

Vorwort

Das folgende Tutorial soll einen möglichen Weg beschreiben, wie in Sefrengo ein vorgefertigtes Design in das CMS eingearbeitet wird.

Normalerweise werden diese vorgefertigten Designs unter dem Begriff „Template“ behandelt. Da in Sefrengo der Begriff „Template“ das zentrale Verbindungselement von Sefrengo bezeichnet, das die Module mit dem Layout verbindet, wird im Folgenden der Begriff „Design“ für das vorgefertigte HTML-Template verwendet.

Zunächst muss noch erwähnt werden, dass der Administrator eine Gliederung der gewünschten Seiteninhalte hinsichtlich Struktur und gewünschtem Funktionsumfang erstellt haben sollte, bevor sie/er sich an die Arbeit macht eine Internetseite in ein CMS einzuarbeiten.

Inhalte dieses Tutorials:

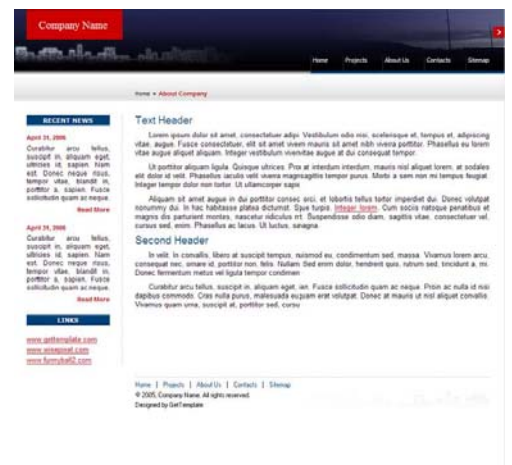
- Anlegen eines Layouts mit CMS-Containern
- Import und Installation von Modulen
- Einfügen von Modulen in Templates
- Zuweisung von Templates auf Seiten und Ordner
- CSS und Grafiken hochladen
- Unterschiedliche Einbindung von CSS

Vorgefertigtes Design von [oswd.org](http://www.oswd.org) herunterladen

Als Einstiegsbeispiel wird hier ein vorgefertigtes Template von www.oswd.org verwendet. Bei diesem Layout verwendet der Autor eine html-Seite mit <DIV>-Layern, die über CSS gestaltet werden. Darüber hinaus werden auch weitere TAGs mit diesem zentralen CSS formatiert. (Quelle: http://www.oswd.org/files/designs/2996/night_city/)

Seitenstruktur des vorgefertigten Designs analysieren

Als erstes öffnet man die HTML-Seite des vorgefertigten Designs und analysiert die Struktur des vorgegebenen Designs. Dabei hat sich die Extension „Web-Developer-Bar“ des Browsers „Firefox“ als sehr hilfreiches Werkzeug erwiesen.



Das Seitendesign enthält drei Navigationsbereiche:

- Die Hauptnavigation oben,
- eine Pfad-Navigation darunter und
- eine weitere Navigation mit den Hauptmenüpunkten im sog. „Footer“.

Des Weiteren ist der Hauptbereich in 2 Spalten aufgeteilt. Die linke Spalte enthält Texte, die später mit Textfeldern bestückt werden sollen bzw. für Seitenbereiche mit mehr Ebenen eine Unternavigation bereitstellen soll. Der rechte Bereich soll verschiedene Inhalte aufnehmen. Hier stehen später Texte und Bilder – je nach Bedarf.

Betrachtet man den HTML-Code, so werden diese Bereiche mit <DIV>-Layer erzeugt. Die Positionierung und Formatierung geschieht über die CSS-Datei „Styles.css“ im Ordner „Images“ des downgeloadeten Pakets. Diese Dateien sollen hier nur als exemplarisches Beispiel verstanden werden.

Struktur des vorgefertigten Designs:

Betrachtet man den Quellcode des HTML-Gerüsts näher, fällt eine lineare Struktur von DIV-Layern auf:

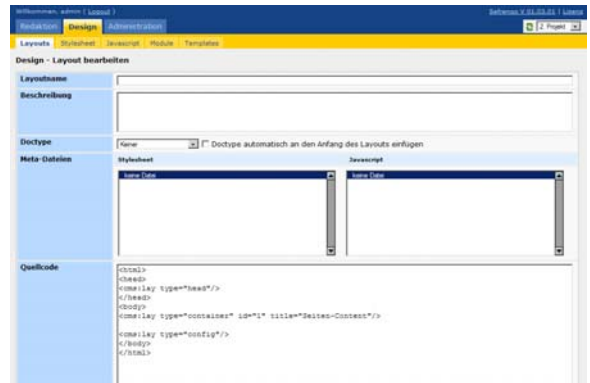
```
...<div id="HEADER">...  
...<div id="PATH">...  
...<div id="SIDEBAR">...  
...<div id="CONTENT">...  
...<div id="FOOTER">...
```

Grundlayout von Sefrengo:

Sefrengo bindet über das Layout verschiedene Strukturelemente, wie z. B. Header-Informationen (Titel, Doctype...) und die schon erwähnten Container ein. Das Grundlayout bietet hier die Mindestinhalte, die in einem Sefrengo-Layout auftauchen müssen:

```
<html>  
<head>  
<cms:lay type="head"/>  
</head>  
<body>  
<cms:lay type="container" id="1" title="Seiten-Content"/>  
  
<cms:lay type="config"/>  
</body>  
</html>
```

Um die Seitengestaltung in Sefrengo zu übertragen benötigt man zunächst das HTML-Gerüst der späteren Seite – das sog. „Layout“. Im eingeloggten Backend-Bereich von Sefrengo kann man ein vorgefertigtes Layout über „Design > Layouts > Neues Layout“ ein Standard-Layout-Gerüst für Sefrengo angelegt werden. In diesem Dialogfenster gibt einen Namen, eine mögliche Beschreibung und den Doctype an. Entscheidend für die spätere Nutzung des Layouts sind die sog. „CMS-Container“. Sie stehen für spätere Module, die dann mittels dem sog. „Template“ in dieses Layout an Stelle der Container eingebunden werden.



Bevor die Container jedoch in das HTML-Gerüst eingesetzt werden, muss zunächst das vorgefertigte Design von Überflüssigem befreit werden. Grundsätzlich gilt:

Alle Inhalte, die dynamisch veränderbar sein sollen, werden durch Container ersetzt. Also alle Navigationsbereiche, Textbereiche, die auf unterschiedlichen Seiten neu beschrieben werden sollen oder Spezialfunktionsbereiche wie z. B. ein Suchfeld werden in der HTML-Vorlage gelöscht und durch Container ersetzt. Ein Beispiel soll das verdeutlichen.

