

# 0. Einführung

## 0.2 Begriffe/Funktionsweise

Bei Contenido wird zwischen dem "Backend", dem Bereich, in dem die Redakteure und Administratoren arbeiten und dem "Frontend", der Webseite, dem Ergebnis aller Bemühungen, unterschieden.

### **Mein Contenido: Das Backend:**

Das Backend ist über das Unterverzeichnis /contenido erreichbar:

Backend: `http://<IhreDomain>/<ggf. Installationsverzeichnis>/contenido`

Das Contenido-Backend basiert auf PHP und JavaScript, daher muss für die Nutzung des Backends (also Contenido) JavaScript aktiviert sein. Auch PopUp-Blocker sollten für die Installationsdomain deaktiviert werden, da sonst einige Funktionen nicht genutzt werden können.

### **Meine Webseite: Das Frontend:**

Contenido ist mandantenfähig - es können also Webseiten mehrerer Mandanten in einer Contenido-Installation verwaltet werden. Für jeden Mandanten wird i.A. ein eigenes Mandantenverzeichnis festgelegt, über welches das Frontend (die Webseite) erreicht werden kann. Die Webseite des Beispielmantanten liegt z.B. in /cms:

Frontend (Beispielmantant): `http://<IhreDomain>/<ggf. Installationsverzeichnis>/cms`

Welche Anforderungen an den Browser für das Frontend (die Webseite) gelten, ergibt sich aus der Gestaltung durch den Administrator bzw. Redakteur. Für den Besuch einer durch Contenido erzeugten Webseite ist also nicht automatisch JavaScript erforderlich.

### **Ganz viele Bausteine: Layouts/Module/Templates:**

Was bei anderen Content Management Systemen das Template ist, ist bei Contenido das Layout. Es enthält das HTML-Grundgerüst einer Seite.

Es kann mit jedem beliebigen Webseiten-Editor erstellt werden - der einzige Unterschied ist, dass später an den Stellen, an denen unterschiedliche Bilder, Überschriften, Navigationselemente/Menüs und andere Funktionen stehen würden, so genannte Container-Tags gesetzt werden. Container-Tags sind Platzhalter, die später mit den eigentlichen Funktionen ersetzt werden.

(ganz einfaches, heutzutage keinen Internet-Standard mehr erfüllendes Beispiel für ein Layout mit einem Container):

```
<html>
<head>
<title>Meine Seite</title>
</head>
```

```
<body>
<container id="1" name="Text" types="Content" mode="optional" default="Text-
```

Seite 1 / 2

# 0. Einführung

```
HTML (1)">Text</container>
</body>
</html>
```

Auf diese Weise kann ein Grundgerüst für unterschiedlichste Seiten verwendet werden, z.B. als Startseite, als Textseite oder als Bildergalerie.

Jeder variable Inhalt (genau genommen alles, was sprachabhängig sein könnte) und jede Funktion wird dabei als Modul umgesetzt. Module können - müssen aber nicht - PHP-Code enthalten. Die einfachsten Module stellen einfach nur einen Bereich zum Eingeben von Text und zur Auswahl von Bildern zur Verfügung, andere Module erzeugen das Menü der Webseite (die Navigation) oder ermöglichen dem Nutzer, sich auf der Webseite zu registrieren.

An welcher Stelle im Layout welches Modul eingebunden wird, wird im Template definiert. I.A. verfügt eine Contenido-Webseite über wenige Layouts, ca. 20 Module und in etwa genau so viele Templates.

Für die Mathematiker: Layout + Module = Template

## **Von Bäumen, Kategorien und Artikeln:**

Mit Kategorien werden die Inhalte (Artikel) gegliedert und zusammengefasst (z.B. wie die Dateien in einem Verzeichnis). Alle Kategorien, die unter der obersten Kategorie liegen, werden als Baum bezeichnet. Man kann mehrere Bäume anlegen (z.B. Hauptnavigation, Hilfsnavigation) - und nein, den "Wald" gibt es als Begriff in Contenido dann doch nicht...

Templates können Kategorien und Artikeln zugewiesen werden. Verfügt ein Artikel nicht über ein eigenes Template, wird das Template der Kategorie verwendet.

## **Und wo werden die fertigen Webseiten gespeichert?**

Nirgends. Die eigentliche Webseite wird aus Layout, Modulen, Inhalten (z.B. Texten, Verweise auf Bilder) und Funktionen virtuell aus den Angaben in der Datenbank zusammengesetzt und ausgegeben. Na gut, zugegeben, eine Art Rohfassung wird auch noch in der Datenbank gespeichert - damit diese Zusammenstellung nicht bei jedem Aufruf erfolgen muss; das ist aber für die normale Nutzung nicht relevant.

Aus diesem Aufbau ergibt sich, dass alle Seiten im Prinzip die gleiche URL haben: `http://<DeineDomain>/<ggf. Installationspfad>/<Mandanten-Verzeichnis>/front_content.php?idart=1234` (nur die "1234" ändert sich)

Es ist mit Drittlösungen möglich, eine "virtuelle" URL mit Verweisen auf Ordner- und Dateistrukturen zu verwenden, die wie eine normale URL aussieht. Stichwort: `mod_rewrite`

Eindeutige ID: #1047

Verfasser: Björn Behrens

Letzte Änderung: 2007-09-30 18:16