



SEBLOD<sup>®</sup>  
by Octopus

Axel Tüting

# SEBLOD Handbuch

Mehr als nur das Standard-Joomla!  
Das Content Construction Kit SEBLOD<sup>®</sup> in der Praxis.

- Installation, Konfiguration und Erweiterung mit bestehenden Apps
- Alles für die eigene App: Felder, Formulare, Inhaltstypen, Listen und Suchtypen
- Profiwissen: Beziehungslisten, Module, Multisiten-Plug-ins und Templates

# Kein Vorwort

Vorworte werden nicht gelesen. Vermutlich wegen der ganzen Dankesworte, die dort stehen und niemanden interessieren. Dabei stehen zumindest in meinen Vorworten auch immer die wichtigen Dinge: für wen dieses Buch geeignet ist, und für wen nicht. Oder welche Voraussetzungen es gibt, um mit diesem Buch effektiv arbeiten zu können. Dass die allerwenigsten das lesen, sieht man spätestens in den Kritiken auf Amazon & Co., wo man sich darüber beschwert, dass man nicht die Voraussetzungen für das Buch hätte und dass man das ja irgendwo hätte hinschreiben können. Und dann schaue ich zur Sicherheit noch einmal ins Vorwort. Aber lassen wir das, verzichten wir auf das ungelesene Vorwort und kommen wir gleich zur Sache.

Wer von Joomla! keine Ahnung hat, kann mit diesem Buch nichts anfangen. Wer glaubt, er müsse nicht tiefer einsteigen und es reiche, Joomla! mal eben zu installieren, um dann tolle Sachen mit SEBLOD machen zu können, kann mit diesem Buch nicht allzu viel anfangen. Wer Joomla! zumindest rudimentär kennt und weiß, wie man Beiträge anlegen und Menüs machen kann, der kann mit SEBLOD mehr machen und das eine oder andere herausholen, was bisher nur mit Programmierkenntnissen möglich war. Wer sich mit Joomla! richtig gut auskennt, wird mit SEBLOD und diesem Buch Dinge machen können, die mit Joomla! bislang außerhalb der Möglichkeiten lagen.

SEBLOD ist eines der wenigen *Content Construction Kits* (CCK), die diesen Namen wirklich verdienen. Mehr noch: mit SEBLOD können Dinge erstellt werden, die bislang nur den Programmierern von mehr oder weniger aufwendigen Erweiterungen vorbehalten waren. Mit SEBLOD ist es oftmals möglich, in deutlich weniger Zeit komplexe Erweiterungen zu erstellen als mit der Programmierung.

Aber alles hat seinen Preis. SEBLOD ist sehr komplex und demzufolge auch nicht ganz so einfach zu durchschauen. Ein erstes Formular mit SEBLOD zu entwickeln wird hingegen verhältnismäßig schnell gehen. Komplexe Apps zu erstellen braucht seine Zeit, bis die vielen Einstellungsmöglichkeiten durchschaut und angewendet werden können. SEBLOD-Kenner sprechen davon, dass ca. 70 % aller Erweiterungen in Joomla! nicht mehr benötigt werden, da sie mit SEBLOD abbildbar sind.

Wer sich bereits mit SEBLOD befasst hat, wird aber auch wissen, dass Einstiegsliteratur mittlerweile in allen Sprachen recht verbreitet ist, weiterführende Literatur ist hingegen spärlich bis gar nicht vorhanden. Somit kann man dieses Buch durchaus auch als Pionierarbeit betrachten. Hier werden viele Dinge genannt, die nirgendwo niedergeschrieben sind und nur durch Ausprobieren und Erfahrungsaustausch mit anderen gefunden werden konnten.

Die Arbeit an diesem Buch hat lange gedauert. Sehr lange. Auch ich bin irgendwann ein Opfer der spärlichen Dokumentation geworden. Die Hilfe des französischen

SEBLOD-Teams war stets da, aber Terminkalender und meine mangelnde Kenntnis der englischen Sprache forderten auch ihren Tribut. Manchmal habe ich sogar russische Videos mit kyrillischer Schrift studiert in der Hoffnung, dass mir Klickwege Ziele verraten, zumindest ein paar mögliche Ideen geben könnten.

Auch wenn dies kein Vorwort ist, so gehören doch ein paar Dankesworte an unermüdete Helfer, Zuhörer und Ideengeber hinein.

