

Harry Glatz

Workshop TemplaVoilà

Teil 2: Eigene Content-Elemente erstellen

Nachdem der Workshop in T3N Nr. 13 gezeigt hat, wie Templates erstellt und gemappt werden, geht es diesmal darum, eigene Content-Elemente zu konzipieren. Mit Hilfe von TemplaVoilà ist es sehr einfach, diese zu erstellen und dem Redakteur zur Verfügung zu stellen.

HTML und CSS

Wie bereits in Teil 1 werden auch in diesem Workshop ganz einfache und kurzgehaltene Vorlagen verwendet. Prinzipiell können die verschiedenen Bereiche und Content-Elemente beliebig erweitert und mit CSS gestaltet werden. Sämtliche Dateien und Listings in diesem Workshop finden Sie über den am Ende des Artikels abgedruckten Softlink auf der T3N-Website.

Als erstes Content-Element erstellen Sie einen zweispaltigen Container, der es ermöglicht, Standard-Content-Elemente in zwei Spalten zu platzieren. Das HTML-Template besteht aus einem umschließenden Container, der wiederum zwei Div-Container enthält. Diese innen liegenden Container werden dann über CSS so formatiert, dass sie nebeneinander stehen.

```
HTML-Vorlage (zweispalten-50-50.html)
<html>
<head>
<title>Zwei Spalten</title>
<link href="mainstyle.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="zweispaltencontainer">
  <div class="linke-spalte"> </div>
  <div class="rechte-spalte"></div>
  <div class="clearer"></div>
</div>
</body>
</html>
```

Listing 1

```
CSS-Vorlage (mainstyle.css)
.zweispaltencontainer .linke-spalte{
float:left;
width:50%;
}

.zweispaltencontainer .rechte-spalte{
float:left;
width:50%;
}

.clearer{
float:none;
clear:both;
}
```

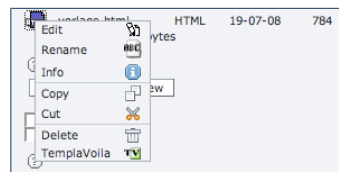
Listing 2

Die HTML-Datei muss, wie auch das Page-Template, im Fileadmin-Ordner auf dem Server Ihrer TYPO3-Installation in einem Unterverzeichnis mit dem Namen „templates“ bereitgestellt werden. Die CSS-Definitionen werden einfach dem vorhandenen Stylesheet hinzugefügt. Sind die Daten bereitgestellt, können Sie diese

im nächsten Schritt mit Hilfe von TYPO3 und TemplaVoilà ansprechen.

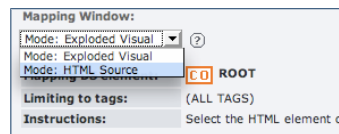
Content-Element für „2 Spalten“ erstellen

Wechseln Sie im Backend zunächst in das Modul „Filelist“. Dort navigieren Sie zu Ihrem Template-Verzeichnis und klicken mit der Maus auf das Icon links neben dem HTML-Template „zweispalten-50-50.html“. In dem kleinen Aufklappmenü wählen Sie den untersten Punkt „TemplaVoilà“.



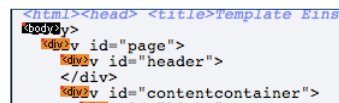
Auf der darauffolgenden Seite können Sie das Mapping anlegen, der erste Schritt ist das Mapping des Root-Elements. Das Root-Element ist in diesem Fall, genau wie beim Page-

Template, das Body-Tag der HTML-Datei. Klicken Sie also auf den Button „Map“ auf der rechten Seite.



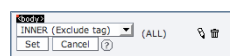
Um das Mapping zu erleichtern, wählen Sie in dem Dropdown-Menü „Mapping Window“ die Option „Mode: HTML Source“ aus. Sie sehen nun den

HTML-Code der Seite, in dem Sie auf das Body-Tag klicken.

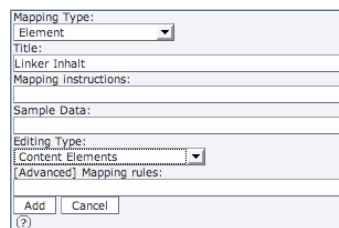


Dieses wird dadurch verwendet. Wählen Sie in der anschließenden Maske den Mapping-Type „INNER“ aus. Sie können

jetzt mit dem Mapping der verschiedenen Elemente innerhalb des Bods fortfahren.



Als nächstes legen Sie ein neues Feld an (Add new field), benennen es mit dem (frei wählbaren) Namen „field_inhaltlinks“ und klicken auf „Add“.