Schnittmenge ohne überflüssige Inhalte:

Löscht man die überflüssigen Elemente aus der HTML-Seite des vorgefertigten Designs, kann man diese Lücken innerhalb der DIV-Layer mit den CMS-Containern auffüllen.

```
<html>
<head>
<cms:lay type="head" />
</head>
<body>

<div id="HEADER">
    <h1><a href="#">Company&nbsp;Name</a></h1>
    <cms:lay type="container" id="1" title="Hauptnavigation" />
</div> <!-- header -->

<div id="PATH">
    <cms:lay type="container" id="2" title="Pfad-Navigation" />
</div> <!-- PATH -->

<div id="SIDEBAR">
    <cms:lay type="container" id="3" title="Linke Spalte" />
</div> <!-- sidebar -->

<div id="CONTENT">
    <cms:lay type="container" id="4" title="Rechte Spalte - Hauptinhalt" />
</div> <!-- content -->

<div id="FOOTER">
<cms:lay type="container" id="5" title="Footer-Navigation" />
<p>&copy; 2005, Company Name. All rights reserved.</p>
<p>Designed by <a href="http://www.gettemplate.com/?fromnightcity">GetTemplate</a></p>
</div> <!-- footer -->

<cms:lay type="config" />
</body></html>
```

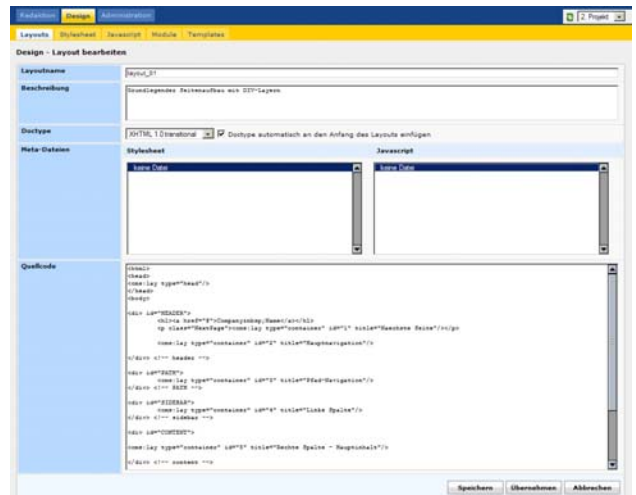
Wichtig: "<cms:lay type="head"/>" muss vor dem schließenden Head-Tag stehen. „<cms:lay type="config"/>" muss vor dem schließenden Body-Tag stehen.

Diese Mischung aus HTML-Quellcode und Sefrengo-Code fügt man in das Feld „Quellcode“ des Dialogfenster „Design > Layouts“ ein.

Des Weiteren muss ein Name für das Layout vergeben werden. Die Einstellungen für Doctype und die Beschreibung der Layout-Inhalte für spätere Erklärungen sind optional.

Wichtig: Die Doctype-Deklaration entscheidet, wie der Browser mit den Quellcode-Daten verfährt. Browser arbeiten hier in zwei Modi: Dem Quirks- und dem Standard-Modus. Die unterschiedliche Interpretation des Quellcodes hat oft ein unterschiedliches Aussehen zur Folge.

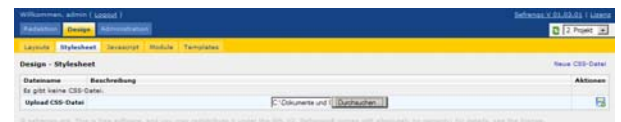
Wenn alle gewünschten Inhalte eingegeben sind bestätigt man mit „Speichern“. Für die korrekte Darstellung der Seite fehlt nun jedoch noch die CSS-Datei, die die Positionierungs- und Formatierungsinformationen der DIV-Layer enthält. Diese wird auch über den Layout-Dialog eingebunden. Bevor dies jedoch geschehen kann, muss die CSS-Datei erst in Sefrengo hoch geladen werden.



CSS-Dateien:

CSS hochladen

Über den Menüpunkt „Design > Stylesheet“ kann nun die CSS-Datei „styles.css“ vom eigenen Rechner ausgewählt und mit einem Klick auf das Diskettensymbol hoch geladen werden.

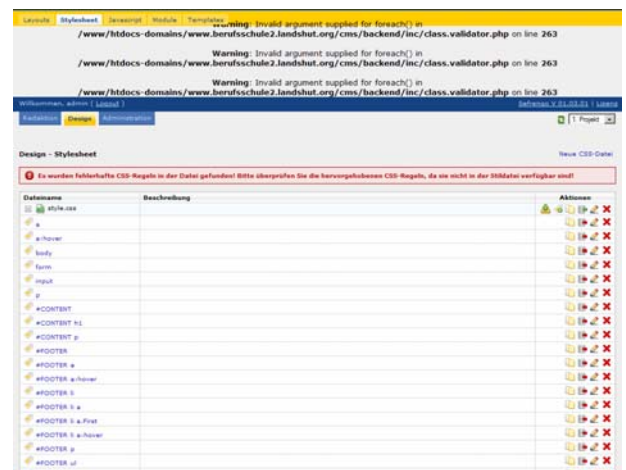


Kommt es hier zu der Fehlermeldung: „Datei konnte nicht auf den Server geladen werden.“ Liegt meistens ein Berechtigungsproblem vor. Kontrollieren Sie in diesem Fall die Dateiordnerrechte des Ordners „root/projekt01/cms/css“. Diese müssen auf 0777 eingestellt sein, damit das System die CSS-Datei speichern darf.

Wurde das CSS erfolgreich gespeichert, wird keine Fehlermeldung ausgegeben. Hat das CSS jedoch fehlerhafte Deklarationen, so weist der CSS-Parser von Sefrengo darauf hin:

Fehler im CSS

Im folgenden Fall sind in den rot gekennzeichneten IDs die „display“-Informationen doppelt vorhanden. Dies kann man mit einem Klick auf das Bearbeitungssymbol schnell beheben, indem man im folgend geöffneten Editor die doppelten Einträge löscht und die Regel anschließend „speichert“. Findet Sefrengo keine Fehler mehr, zeigt es die CSS-Regeln der Reihe nach ohne rote Markierung an.

