

OVB Konstanten

Umgang mit mehrfach eingesetzten Begriffen, Daten und Phrasen

Autor: Ronny Bartsch, OVB Holding AG

Problemstellung

Im redaktionellen Betrieb einer OVB Webseite kommen einige Begriffe, Daten und Phrasen zum Einsatz, die mehrfach und an verschiedensten Stellen eingesetzt werden. Solche Elemente sind zum Beispiel der Name der Landesgesellschaft oder auch die einzelnen Kontaktdaten einer OVB Landesgesellschaft.

Solche Elemente kommen an verschiedensten Stellen zum Einsatz. Diese können zum Beispiel einzelne Fließtexte sein, ein Vorkommen in Impressum und Datenschutzerklärungen ist die Regel und Kontaktboxen oder Footerbereiche sind weitere Einsatzbereiche.

Auch sind die technischen Wege, über die diese Elemente in die Darstellung gelangen, vielfältig. Zum einen soll der Redakteur diese Texte mit dafür vorgesehenen Platzhaltern im Editor einsetzen können aber auch direkt durch ein Fluid-Template sollen die Elemente auf die Seite gelangen. Zum Beispiel bei der Angabe, für welches Unternehmen ein Berater berät oder in der interaktiven OVB Europakarte.

Durch die Vielfalt der Einsatzmöglichkeiten und technischen Publikationswege ergibt sich ein Problem: **Wie lassen sich alle Vorkommen eines bestimmten Elementes identifizieren, wenn sich der Inhalt des Elementes verändert und aktualisiert werden muss?**

Wie sieht die derzeitige Umgang aus?

In der gegenwärtigen Konfiguration sind die Begriffe, Daten und Phrasen aus unterschiedlichen Quellen in denen sie redundant gepflegt sind. Zum Teil sind sie in Sprachressource definiert zum anderen Teil aber auch in verschiedenen Typoscriptfiles, verstreut im Webspace. Auch sind viele dieser Elemente einfach als Klartext in den Fließtexten eingesetzt worden. Im Falle einer Aktualisierung ist es leicht, einige Vorkommen des Elementes zu übersehen. Auch gibt es keine Konsistenz darin, wer die Inhalte eines der Elemente bearbeiten kann. (Berechtigung)

Lösungsansatz Typo3-Konstante

Die Typo3-Konstante bietet hierfür einen Lösungsansatz. In einer Typoscript-Konstante können Begriffe, Daten, Phrasen und sogar HTML-Code (z.B. Links) an einer einzigen Stelle gespeichert werden. Und wenn diese Definition möglichst weit oben im Seitenbaum geschieht, kann der entsprechende Wert global bzw. unterhalb unterschiedlicher Einstiegspunkte als Platzhalter oder als Variable in Fluid eingesetzt werden. Dies ist zum Beispiel für die Generierung der interaktiven OVB Europakarten notwendig, sollen doch hier immer aktuelle Daten aller OVB Landesgesellschaften dargestellt werden.

Einsatz

Die Bereitstellung der Konstanteninhalte kann über alle benötigten technischen Wege, als Platzhalter und auch über Fluidtemplates, geschehen. Damit ist nur eine einzige zentrale Konstantendefinition nötig. Alle derzeit bestehende Redundanzen würden entfallen.

Aktualisierung

Der Inhalt einer Variable kommt unterhalb unterschiedlicher Einstiegspunkte oder anders formuliert, auf unterschiedlichen OVB Landeswebseiten zur Darstellung. Nicht nur daher ist eines konsequent: **Die Konstante darf nicht durch einen Redakteur bearbeitet werden sondern nur durch einen Administrator.** Der Unterschied zwischen Administration und Redaktion besteht darin, ob global Einfluss genommen werden kann oder nur lokal eingeschränkt auf einen bestimmten Teil des Seitenbaumes.

Administratoren wird im Templatemodul ein Interface zur Verfügung gestellt, in dem alle Constanten mit ihren Werten verwaltet werden können. Die Konstanten sind dabei in Kategorien geordnet und mit Standardwerten vorbelegt.

Nachteile

Diese Vorgehensweise birgt den Nachteil, dass Kollegen mit Administrationsaccount diese zusätzliche Aufgabe erhalten. Dieser Nachteil ist aber gering weil sich Konstantenwerte nicht allzu häufig ändern. Auch wird die Administration der Konstanten durch die Einrichtung des Interfaces stark vereinfacht.

Vorteile

Die Vorteile überwiegen. Erfahrungsgemäß erreicht uns die Information über die Aktualisierung von Unternehmensdaten vor dem entsprechenden Request des Redakteurs der betreffenden OVB Landesgesellschaft. Sollte uns die Information doch erst über einen Request erreichen, so haben wir an dieser Stelle einen wertvollen Hinweis, um zu überprüfen, ob die Information über die Änderungen von zum Beispiel Kontaktdaten einer Landesgesellschaft sauber an die OVB Holding AG gemeldet wurden und alle Kollegen Bescheid wissen.

Ein Vorteil dieses Lösungsansatzes gegenüber des zum Teil im Einsatz befindlichen alternativen Lösungsansatz über die Pflege in Language-Ressource-Files besteht darin, dass der Administrator die Werte im Backend umgehend aktualisieren kann und wir hierfür keinen Merge-Request stellen müssen. Dies würde automatisch einen Testlauf initiieren, der bis zu einer Woche dauern kann. Auch können wir somit sprachabhängige Inhalte (z.B. Bezeichner von geografischen Ländern in Europakarte) von nicht-sprachabhängigen Inhalten (z.B. Name einer OVB Landesgesellschaft) trennen und die Konsistenz in unserer Anwendung steigern.

