erforderlich)

eMail (erforderlich) (wird nicht veröffentlicht)

Notify me of followup comments via e-mail

Webseite

Kommentar