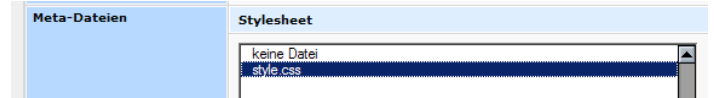


Entscheidend für dieses Verhalten ist die Einstellung unter „Administration > Projekte > Schraubenschlüssel-Symbol > Einstellungen CSS-Editor“. Hier kann man die Überprüfung der Regeln beeinflussen bzw. die Übernahme von Fehlerhaften Regeln ein- u. ausschalten.



CSS-Datei in Layout einbinden

Wurde das CSS ohne Fehler abgespeichert (alle Regeln werden ohne rote Markierungen dargestellt und es wird keine Meldung ausgegeben) muss die CSS-Datei im Layout eingebunden werden. Dazu wechseln Sie wieder zu „Design > Layout“ und wählen im Bereich „Meta-Dateien“ die CSS-Datei „style.css“ aus, indem diese mit der Maus markiert wird. Sie wird dann automatisch per <link rel ...> eingebunden.



Module

Nun wird es Zeit die CMS-Container des Layouts mit Modulen zu füllen. Diese Funktion findet sich im Menüpunkt „Design > Template“. Damit man im Template jedoch ein Modul für einen Container platzieren kann, muss man die Module zunächst in das System laden.

Deshalb wechselt man zunächst in das Menü „Design > Module“. Über den rechten oberen Link „Modul importieren“ kann ein mitgeliefertes Modul ausgewählt werden. Die in diesem Projekt verwendeten Module werden mit der Version 01.04.00 mit geliefert. Zunächst soll nun die Modulplanung für die Seite kurz skizziert werden:

Navigationmodule

Für dieses Tutorial sollen mehrere Navigationsmodule zum Einsatz kommen:

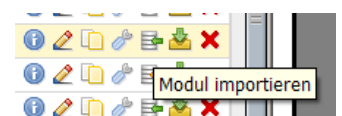
- Navigation Listen-Navigation 1.1 (2 mal)
- Navigation Navigation 2.01.00
- Navigation URHere 1.1.1

Inhaltsmodule für die linke und rechte Spalte

Die linke Seite soll mit zunächst mit Textfeldern befüllt werden und der Hauptbereich, die rechte Spalte wird mit dem Modul „Contentflex“. In einem 2. Template soll später die linke Spalte mit einer vertikalen Navigation versehen werden, damit auch Unterordner in dieser zusätzlichen Navigation erreichbar und die Seitenstruktur damit flexibler wird.

Module importieren

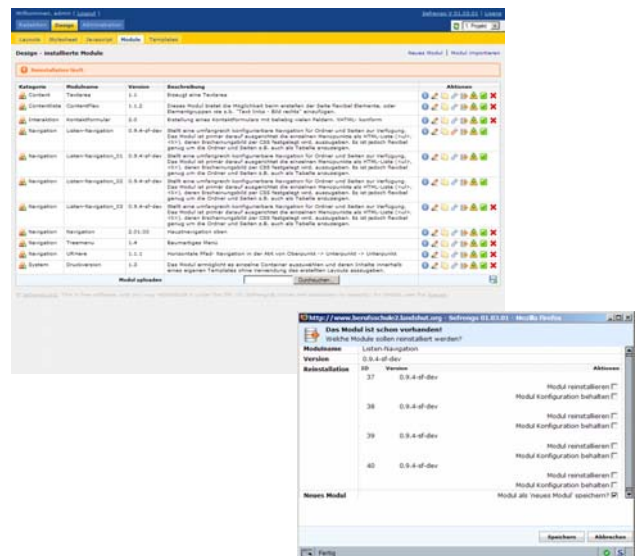
Alle im Tutorial verwendeten Module werden mit Sefrengo schon mit geliefert und können über „Design > Module > Modul importieren“ in das Projekt übernommen werden: Importieren Sie über den Button „Modul importieren“ die Module "Contentflex", "Textarea", "URHere", "Listen-Navigation" und "Navigation"



Module installieren

Falls Sie weitere Module benötigen, können Sie diese auf www.sefrenco.org herunterladen und über „Modul uploaden“ installieren. Beachten Sie hierbei bitte, dass Ihre Browsereinstellungen Pop-up-Fenster erlauben. Für den Fall, dass eine neue Modulversion upgedatet wird, können Sie in dem Pop-up-Fenster dazu Einstellungen vornehmen

Falls Sie einen Popup-Blocker installiert haben, deaktivieren Sie diesen für Ihr Sefrenco-Projekt um alle Einstellungen vornehmen zu können. Bei diesem Beispiel sehen Sie das Pop-up-Abfrage-Fenster bei mehreren Listennavigationen:



Die Listennavigation kann mehrmals in einem Template verwendet werden. Ein mehrfaches Abspeichern dieses Modul ist – im Vergleich zur Vorgängerversion – nicht mehr notwendig.

Installierte Module

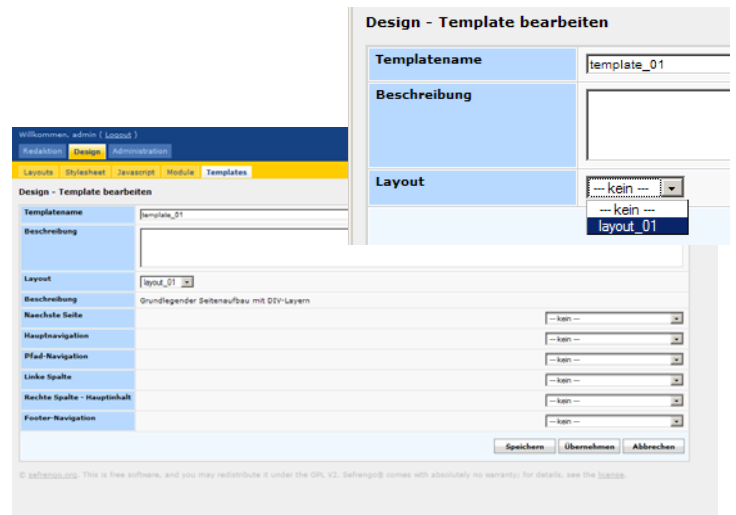
In dem Modulbereich sollten nun folgende Module angezeigt werden (die duplizierte und die originale Modulversion der Listennavigation werden mit Nummern 01 und 02 unterschieden):

- Textarea 1.1
- ContentFlex 1.4
- Listen-Navigation 1.1
- Navigation URHere 1.1.1
- Navigation 2.01.00



Template anlegen

Über „Design > Templates > Neues Template“ wird eine leeres Template geöffnet. Das Template wird im Feld „Layout“ das vorher angelegte Layout mit dem Namen „layout_01“ ausgewählt. Anschließend zeigt Sefrengo für jeden CMS-Container, der im Layout steht, eine Möglichkeit an, ein Modul auszuwählen. Hat man die Module – wie zuvor – noch nicht installiert, kann man keine auswählen. Dieser Schritt ist also zwingend notwendig.