Diesem neuen Feld geben Sie in der angezeigten Eingabemaske den Titel „Inhalt Linke Spalte“ und wählen als Editing Type „Content Elements“ aus. Dies ermöglicht Ihnen später im Backend, in diesem Bereich der Seite beliebige Content-

Elemente hinzuzufügen. Danach klicken Sie auf „Add“.

Jetzt müssen Sie auch dieses Feld noch mappen und klicken hierfür auf der rechten Seite auf „map“. In der HTML-Ansicht wählen Sie den Container für die linke Spalte aus (<div class=„linke-spalte“>) und wählen „INNER“ als Mapping-Type.

Den selben Prozess wiederholen Sie anschließend nochmal für die rechte Spalte. Benennen Sie das Feld mit „field_inhaltrechts“ und klicken Sie auf „Add“.

Diesem neuen Feld geben Sie in der angezeigten Eingabemaske den Titel „Inhalt Rechte Spalte“ und wählen als Editing Type

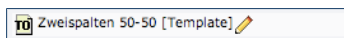
„Content Elements“ aus. Nachdem Sie auch dieses Feld erstellt haben, müssen Sie es noch mappen.



Das Mapping ist damit abgeschlossen, und Sie können auf „Save as“ in der unteren Leiste klicken. Auf der nächsten Seite geben Sie dem Content-Element einen Namen (z. B. „Zweispalten 50-50“), wählen in der Dropdownbox „Content Element“ aus und klicken auf „Create TO and DS“. Das Content-Element wird jetzt im General-Storage-Folder als „DS“ und „TO“ gespeichert.

Dieses Content-Element sollte jetzt beim Anlegen eines neuen Content-Elements schon in der Liste der vorhandenen Elemente (Assistent) erscheinen. Jedoch verwendet es hier noch das Standard-Icon, das wenig über das Element aussagt.

Gerade wenn man als Administrator seinen Redakteuren mehrere eigene Content-Elemente zur Verfügung stellen will, ist es hilfreich, eigene Icons und Beschreibungen für jedes Element zu hinterlegen. Hierfür wechseln Sie in das TemplaVoilà-Modul in der linken Hauptnavigation und wählen den General-Storage-Ordner (im deutschen Backend: „Allgemeine Datensatzsammlung“) aus. Aus den oberen Reitern wählen Sie den Reiter „flexible Content Elements“ und sehen dort das „DS“ und „TO“ Ihres neuen Elements.



Klicken Sie auf den Bleistift neben Ihrem Template-Objekt (TO). Daraufhin öffnet sich eine eigentlich selbsterklärende Maske.

Hier können Sie eine Beschreibung des Elements hinterlegen. Tragen Sie zum Beispiel folgenden Text ein: „Zweispalten Container mit einer 50%-50%-Aufteilung“. Im Feld „Icon“ können Sie ein Icon für das Flexible-Content-Element (FCE) angeben. Dieses Icon muss entweder vom Dateityp „PNG“ oder „GIF“ sein und wird in unveränderter Größe dargestellt, von TYPO3 also nicht skaliert. Standard-TYPO3-Icons haben eine Größe von 18x16 Pixeln (Breite x Höhe).

Nach dem Speichern wird Ihr Content-Element beim Anlegen neuer Inhalte auf einer Seite im Assistenten mit Icon und Beschreibung sehr übersichtlich gelistet und einem Redakteur wird somit die Arbeit erleichtert.

Content-Element „Bilderüberschrift“ erstellen

Um zu veranschaulichen, wie TypoScript in den Flexible-Content-Elements funktioniert, erstellen Sie als nächstes ein Content-Element für „Bildüberschriften“. Dieses soll einem Redakteur ermöglichen, eine Überschrift einzufügen, die als Bild dargestellt wird. Dadurch werden Überschriften möglich, die zum Beispiel mit einer bestimmten Schriftart, etwa der Hausschriftart, gerendert werden. Auch hierfür brauchen Sie eine HTML-Vorlage.



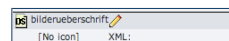
Listing 3

Wie bereits die HTML-Vorlage für das zweispaltige Layout, speichern Sie auch diese Datei im Template-Ordner des Fileadmins. Wechseln Sie anschließend wieder in das Filelist-Modul in der linken Navigation, navigieren Sie in den Template-Ordner und rufen per Klick auf das Icon der Datei das TemplaVoilà-Modul auf. Mappen Sie erneut das Body-Tag als Root-Element und wählen Sie „INNER“ als Mapping-Type. Anschließend erstellen Sie ein neues Field (field_headline) und klicken auf „Add“.

Der folgenden Maske vergeben Sie den Titel „Bilderheadline“, wählen als Editing-Type „Header field, Graphical“ aus und klicken auf „Add“. Danach mappen Sie dieses Feld und wählen beim Mapping das Tag „<div class=“bilderheadline“></div“ aus. Auch hier ist der Mapping-Type „INNER“.

