

TYPO3-Tutorial

„Umwandeln von bestehenden Inhaltselementen in Pseudo- „FCE“ mittels TCEFORM“

Inhaltsverzeichnis

TYPO3-Tutorial

„Umwandeln von bestehenden Inhaltselementen in Pseudo- „FCE“ mittels TCEFORM“.....	1
1. Schritt „Anlegen des SYS-Ordner zur Datenablage“.....	1
2. Schritt „Hinterlegen der TS-Config“.....	1
3. Schritt: „Anlegen des Inhaltselements für den Ansprechpartner“.....	2
4. Schritt: „Die Konfiguration im TypoScript SETUP“.....	4
5. Schritt: „Verknüpfen über das Inhaltselement Datensatz Einfügen“.....	6

In diesem Tutorial werden wir beschreiben wie man mit „TYPO3 Bordmitteln“ mit Hilfe von TCEFORM ein bestehendes Inhaltselement vom Typ Text mit Bild abwandelt für einen bestimmten Einsatzzweck.

Folgende Projektstellung:

Wir haben bei einem Webprojekt die Aufgabe als Entwickler dafür zu sorgen, das die Redakteure in einem Bereich der Website immer wieder Kontaktdaten in Form von Anschrift, E-Mail, Telefon, Anschrift in Verbindung mit einem Passbild zu hinterlegen. Die Kontaktdaten können sich auf den einzelnen Unterseiten des Webauftritts auch wiederholen.

Die Pflege und Aktualisierung soll so einfach wie möglich bedienbar sein, da sich die Ansprechpartner desöfteren ändern können. Außerdem sollen die Bilder für die Passfotos durch die Redakteure einfach hochgeladen werden können, ohne große Bearbeitungsschritte mit einem Grafikprogramm zu machen.

Der Einsatz von TemplaVoila ist nicht gewünscht, ebenso wenig wie das Entwickeln einer maßgeschneiderten Extension für diese partiellen Seitenview für dieses Projekt, aufgrund von Budgetbeschränkungen.

Ein Lösungsansatz:

1. Schritt „Anlegen des SYS-Ordner zur Datenablage“

Wir legen dazu einen neuen SysOrdner im Seitenbaum an welchen wir als zentralen Ablageort für die Datensätze vom Typ „Text mit Bild“ benutzen. Wie man bereits oben herauslesen konnte empfiehlt es sich nicht die Datensätze mehrfach anzulegen, da ansonsten bei Änderung der Daten bei nur einem Ansprechpartner dies mit einem erhöhten Aufwand für die Redakteure verbunden wäre.

Daher benutzen wir den SysOrdner als zentralen Ablageort für die Datensätze.

2. Schritt „Hinterlegen der TS-Config“

Wir hinterlegen nun die für unsere Zwecke bereits vorbereitete TCEFORM Einstellung für diesen Sysordner. Vereinfacht gesagt: Wir blenden alle für unseren Zweck nicht benötigten Eingabefelder aus. Desweiteren benennen wir einige Felder entsprechend um, für den hiesigen Einsatzzweck.

```
TCEFORM.tt_content {
```

```

    imageorient.disabled = 1
    image_compression.disabled = 1
    imagewidth.disabled = 1
    imageheight.disabled = 1
    longdescURL.disabled = 1
    altText.disabled = 1
    titleText.disabled = 1
    header_link.disabled = 1
    bodytext.disabled = 1
    header_position.disabled = 1
    header_layout.disabled = 0
    date.disabled = 1
    image_effects.disabled = 1
    imagecaption_position.disabled = 1
    image_noRows.disabled = 1
    imagecols.disabled = 1
    imageborder.disabled = 1
    image_zoom.disabled = 1
    image_link.disabled = 1
    rte_enabled.disabled = 1
    header.label = Überschrift für den Ansprechpartner
    imagecaption.label = Bildbeschreibung für Ansprechpartner
    header_layout.disabled = 1
}

```

Somit haben wir letztendlich nur noch zwei Eingabefelder sichtbar.

3. Schritt: „Anlegen des Inhaltselements für den Ansprechpartner“

Wir legen nun entsprechend ein Inhaltselement vom **Typ Text mit Bild** an, in unserem neu angelegten SysOrdner. Es ist egal in welcher Spalte wir das Inhaltselement letztendlich hinterlegen, die passende Zuordnung im Layout erfolgt später über eine andere Methode. Wir überspringen hier die einzelnen Schritte für das Anlegen eines Inhaltselements.