Die Benennung der CMS-Container im Layout macht hier eine klare Zuordnung leichter.

Es werden jetzt die Module für die jeweiligen Bereiche ausgewählt.

- Hauptnavigation > Listennavigation
- Pfad-Navigation > URHere
- Linke Spalte > Textarea
- Rechte Spalte – Hauptinhalt > Contentflex
- Footer-Navigation > Listennavigation

Das Template wird mit „Speichern“ – zunächst ohne weitere Konfigurationen – abgespeichert.

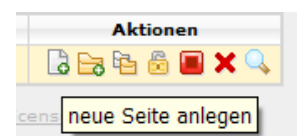
Um die Konfiguration der einzelnen Module anschaulicher zu machen, legt man zu diesem Zeitpunkt die Ordnerstruktur und die Seiten fest. Man kann sich die Auswirkungen der Modulkonfigurationen dann direkt ansehen.

Ordner und Seiten anlegen

Über „Redaktion > Seiten > Aktionen > Neuen Ordner anlegen“ werden jetzt die Hauptordner angelegt. Im Feld „Template“ wählt Sefrengo automatisch das erste erkannte angelegte Template „template_01“ aus und zeigt es auch für diesen Ordner an. Die Ordner erhalten die Namen, die später in der Hauptnavigationleiste angezeigt werden sollen.

Zum grundlegenden Verständnis: Der Ordner ist nun mit dem Template verknüpft. Alle Einstellungen, die im Folgenden im Template vorgenommen werden, müssen ganz am Ende mit dem Kontrollkästchen „Änderungen in den Templatekopien für die Ordner und Seiten übernehmen“ abgespeichert werden. Sonst werden die Änderungen für dieses im Hauptordner verankerte Template nicht übernommen. Dieses Vorgehen ist jedoch nur am Anfang, bei der grundlegenden Seitenanlegung sinnvoll. Es kann später auch vorkommen, dass man spezielle Seiten/Ordner zwar mit dem gleichen Grundtemplate verknüpft hat, aber unterschiedliche – auf den jew. Ordner/Seite angepasste – Konfigurationen abspeichern möchte. Für diesen Fall kann jeder Ordner mit seinem eigenen Template konfiguriert werden.

Innerhalb der Ordner werden jeweils noch zusätzlich eine Seite angelegt, die dann als Inhalt des Ordners angezeigt werden. Ein Ordner muss eine Seite beinhalten, da ein Ordner alleine nicht angezeigt werden kann. Es fehlt ihm an Inhalten. Diese stellt eine Seite zur Verfügung.

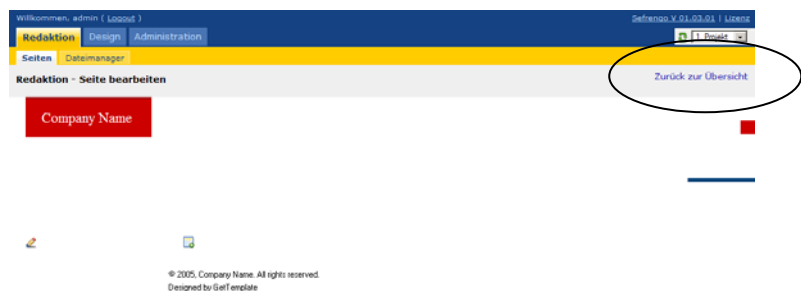


Ordner- und Seitenstruktur im Überblick

- Home (Ordner)
 - Home (Seite)
 - Projekte (Ordner)
 - Projekte (Seite)
 - Projekt 1 (Ordner)
 - 1. Projekt Seite 1 (Seite)
 - 1. Projekt Seite 2 (Seite)
 - Projekt 2 (Ordner)
 - 2. Projekt Seite 1 (Seite)
 - Projekt 3 (Ordner)
 - 3. Projekt Seite 1 (Seite)
 - 3. Projekt Seite 2 (Seite)
 - 3. Projekt Seite 3 (Seite)
 - Über uns (Ordner)
 - Über uns (Seite)
 - Kontakt (Ordner)
 - Kontakt (Seite)
 - Sitemap (Ordner)
 - Sitemap (Seite)

Am Ende einer Seitenkonfiguration steht standardmäßig die Einstellung: „Ordner-template“. Diese kann natürlich auch abgewandelt werden. Für dieses Beispiel soll hier jedoch zentral das für den Ordner vergebene Template verwendet werden.

Beim Seiten anlegen springt Sefrengo automatisch nach dem „Speichern“ zu der Seitenansicht. Es fällt auf, dass im Vergleich zu dem vordefinierten Design keine Hintergrundgrafiken angezeigt werden. Legen Sie jedoch trotzdem alle Seiten wie angegeben an, indem Sie mit dem Befehl „Zurück zur Übersicht“ zur Ordneransicht in „Redaktion > Seiten“ zurück wechseln.

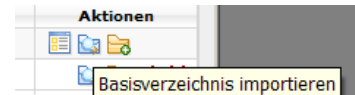


Bevor nun die Module im Template konfiguriert werden, soll nun eine kurze Fehleranalyse Aufschluss über den Verbleib der Hintergrundgrafik geben.

Grafiken hochladen

Zunächst einmal wurden noch gar keine Grafiken, die Sefrengo für die Verwendung als Hintergrundgrafiken verwenden kann, hochgeladen. Dies soll nun im Menü „Redaktion > Dateimanager“ nachgeholt werden.