Um dem gerenderten Bild ein Alt-Tag vergeben zu können, fügen Sie jetzt noch ein Feld hinzu (field_alntag) und geben auch diesem in der folgenden Maske einen Titel (Alt-Tag der Headline). Als Editing-Type wählen Sie hier „Plain Input“ aus und klicken wieder auf „Add“. Da dieses Element nicht dargestellt wird, müssen Sie es auch nicht mappen.

Das Feld fügen wir dem gerenderten Bild als Alt-Tag erst später im TypoScript hinzu. Speichern Sie das Element nur per Klick auf „save as“, geben Sie ihm einen Namen (bildueberschrift) und speichern Sie es als neues Content-Element.



Nun müssen Sie in das TemplaVoilà-Modul in der linken Navigation wechseln, den Reiter „Flexible Content“ wählen und dort zu dem eben gespeicherten Element scrollen. Klicken Sie auf den Bleistift der ersten Zeile, um das XML dieses Elements betrachten zu können. Suchen Sie darin folgenden Bereich:



Listing 4

Hier sehen Sie, dass das Bild mittels GIFBUILDER und den entsprechenden TypoScript-Angaben erstellt wird. Diese können Sie nun an Ihre Bedürfnisse anpassen. Informationen zu den Parametern, die Sie hier verwenden können, finden Sie in dem jeweiligen Kapitel der TypoScript-Referenz „TSref“ [1] [2].

Ein angepasstes Beispiel könnte wie folgt aussehen:



```

10.text.current = 1
10.text.case = upper
10.fontColor = #FFCC00
10.fontFile = t3lib/fonts/vera.ttf
10.niceText = 0
10.offset = 0,14
10.fontSize = 14
}
</TypoScript>

```

Listing 5

Wichtig ist hier die Zeile mit dem Alt-Tag (altText). Wie man sehen kann, wird das nicht gemappte Feld „field_alttag“ verwendet, um dem Bild einen Alt-Text zu vergeben. Ist „field_alttag“ leer, wird stattdessen der Inhalt des Felds „Bilderheadline“ verwendet.

Die zweite Änderung wurde an der Breite des erstellten Bilds vorgenommen. Hier wurde die feste Breite von 200 Pixeln auf einen variablen Wert gesetzt. Dieser orientiert sich an dem eingegebenen Text (10=TEXT).

Wenn Sie mit den Änderungen an der Definition fertig sind, speichern Sie das XML. Wichtig: Wenn Sie das Mapping dieses Elements verändern, werden alle von Ihnen im TypoScript dieses XML-Dokuments getätigten Änderungen gelöscht. Es empfiehlt sich also, vor einem Ändern des Mappings alle eingegebenen Parameter zu kopieren, da diese durch das neue Mapping überschrieben werden.

Nun sollte im Assistenten für Content-Elemente auch dieses Element erscheinen. Redakteure können einfach einen Text und ein Alt-Tag angeben, im Frontend wird daraufhin ein Bild mit den gewählten Parametern dargestellt. Optional können Sie noch, wie schon beim vorherigen Flexible-Content-Element, auch diesem Element ein eigenes Icon und eine Beschreibung hinzufügen.

Generell ist Folgendes festzuhalten: Jedes Feld im XML hat einen Part, in dem man TypoScript angeben und somit seine Flexible-Content-Elemente an die eigenen Bedürfnisse anpassen kann.

Fazit

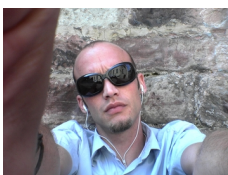
Benötigt man ein Content-Element, das TYPO3 standardmäßig nicht mitliefert oder bei dem es sehr aufwändig wäre, es an den Style der Seite anzupassen, empfiehlt sich der Einsatz von TemplaVoilà. Damit ist es auf einfache Weise möglich, ein eigenes individuelles Content-Element zu erstellen. Haben Sie erst das HTML und das dazugehörige CSS erstellt, ist das eigentliche Erstellen des Content-Elements nur noch ein kleiner Schritt. Mit TemplaVoilà lassen sich so auch komplexe Elemente erstellen und durch die Möglichkeit, TypoScript einzusetzen, bleiben fast keine Wünsche offen.

Links und Literatur

 [Softlink 2211](#)

- [1] Original-TSref auf typo3.org: http://typo3.org/documentation/document-library/references/doc_core_tsref/4.1.0/view/
- [2] Deutsche Übersetzung der TSref: <http://www.typo3.net/tsref/>

DER AUTOR



Harry Glatz ist Mitbegründer von analog multimedia (www.analog.de) und seit 1996 im IT-Bereich tätig. Seit Version 3.5 arbeitet er ausschließlich mit TYPO3 als CMS und hat damit zahlreiche Projekte umgesetzt.