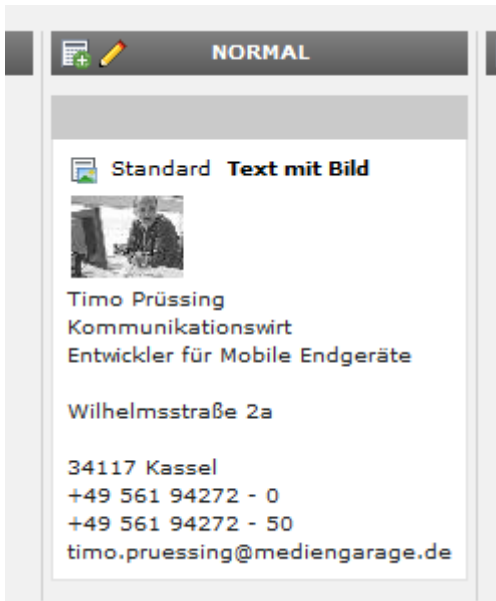


Abbildung 1: Exemplarisches Inhaltselement

Haben wir die TS-Confog Einstellungen von Schritt 2 richtig hinterlegt, besteht nun nach erfolgreichen Anlegen die **Registerkarte Medien** des Inhaltselements nur noch aus 2 relevanten Eingabefeldern. Genau passend für unseren Einsatzzweck. Die **Registerkarten Allgemein und Zugriff** brauchen uns nicht zu interessieren für dieses Tutorial.

Das nachfolgende Bild sollte bereits alles entsprechend zeigen, was für die weiteren Schritte relevant sein sollte.



Abbildung 2: Die beiden relevanten Eingabefelder

4. Schritt: „Die Konfiguration im TypoScript SETUP“

Wir benötigen nun für die Einbindung und Verarbeitung der eingegebenen Daten noch eine Erweiterung unseres TypoScriptes. Wir binden später die einzelnen Datensätze in unseren SysOrdner mittels des **Inhaltselement „Datensatz einfügen“** an den jeweiligen Stellen im Frontend ein.

Dafür müssen noch einige Vorbereitungen getroffen werden.

Daher erweitern wird den Typoscript Code des tt_content.shortcut Elements, welchen wir aus dem css_styled_content Template der Einfachheit halber komplett übernehmen in unser TypoScript Template und erweitern dies wie folgt.

```
tt_content.shortcut = COA
tt_content.shortcut {
    20 = CASE
    20.key.field = layout
    20.0 = RECORDS
    20.0 {
        source.field = records
        tables = {$content.shortcut.tables}
        conf.tt_content = < tt_content
        conf.tt_address = < tt_address
        conf.tt_links = < tt_links
        conf.tt_guest = < tt_guest
        conf.tt_board = < tt_board
        conf.tt_calender = < tt_calender
        conf.tt_rating < tt_rating
        conf.tt_products = < tt_products
        conf.tt_news = < tt_news
        conf.tt_poll = < plugin.tt_poll
    }
    20.1= RECORDS
    20.1 {
        source.field = records
        tables = {$content.shortcut.tables}
        conf.tt_poll = < plugin.tt_poll
        conf.tt_poll.code = RESULT,SUBMITTEDVOTE
    }
}
##### ####

#### Neues TS-Script für die Ansprechpartner ####
20.10 = RECORDS
20.10 {
    source.field = records
    tables = {$content.shortcut.tables}

    stdWrap.outerWrap = <div class="contactperson">|</div>

    conf.tt_content = COA
    conf.tt_content {
        15 = TEXT
        15.field = header
        15.required = 1
        15.wrap = <h2>|</h2>

        ### Beschriftungstext für das Bild ###
        30 = IMAGE
        30.stdWrap.outerWrap = <div class="consultantcontainer">|</div>
        30.stdWrap.append = COA
        30.stdWrap.append {

            10 = TEXT
            10.parseFunc < lib.AdvancedCustomTags
            10 {
```

```

        field = imagecaption

    split {
        token.char = 10
        cObjNum = 1||2||3|*|
        ### Überschrift ###
        1 = TEXT
        1.current = 1
        1.wrap = <h3>|</h3>

        2 = TEXT
        2.current = 1
        2.wrap = <h6>|</h6>

        3 = TEXT
        3.current = 1
        3.wrap = |

    }

}

### einlesen des Bildes
30.params = class="teaserimage"
30.file {
    params = class="teaserimage"

    import {
        cObject = TEXT
        cObject {
            value = fileadmin/placeholder.gif

            ### Wenn Bild gesetzt hier der Conf Block ###
            override {
                required = 1
                field = image

                wrap = uploads/pics/
                listNum.stdWrap.data = register:IMAGE_NUM
            }
        }
    }
}

### Zurechtschneidung passbild auf die Crop Maße von 120x175 Pixel Abmessung
30.file.width = 120c+75
30.file.height = 175c

}

}

```

Hinweis:

Wie in Abbildung 1 vermutlich einigen aufgefallen ist, sind in dem **Eingabefeld „Bildbeschreibung für Ansprechpartner“** einige TAGS definiert, die nicht HTML Konform sind.