Da ist zuallererst natürlich das SEBLOD-Entwicklerteam – das Team *Octopos* – zu nennen und dort mein Ansprechpartner Sebastien Lapoux.

Ein ganz besonderer Dank gebührt Paul Brand, der sich an der sonnigen Küste Spaniens mit den Tiefen von SEBLOD befasst und nicht nur die deutsche Übersetzung geschrieben hat, sondern auch ein immenses Wissen über SEBLOD hat und mich ein wenig daran teilhaben ließ.

Screendesign Whykiki kennen die einen oder anderen sicherlich von seinen tollen Videos, nicht nur zu SEBLOD. Mit bürgerlichem Namen heißt er Christian Schilling und hat ein Gastkapitel in diesem Buch bekommen. Sie finden es ab Seite 207.

Christian Linsner, der so etwas wie ein Stammtestleser geworden ist. Was dieser Mensch alles findet, ist enorm! Nie wieder ein Buch ohne ihn!

Neco Papakci von Pixelkultur hat sich durch etliche Kapitel gequält und nicht mit Anmerkungen, Tipps und Tricks gegeizt.

Dr. Markus Stäuble vom Franzis Verlag gebührt ein ganz besonderer Dank. Seine Geduld wurde auf eine harte Probe gestellt, und dennoch hat er stets an mir festgehalten und mich unterstützt!

# Inhaltsverzeichnis

Teil I – Einstieg in SEBLOD .....	13
<b>1 Was ist SEBLOD?.....</b>	<b>15</b>
1.1 Was ist ein CCK?.....	15
1.2 Was ist mit SEBLOD möglich?.....	16
1.3 Versionen.....	17
<b>2 Der schnelle Überblick.....</b>	<b>19</b>
2.1 Installation.....	19
2.2 Apps downloaden und installieren .....	20
2.3 SEBLOD spricht deutsch .....	21
2.4 Grundsätzliche Funktionsweise.....	21
2.4.1 App-Folder .....	22
2.4.2 Formulare und Inhaltstypen.....	23
2.4.3 Felder .....	23
2.4.4 Listen & Suchtypen .....	23
2.4.5 Templates.....	24
<b>3 Der detaillierte Blick .....</b>	<b>27</b>
3.1 Die erste App: Newsanzeige.....	28
3.1.1 Einfache Anlage im App-Verzeichnis .....	29
3.1.2 Direkt zu den Feldern.....	32
3.1.3 Die Felder im Überblick.....	41
3.1.4 Eine Form muss her.....	41
3.1.5 Ein- und Ausgabe einer News.....	50
3.2 Die News erweitern.....	52
3.2.1 Haupteintrag und Kategorie.....	52
3.2.2 Bilder sagen mehr als tausend Worte.....	54
3.2.3 Mehr Bilder mit Feld X.....	57
3.2.4 Abhängige Bilder .....	59
<b>4 Die Speicherung und ihre Hürden.....</b>	<b>61</b>
4.1 Ein Blick hinter die Kulissen: MySQL.....	61
4.2 Speicherformat: Standard .....	63
4.3 Speicherformat: Benutzerdefiniert .....	65
4.4 Speicherformat: Json.....	67
4.5 Speicherformat: Kein.....	68
4.6 Speicherposition .....	68
4.7 Datentyp wechseln.....	70
4.8 Eigene Tabelle erstellen.....	70
4.9 CSV-Import .....	73