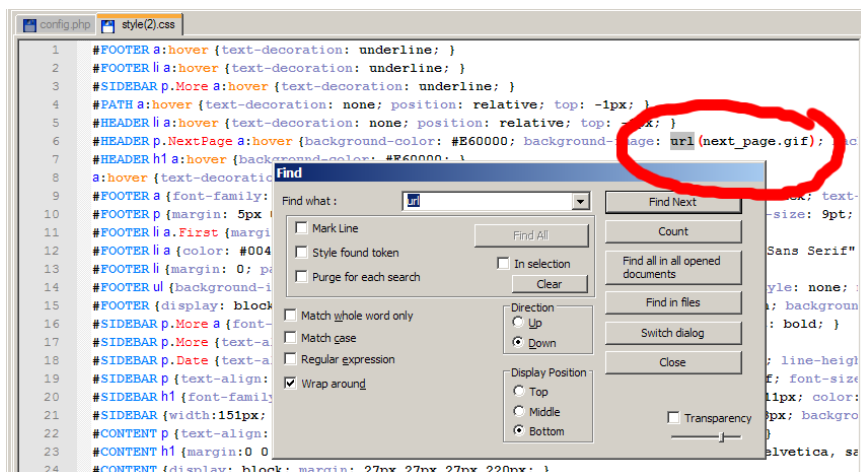
Falls Sie den Dateimanager das erste mal öffnen, müssen Sie noch das Basisverzeichnis importieren, damit ein Abgleich mit der Datenbank vorgenommen wird. Laden Sie alle Grafiken, die mit dem vordefinierten Design mitgeliefert werden in den Ordner „/img“.



Es muss aber noch ein zweiter Schritt unternommen werden, um die Grafiken für Sefrengo zugreifbar zu machen. Die CSS-Datei enthält Pfadangaben für die Hintergrundgrafiken. Diese müssen nun auf Sefrengo angepasst werden.

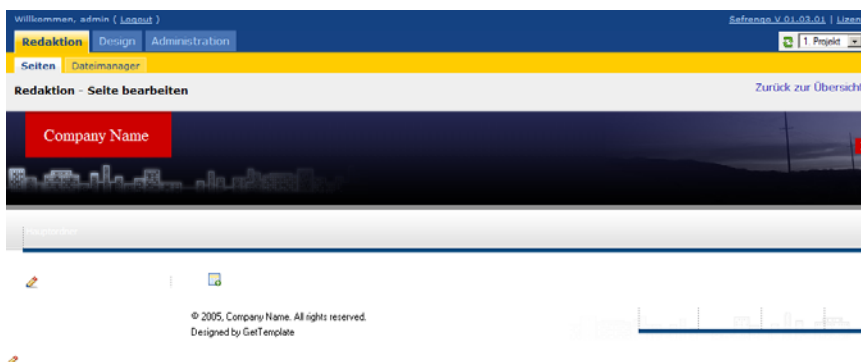
Hier wäre es möglich jede einzelne Regel in Sefrengo zu öffnen und einzeln zu editieren. Einfacher ist es, man lädt das (vorher ja reparierte file) lokal auf den Rechner, öffnet es mit einem Editor und bearbeitet es komplett. Wenn alle Pfad angepasst wurden kann man das CSS-File wieder in Sefrengo hoch laden.

Mit dem Editor sucht man alle „url“-Einträge und schreibt davor „.././media/img“. Der Grund: Die CSS-Datei liegt im Ordner „root/projekt01/cms/css“. Die Hintergrundgrafiken, auf die vom CSS aus verwiesen werden soll jedoch im Ordner „root/projekt01/media/img“. Mit dieser Pfadänderung weist man Sefrengo also an, zweimal in den übergeordneten projekt01-Ordner und anschließend zu media/img zu wechseln.



Hilfreich ist hier die Suchen und Ersetzen-Funktion des Editors (meist STRG + H). Sucht man nach der Zeichenfolge „url(“ kann man diese zu „url(.././media/img/“ automatisch ersetzen lassen und übersieht nichts.

Wechselt man nun in die Seitenansicht, indem man über „Redaktion > Seiten“ auf die jew. Seite klickt, werden die Hintergrundgrafiken angezeigt. Die Bearbeitungssymbole der einzelnen Module (Textfeld, Contentflex) werden auch gezeigt. Aber die Navigationsinhalte – die Seiten selbst – werden noch nicht dargestellt. Nun ist es Zeit, das Template mit den einzelnen Modulen zu konfigurieren.



Template konfigurieren

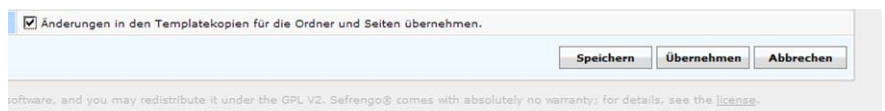
Wechselt man über „Design > Templates > template_01“ zu dem vorher angelegten Basistemplate werden die eingefügten Module nun in Ihrer Standardkonfiguration gezeigt.

Listennavigation für Hauptnavigation konfigurieren

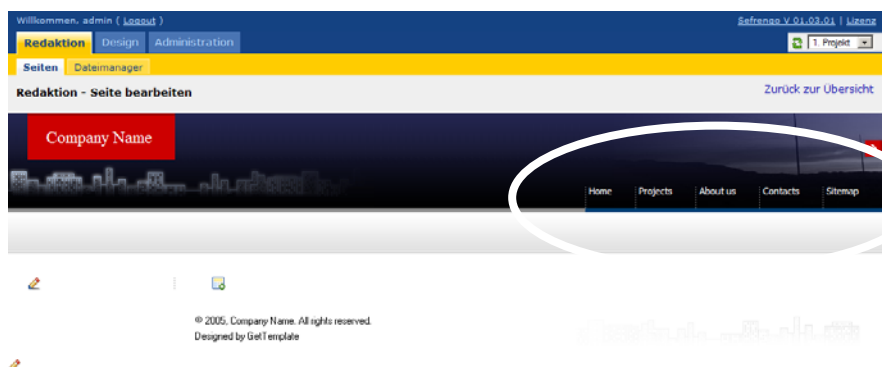
Die erste Listennavigation wird nun mit folgenden Einstellungen konfiguriert:

- Konfigurationsmodus: Erweitert
- Startordner: Home
- Anzeigeart: Ordner mit Seiten
- Sortierung: Standard (wie Backend); aufsteigend
- Erst Ordner, dann Seiten anzeigen: nein
- Bei Ordnerlinks immer Link zur Startseite des Ordners generieren (statt Link zum Ordner): ja
- Komplette Ordnerstruktur anzeigen: nein
- Anzahl darzustellender Ebenen (0: alle): 1
- Startseiten anzeigen: ja
- Ordner nur als aktiv markieren, wenn Startseite angezeigt wird: nein
- Pfad als aktiv verfolgen: nein
- Ordner ausschließen (idcat kommasepariert angeben): leer
- Seiten ausschließen (idcatside kommasepariert angeben): leer
- Ausgeschlossene Seiten und Ordner auch im Backend verbergen: nein
- Einleitung/Ende der Navigation anzeigen, falls Navigation "leer": nein
- Startnummer für tabindex: leer

Bevor man das Template nun speichert, muss das Kontrollkästchen für die Übernahme in das Ordner-template aktiviert werden:



In der Seitenansicht sieht man folgenden Effekt: Die Seiten des Ordners „Hauptordner“ werden nun angezeigt.