Das sind TYPO-Tags die später über den Frontendparser in konformes HTML umgewandelt werden und es so den Redakteuren zudem intuitiv ermöglichen etwa eine Telefon oder Faxnummer entsprechend zu setzen.

Diese können im Frontend später entsprechend mit CSS Klassen versehen werden. Aber dazu mehr jetzt in im folgenden Abschnitt des folgenden Abschnitts.

Das folgende Script MUSS vor dem obigen Script im Template definiert und wird später mittels **10.parseFunc < lib.AdvancedCustomTags** entsprechend zugewiesen werden. Der Sinn sollte sich entsprechend selbst erklären, wenn man ein Minimum an Typoscript Kenntnissen hat.

Benutzerspezifische CustomTags für das ShortCutElement

lib.AdvancedCustomTags = TEXT

lib.AdvancedCustomTags {

tags.tel = TEXT

tags.tel {

current = 1

wrap = <div class="phone">Tel.: |</div>

}

tags.fax = TEXT

tags.fax {

current = 1

wrap = <div class="fax">Fax.: |</div>

}

tags.mail = TEXT

tags.mail {

outerWrap = <div class="mail">|</div>

innerWrap.cObject = TEXT

innerWrap.cObject {

**value = **

typolink.parameter.data = parameters : allParams

typolink.ATagParams = class="mail"

}

preCObject = TEXT

preCObject.value = eMail:

}

tags.address = TEXT

tags.address {

current = 1

wrap = <div class="address"> |</div>

}

}

Diese Tags ermöglichen das gesonderte Formatieren mittels CSS später nach der Bearbeitung durch den Parser, wie etwa Fax- und Telefonnummer, wie auch der Email Adresse ohne hier die teilweise für Redakteure komplexe Syntax von „richtigen“ HTML.

5. Schritt: „Verknüpfen über das Inhaltselement Datensatz Einfügen“

Sind die vorangegangenen Schritte entsprechend befolgt worden, müssen wir in der Root Seite im TS-Config Feld noch final eine Eintragung ergänzen, so dass die Änderungen TS-Setup entsprechend verwendet werden können mit dem Inhaltselement Datensatz einfügen.

Anpassungen für das Element Shortcut

für einfügen in Bildelementauflistung

TCEFORM.tt_content.layout.types {

shortcut.addItems.10 = Layoutvorlage für die Ansprechpartner

}

Damit erweitern wir später beim Anlegen des Inhaltselements „Datensatz einfügen“, das Layout Feld um ein weiteres Listenelement. Den notwendigen Code für die Funktion dahinter haben wir bereits in Schritt 4 dieses Tutorials eingefügt.

Wir können nun ein Inhaltselement vom Typ „Datensatz einfügen“ erstellen und binden auf diesen Weg das Inhaltselement ein was wir vorher in Schritt 3 in dem SysOrdner mit den erweiterten TCEFORM Einstellungen hinterlegt haben. Schlussendlich stellen wir in diesem Inhaltselement die Layoutvorlage entsprechend um auf den Wert der in *Abbildung 3* abgebildet ist.

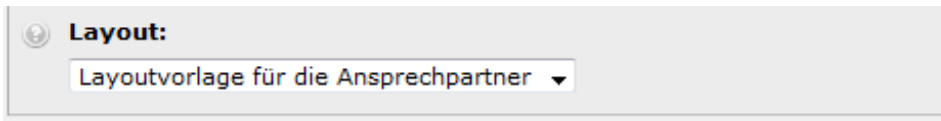


Abbildung 3: Die Layoutvorlage für „Datensatz einfügen“

Das Ergebnis im Frontend sollte aussehen in *Abbildung 4*. In diesem Beispiel haben wir nur reine TYPO3 spezifischen Maßnahmen erläutert, die Erstellung des notwendigen CSS Codes wurde hier außen vorgelassen.

Fazit:

Wir erhalten ein Maßgeschneiderten „View“ wo letztendlich einfach spezifische Inhalte ergänzt werden können und was bereits falls nicht gesetzt ein entsprechendes Bild für den Ansprechpartner kleiner rechnet. Voraussetzung für letzteres ist eine passende ImageMagick Installation natürlich



Abbildung 4: Endergebnis im Frontend für Tutorial

Praxisbezogene Beispiele für dieses Tutorial

<http://www.tug-ks.de/index.php?id=11> (Datensatz einfügen)

<http://www.tug-ks.de/index.php?id=19> (SysOrdner mit hinterlegten Datensatz und TCEFORM)