Teil 2 - Die Schaltzentrale .....	77
5 App-Optionen.....	79
6 ACL in der APP .....	81
6.1 Der ACL-Generalschlüssel.....	83
7 Optionen.....	85
7.1 Allgemein.....	85
7.2 Komponenten.....	88
7.2.1 App-Folder .....	88
7.2.2 Konstruktion .....	88
7.2.3 Entwicklung.....	89
7.2.4 Versionen.....	89
7.3 Integration .....	91
7.4 Sprache.....	94
7.5 Media.....	95
7.6 Multisite .....	95
7.7 Berechtigungen .....	96
7.8 Seite.....	96
7.8.1 Site (Frontend).....	96
7.8.2 Scripts & Styles .....	98
8 Alle Felder .....	99
8.1 JavaScript und CSS .....	100
8.2 Auswahl .....	101
8.2.1 Dynamisch .....	101
8.2.2 Einfach .....	104
8.2.3 Mehrfach .....	105
8.2.4 Zahlen .....	106
8.3 Button .....	106
8.3.1 Senden.....	106
8.4 Captcha.....	110
8.4.1 Mathematisch .....	110
8.5 Formular .....	110
8.5.1 E-Mail.....	110
8.5.2 Checkbox .....	110
8.5.3 Passwort.....	113
8.5.4 Radiobutton.....	113
8.5.5 Text.....	113
8.5.6 Versteckt.....	113
8.6 HTML.....	113
8.6.1 div.....	113
8.6.2 Icon.....	114
8.6.3 Tabs.....	114
8.7 Inhalt .....	117
8.7.1 Freier Text.....	117
8.7.2 Link.....	117
8.8 Joomla-Bibliothek (JForm) .....	118
8.8.1 Assoziationen.....	118

8.8.2	Benutzer .....	119
8.8.3	Benutzergruppen .....	119
8.8.4	Editoren .....	119
8.8.5	Elterngruppe .....	119
8.8.6	Hilfeseiten .....	119
8.8.7	Inhaltssprache .....	119
8.8.8	JForm - Kategorie .....	120
8.8.9	JForm - Zugangslevel .....	120
8.8.10	Komponenten Layout .....	121
8.8.11	Medien .....	121
8.8.12	Menüeintrag .....	122
8.8.13	Regeln .....	122
8.8.14	Schlagwörter .....	122
8.8.15	Template: Stil .....	123
8.8.16	Zeitzone .....	123
8.9	Joomla! .....	123
8.9.1	Artikel (Ähnliche) .....	124
8.9.2	Module .....	125
8.10	Kollektion/Sammlung .....	127
8.10.1	Gruppe .....	127
8.10.2	Feld X .....	127
8.10.3	Gruppe X .....	128
8.11	Picker .....	129
8.11.1	Farbauswahl .....	129
8.11.2	Kalender .....	129
8.12	Textarea .....	134
8.12.1	Textbereich .....	134
8.12.2	WYSIWYG-Editor .....	134
8.13	Upload .....	135
8.13.1	Datei .....	135
8.13.2	Image .....	137
8.14	Video .....	137
8.14.1	YouTube .....	137
8.15	# Core .....	138
8.15.1	42 .....	138
<b>9</b>	<b>Formulare &amp; Inhaltstypen .....</b>	<b>141</b>
9.1	Konfiguration: Admin- und Seitenformular .....	141
9.1.1	Konfiguration .....	141
9.1.2	Datensicherheit .....	142
9.1.3	Allgemein .....	142
9.1.4	Kein Zugang .....	143
9.1.5	Variante .....	143
9.1.6	Validation .....	143
9.2	Konfiguration: Intro und Inhalt .....	143
9.3	Grundeinstellungen .....	144
9.4	Konstruktionsbereich .....	146
9.5	CSS-Formatierung .....	149
9.6	Menüverweis .....	150

<b>10 Listen &amp; Suchtypen</b>	<b>151</b>
10.1 Eine neue Suche	151
10.1.1 Template: Blog	152
10.1.2 Template: Table	153
10.1.3 Template: Kein	155
10.2 Konfiguration	155
10.2.1 Bereich: Konfiguration	155
10.2.2 Bereich: Allgemein	157
10.2.3 Bereich: Kein Ergebnis	159
10.3 Fields	160
10.4 Das Feld CCK	160
10.5 Nummer 3 im Suchformular	160
10.5.1 Basis	161
10.5.2 Datum und Uhrzeit	161
10.5.3 Ausschluss	161
10.5.4 Null	162
10.5.5 Numerisch	162
10.5.6 Wörter	162
10.6 Ordering	163
10.7 Liste	163
10.8 Element	163
10.9 Menüverweis	163
<b>11 Die Nummern in den Formularen &amp; Inhaltstypen</b>	<b>165</b>
11.1 Nummer 1	166
11.1.1 Einzelne Felder	166
11.1.2 Admin- und Seitenformular	166
11.2 Nummer 2	167
11.2.1 Admin- und Seitenformular - Live und Live-Wert	168
11.2.2 Intro und Inhalt	170
11.3 Nummer 3	170
11.3.1 Admin- und Seitenformular	170
11.3.2 Intro und Inhalt	174
11.4 Nummer 4	176
11.5 Nummer 5	176
11.5.1 Voneinander abhängige Zustände	177
11.5.2 Rechenregeln	180
11.6 Nummer 6	181
11.7 Der Punkt	182
<b>12 Das eigene Adminmenü</b>	<b>183</b>
12.1 Add-ons	184
12.2 Benutzerspezifisch	184
12.3 Konstruktion	185
12.4 E-Commerce	186
12.5 Formulare	186
12.6 Lists	187
12.7 Fazit zum Administrationsmenü	187