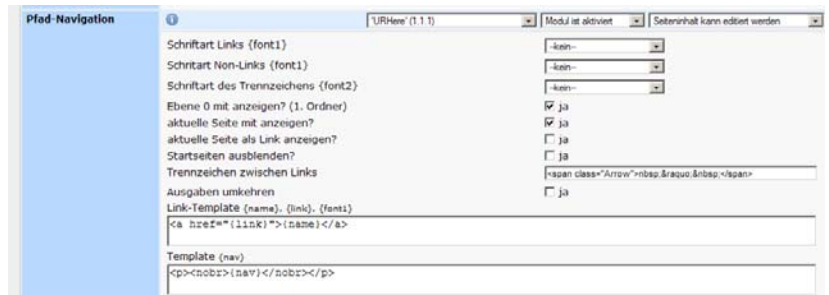
URHere für die Pfadnavigation konfigurieren

Die Pfadnavigation wird durch ein `<p>`-Tag im CSS positioniert. Dies wurde bisher nicht berücksichtigt, da sowohl im Layout

```
<div id="PATH">
  <cms:lay type="container" id="2" title="Pfad-Navigation"/>
</div> <!-- PATH -->
```

als auch im Modul selbst der `<p>`-Tag noch nicht eingesetzt wurde. Dies kann auf zwei Arten nachgeholt werden:

1. Man verschachtelt den CMS-Container: `...<p><cms:lay type="container" id="2" title="Pfad-Navigation"/></p>` und weist ihm dadurch das Aussehen und die Positionierung des Tags `<p>` innerhalb des DIV-Layers "PATH" zu, oder



2. Man stellt diese Verschachtelung im Template innerhalb der Ausgabeeinstellung „Template“ des Moduls „URHere“ ein:

Der Platzhalter `{nav}` steht hier für die Ausgabe des Inhalts, der in dem Feld darüber „Link-Template“ definiert wird.

Dieses `{nav}` wird komplett (auch mit den schon vordefinierten `<nobr>`-Tags) von `<p>...</p>` umschlossen.

CSS in einem Template verknüpfen

Es kann ein CSS wie im Beispiel des `<p>`-Tags, das über die Klasse `.p` der ID `#PATH` definiert wird (siehe CSS-Datei unter „Design > CSS“) direkt in das Template über einen umgebenden Tag und dem zugehörigen „`class=“...“` bzw. „`style=“...“`“ eingegeben werden.

Z. B. kann die Klasse „Arrow“ der ID „PATH“ (siehe CSS-Datei) direkt in einem ``-Tag auf das Trennzeichen angewendet werden: Dazu schreibt man in das Feld „Trennzeichen zwischen Links“ „`nbsp;» `“. Die Leerzeichen dienen lediglich der Abtrennung der Links zu dem Trennzeichen.

Eine weitere und vor allem die zentrale Möglichkeit, CSS-Klassen auf Modulelemente im Template anzuwenden, ist die direkte Auswahl der `{font}`-Platzhalter. Hier können Klassen aus dem CSS direkt mit dem benannten Platzhalter (`{font1...}`) belegt werden. Weiteres hierzu im nächsten Abschnitt „Textfelder konfigurieren“.

Da die CSS-Styles in diesem Beispiel direkt eingegeben werden, kann der Befehl „`class="{font1}`“ aus dem Feld „Link-Template“ gelöscht werden.

Das Template beschreibt nun das Modul „URHere“ wie folgt:

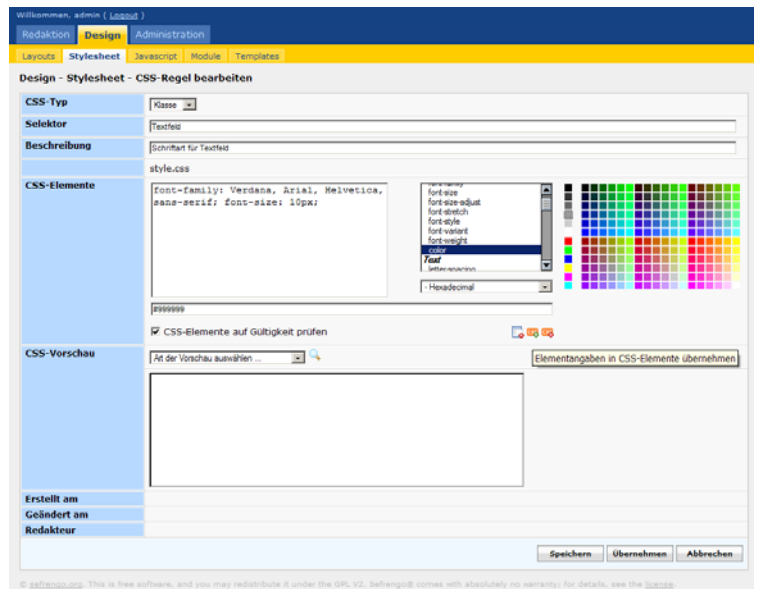
- Schriftart Links {font1}: kein
- Schriftart Non-Links {font1}: kein
- Schriftart des Trennzeichens {font2}: kein
- Ebene 0 mit anzeigen? (1. Ordner): ja
- aktuelle Seite mit anzeigen?: ja
- aktuelle Seite als Link anzeigen?: nein
- Startseiten ausblenden?: ja
- Trennzeichen zwischen Links: `nbsp;» `
- Ausgaben umkehren: nein
- Link-Template: `{name}`
- Template {nav}: `<p><nobr>{nav}</nobr></p>`

Modul Textfeld konfigurieren

Die linke Spalte der Internetseite soll ein Textfeld enthalten. In dem Modul „Textarea“ soll nun die Schriftart zusätzlich über eine neue Klasse definiert werden. Dazu erstellt man im Menü „Design > CSS > Styles.css öffnen > Button Neue CSS-Regel“ eine CSS-Klasse mit dem Selektor „Textfeld“ o. Ä. und legt in dieser die Schriftart, die Farbe und Größe der Schrift fest. In diesem Beispiel wurden folgende Regeln definiert:

```
font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color: #999999;
```

Wichtig: Die Regel muss als Klasse definiert werden, da nur Klassen über den {font1}-Platzhalter eingebunden werden können!