Umsetzung

Schritt 1: Einrichtung des Administrationsinterfaces

In der Preview-Instanz wird Template-Modul der Konstanteneditor um weitere Kategorien und den zugehörigen Eingabefeldern erweitert. Dabei wird eine Kategorie für jede OVB Landesgesellschaft erstellt und dort je zwei Unterkategorien für die Unterscheidung von Kontaktdaten und Kennzahlen erstellt. Innerhalb der Unterkategorien befinden sich dann die Felder für Kontaktdaten und Kennzahlen.

Alle Konstantenbezeichner müssen dabei eine Kennung für den Bezug auf eine OVB Landesgesellschaft beinhalten. So gelingt eine Verwendung aller Konstanten für die Generierung der interaktiven Europakarte.

OVBEUUnternehmensname, OVBDEUnternehmensname, ...

`/app/web/typo3conf/ext/rb_template/Configuration/TypoScript/constants.typoscript`

<code>styles.content {</code>	Konstanten
<code> # customcategory=ovbeuconstants=Konstanten OVB EU</code>	1 .Unternehmensname
<code> # customsubcategory=ovbeucontactdata=Kontaktdaten</code>	2. Straße
<code> # customsubcategory=ovbeukeyfigures=Kennzahlen</code>	3. Hausnummer
<code> #cat=ovbeuconstants/contactdata/a; type=string; label=Unternehmensname</code>	4. Postleitzahl
<code> ovbeuunternehmen = OVB Holding AG</code>	5. Ort
<code> #cat=ovbeuconstants/ovbeucontactdata/b; type=string; label=Straße</code>	6. Telefon 1
<code> ovbeustrasse = Heumarkt 1</code>	7. Telefon 2
<code> #cat=ovbeuconstants/ovbeukeyfigures/a; type=string; label=Kunden</code>	8. Fax
<code> ovbeucostumers = 5</code>	9. E-Mail
<code> </code>	10. Web-URL
<code> # customcategory=ovbdeconstants=Konstanten OVB DE</code>	11. Facebook-URL
<code> # customsubcategory=ovbdecontactdata=Kontaktdaten</code>	1 2.Instagram-URL
<code> # customsubcategory=ovbdekeyfigures=Kennzahlen</code>	13. LinkedIn-URL
<code> #cat=ovbdeconstants/ovbdecontactdata/a; type=string; label=Unternehmensname</code>	14. Twitter-URL
<code> ovbdeunternehmen = OVB Vermögensberatung AG</code>	15. Youtube-URL
<code> #cat=ovbdeconstants/ovbdecontactdata/b; type=string; label=Straße</code>	16. Manager 1
<code> ovbdestrasse = Heumarkt 1</code>	17. Manager 2
<code> #cat=ovbdeconstants/ovbdekeyfigures/a; type=string; label=Kunden</code>	18. Manager 3
<code> ovbdecostumers = 15</code>	19. Kunden
<code> </code>	20. Partner
<code> }</code>	21. Mitarbeiter

Schritt 2: Marker-Mapping

Dem root-Typoscript wird ein Bereich hinzugefügt, in dem die Konstanten auf Marker gemappt werden, die dann redaktionell im Editor in der Form ###unternehmensname### eingesetzt werden können. Hierbei sollen unter jedem Einstiegspunkt die gleichen Marker mit den gleichen Bezeichnern zur Verfügung stehen aber abhängig vom übergeordneten Einstiegspunkt durch unterschiedliche Werte ersetzt werden. Dies ist durch entsprechende Typoscript-Conditions umsetzbar.

Die Liste der zur Verfügung stehenden Marker wird dokumentiert und in Confluence zur Verfügung gestellt.

/app/web/typo3conf/ext/rb_template/Configuration/TypoScript/setup.typoscript

```
[PIDinRootline = 354]
    constants {
        unternehmensname = {$styles.content.ovbeuunternehmen}
        strasse = {$styles.content.ovbeustrasse}
        .....
    }
    lib {
        unternehmensname = {$styles.content.ovbeuunternehmen}
        strasse = {$styles.content.ovbeustrasse}
        .....
    }
[global]

PIDinRootline = 1]
    constants {
        unternehmensname = {$styles.content.ovbdeunternehmen}
        strasse = {$styles.content.ovbdestrasse}
        .....
    }
    lib {
        unternehmensname = {$styles.content.ovbdeunternehmen}
        strasse = {$styles.content.ovbdestrasse}
        .....
    }
[global]
.....
```

Schritt 3: Anpassung von Fluidtemplates

Die bestehenden Fluidtemplates müssen auf die Verwendung der Konstanten umgeschrieben werden. Dabei kann es einen Fallback auf das bestehende Verhalten geben.

Aufruf in Fluid:

```
.....  
  
    {f:cObject(typoscriptObjectPath: 'lib.unternehmensname')}  
  
.....
```

Schritt 4: Test und Deployment

Nach einem gründlichen Test von Administration und Ausgabe der Konstanten werden alle Anpassungen über die Staging-Instanz auf die Live-Instanz deployt.

Schritt 5: Konstanten mit Werten belegen

Ein Administrator muss in diesem Schritt alle Konstanten mit ihren aktuellen Werten belegen.

Schritt 6: Anpassung Content

Jedes Auftreten von nun in Konstanten gespeicherten Begriffen, Daten und Phrasen muss im Content identifiziert werden und durch seinen Marker ersetzt werden.

Schritt 7: Aufräumen

Alle derzeit im Einsatz befindlichen Konstantendefinitionen in Language-Ressourcenfiles und Typoscript-Files müssen entfernt werden.

Ist es ein Projekt?

Nein. Alle Arbeiten können durch mich erledigt werden. Ich würde mich hierzu noch einmal mit TWT abstimmen und letztendlich nur Merge-Requests stellen.