Diese neu angelegte Klasse wird nun über die Auswahl des Felds „Schriftart“ ausgewählt. Im Modul „Textfeld“ wird diese Klasse automatisch als CSS verwendet. Im Gegensatz zu dem folgenden Modul muss hier kein zusätzlicher Platzhalter innerhalb einer „class=“...“-Beschreibung angegeben werden.

Modul ContentFlex konfigurieren

Um die unterschiedlichen Arten der Einbindung von CSS noch einmal im Überblick zu zeigen, werden in der folgenden Modulkonfiguration mehrere Arten angewendet:

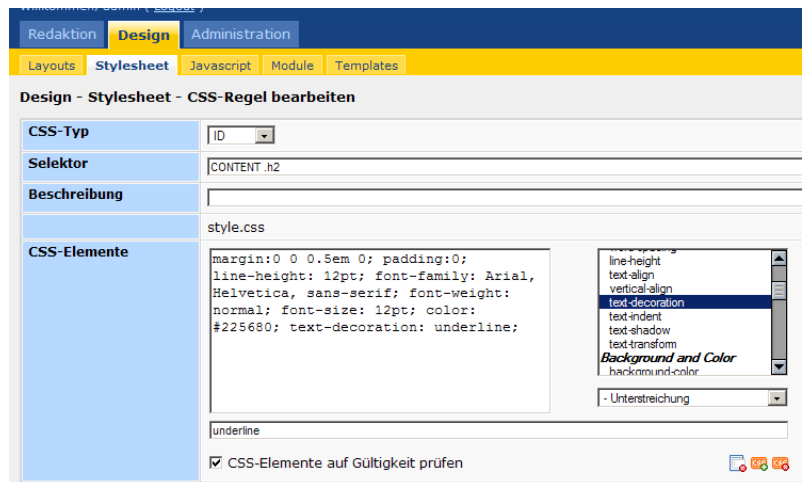
Verwendung der bereits bestehenden CSS-Klassen innerhalb der ID „CONTENT“

CSS-Regeln anlegen

Das vorgefertigte Design bringt für die einzelnen DIV-Layer-Bereiche schon vordefinierte CSS-Klassen mit. Die Regel für das „h1“-Tag innerhalb der ID „CONTENT“ kann direkt für das ContentFlex-Element „Titel“ verwendet werden.

Eine Regel für das Tag „h2“ fehlt jedoch. Um Untertitel über das Tag „h2“ unterschiedlich anzeigen zu können, muss diese Regel noch erstellt werden. Im Menü „Design > CSS >

Styles.css öffnen > Button Neue CSS-Regel“ wird eine neue Regel für den Typ „ID“ mit der Benennung „CONTENT h2“ erstellt. Über diese Typ-Bezeichnung wird innerhalb des DIV-Layers „CONTENT“ – in dem sich ja das Modul ContentFlex befindet – das „h2“-Tag ausschließlich für diese ID definiert und dadurch nur für das darin befindende Modul verwendet.



Kurzbeschreibung der neuen Regeln:

- CSS-Typ: ID
- Selektor: CONTENT h2
- CSS-Elemente: margin:0 0 0.5em 0; padding:0; line-height: 12pt; font-family: Arial, Helvetica, sans-serif; font-weight: normal; font-size: 12pt; color: #225680; text-decoration: underline;

Um die unterschiedliche Einbindung von CSS-Regeln weiter darstellen zu können, legen Sie bitte noch zusätzlich die folgende Regel fest (Diese Klasse wird später als normaler Text in ContentFlex eingebunden):

- CSS-Typ: Klasse
- Selektor: Text
- CSS-Elemente: text-indent: 25px; margin: 0 0 10px 0; padding: 0;

Im Template stellen Sie nun im Modul ContentFlex Folgendes ein:

- Konfigurationsmodus: Alles > Register "Spez. Einstellungen"
 - Schriftart {font1}: Text
 - Schriftart {font2}: kein
- Konfigurationsmodus: Alles > Register Element-Templates:
 - Element 1; Benennung: Titel; Bearbeitbar: Für alle Gruppen; Umschließendes Template: Keine umschließenden Templates; Inhalt: <h1>{text:1}</h1>
 - Element 2; Benennung: Untertitel; Bearbeitbar: Für alle Gruppen; Umschließendes Template: Keine umschließenden Templates; Inhalt: <h2>{text:1}</h2>
 - Element 3; Benennung: Text eingerückt; Bearbeitbar: Für alle Gruppen; Umschließendes Template: Keine umschließenden Templates; Inhalt: <p>{textarea:1}</p>

- Element 4; Benennung: Text normal; Bearbeitbar: Für alle Gruppen; Umschließendes Template: Keine umschließenden Templates; Inhalt: `{textarea:1}`
- Alle folgenden Elemente mit „Element deaktivieren“ ausschalten.

Erläuterung:

- Im Element 1 greift das Tag „h1“ auf die CSS-Einstellungen von „CONTENT h1“ zu.
- Im Element 2 wird das gleiche Prinzip angewendet: Hier wird die vorher angelegte Regel für „h2“ verwendet.
- Element 3 bezieht sich auf die Einstellungen des Tags „<p>“ innerhalb von CONTENT und beinhaltet eine Formatierung der 1. Zeile (siehe CSS-Regel). Diese Einrückung wird in der vorher angelegte Klasse „Text“ nicht verwendet.
- Element 4 bezieht sich auf „font1“. Der Platzhalter wird oben in der Menükonfiguration mit der Klasse „Text“ belegt. Um diese Klasse verwenden zu können, muss hier noch ein Zusatztag (hier „“) mit der Klasse „class=“font1“ belegt werden.

Leertexte zum Testen in die Seite einfügen

Wechseln Sie in die Seitenansicht und fügen Sie Leertexte ein, um die einzelnen Elemente der bisher konfigurierten Module auszuprobieren. Verwenden Sie die Bearbeitungssymbole der einzelnen Module (Textfeld und Contentflex) um die Inhalte einzugeben. Die Seite könnte dann so aussehen:

The screenshot shows a web editor interface for 'Redaktion - Seite bearbeiten'. The page content includes:

- A header with 'Company Name' and navigation links: Home, Projects, About us, Contacts, Sitemap.
- A breadcrumb trail: Home » Home.
- A sidebar with a 'Titeltext' section containing a text field with placeholder text and a sub-section 'Untertitel Test' with a text field.
- Main content area with several text blocks:
 - A paragraph of placeholder text with a blue link 'Untertitel Test'.
 - A paragraph of placeholder text with an indented first line.
 - A paragraph of placeholder text.
- A context menu is open over the third text block, showing options: 'Nach Element 5 (Text normal) einfügen', 'Titel', 'Untertitel', 'Text eingerückt', and 'Text normal'.
- Footer text: '© 2005, Company Name. Designed by GetTemplate'.

CSS-Regeln für Navigation vorbereiten

Zunächst legt man sich jedoch noch eine neue CSS-Regel für den Bereich „SIDEBAR a“ an, damit die Links nicht – wie allgemein definiert – rot und unterstrichen dargestellt werden:

- CSS-Typ: ID
- Selektor: SIDEBAR a
- CSS-Elemente: font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; color: #333333; text-decoration: none;

Um zusätzlich aktuelle Seitendarstellungen in der Navigation – die ja dann keine Links sind, da die Seite aktuell geöffnet ist – den Links anzupassen, erstellt man eine Klasse mit dem Namen „Navigation“ und kopiert die gleichen Informationen aus „SIDEBAR a“ hinein.

Konfiguration des Moduls Navigation

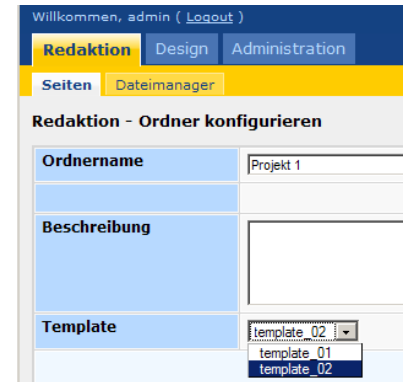
Wechseln Sie dann in das Template “template_02”, öffnen Sie die Erweiterte Konfiguration des Moduls “Navigation” und stellen Sie folgende Konfigurationselemente ein:

- Konfigurationsmodus: Erweitert
- Verzeichnis wählen: Projekte
- anzeigen von: Ordner mit Seiten
- Darstellungsform: Unterordner automatisch anzeigen
- Startseiten anzeigen: ja
- Seiten sortieren nach: Standard (wie backend)
- Anzahl darzustellender Ebenen: alle
- Pfad als aktiv verfolgen: nein
- Startseite bei {page_count} mitzählen?: nein
- Ordner ausschliessen (idcat kommasepariert angeben): leer
- Seiten ausschliessen (idcatsides kommasepariert angeben): leer
- Schriftart {font1}: Navigation
- Schriftart {font2}: Navigation
- Schriftart {font3}: kein

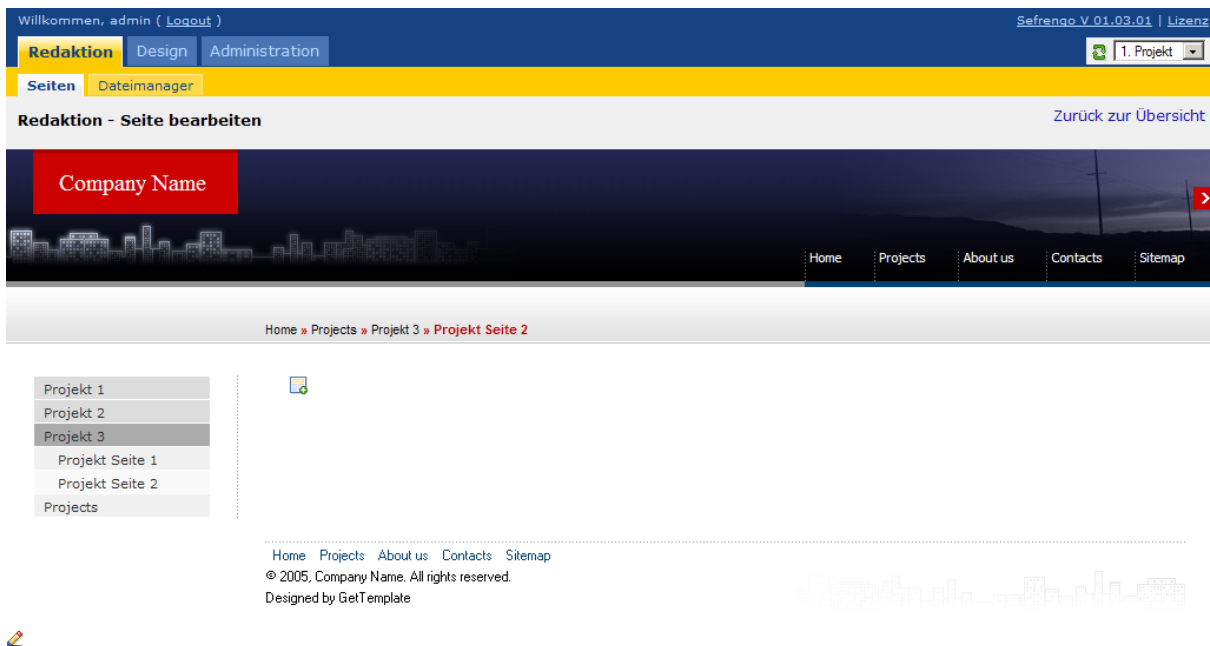
- Templates:
In diesem Bereich tauscht man für dieses Beispiel lediglich die Farben aus, dass sie zum Gesamtdesign passen:
 - #A8BADE wird zu #dcdcdc
 - #5A7BAD wird zu #aaaaaa
 - #C7D5EB wird zu #f0f0f0
 - #DBE3EF wird zu #f9f9f9

Template den jew. Ordnern zuweisen

Dieses neue Template "template_02" muss nun den Ordnern "Projects" und allen seinen Unterordnern zugewiesen werden. Dazu klickt man in der Ansicht „Redaktion > Seiten“ auf das Ordnersymbol und wählt im folgenden Dialog das neue Template aus. Ab sofort werden die so definierten Ordner und deren Seiten mit dem neuen Template ausgegeben.



Die Seiten innerhalb des Projects-Ordnern sehen dann in etwa so aus:



Schlusswort

Das Tutorial endet an dieser Stelle. Ich hoffe, dass Ihnen nun die grundlegenden Verfahren in Sefrenco etwas näher gebracht wurden und Sie die Gesamtstruktur und Arbeitsweise von Sefrenco für Ihrer Projekte besser nutzen können.