<b>Teil 3 – Der praktische Einsatz</b> .....	<b>189</b>
<b>13 Veranstaltungskalender</b> .....	<b>191</b>
13.1 Die Städte .....	192
13.2 Veranstalter .....	193
13.3 Das Event .....	194
<b>14 Portfolio</b> .....	<b>197</b>
14.1 Gestaltung .....	202
<b>15 Beiträge nach eigenem Gusto</b> .....	<b>205</b>
<b>16 Formulare für jeden Zweck</b> .....	<b>207</b>
16.1 Warum ein Kontaktformular? .....	208
16.1.1 Die Vorbereitung: Einrichtung .....	208
16.1.2 Erstellen der Mailfunktion .....	213
16.1.3 Feldtyp Email .....	213
16.1.4 Absender des Formulars festlegen .....	215
16.1.5 Das Feintuning .....	216
16.1.6 Das Erstellen der E-Mail-Nachricht .....	218
16.1.7 Der Inhalt im Detail .....	220
16.1.8 Darstellung des Formulars .....	222
16.1.9 Berechtigungen .....	222
16.2 Erweiterung zum Beschwerdeformular .....	223
16.2.1 Grundlagen des Beschwerdeformulars .....	224
16.2.2 Speichern oder nicht? .....	227
16.2.3 Der Grundaufbau des Formulars .....	227
16.2.4 Abhängigkeiten von Feldern .....	228
16.2.5 Abhängigkeiten im Detail .....	228
16.2.6 Das Beschwerdeformular fertigstellen .....	231
<b>17 Beziehungslisten</b> .....	<b>233</b>
17.1 Beziehungs-Weise .....	235
<b>18 Vorbelegung der Sprache bei Mehrsprachigkeit</b> .....	<b>237</b>
<b>19 Neue und andere Module</b> .....	<b>241</b>
19.1 Modul: From – SEBLOD .....	242
19.2 Modul: List – SEBLOD .....	242
19.3 Modul: Search – SEBLOD .....	243
19.4 Modul: Breadcrumbs – SEBLOD .....	244
19.4.1 Ein Easteregg? .....	244
19.5 Module ändern .....	245
<b>20 Multisites mit SEBLOD</b> .....	<b>247</b>
20.1 Einrichtung einer neuen Multisite .....	248
20.2 Die Multi-Berechtigungen .....	252
20.3 Post von der Mastersite .....	255
20.4 Eigene Felder .....	255
20.5 Mastersite? .....	257

<b>21</b>	<b>Templates</b>	<b>259</b>
21.1	Grundsätzlicher Aufbau	259
21.2	Ein neues Template zuweisen	261
21.3	Ein eigenes Template	262
21.4	Overrides im seb_one	263
21.4.1	Positionen	263
21.4.2	Variationen	267
21.4.3	Aufgeräumter Quellcode mit eigenen Markups	270
21.4.4	Verschiedene Markup-Overrides	272
21.4.5	Zusammenfassung und Fazit	273
21.5	Einstieg mit einem einfachen Template	273
21.5.1	Festlegen der Positionen	274
21.5.2	Damit überhaupt etwas ausgegeben wird: config.php	275
21.5.3	Die eigentliche Ausgabe: index.php	276
21.5.4	Rendering-Variationen	277
21.5.5	Der Ordnung halber: die eigene CSS-Datei	278
21.6	Globale Methoden und Eigenschaften	279
<b>22</b>	<b>Plug-ins</b>	<b>281</b>
22.1	Plug-in-Übersicht	281
22.2	Wo sind die Plug-ins geblieben?	284
22.3	Verbindung zwischen Benutzer und Beitrag	284
<b>A</b>	<b>CSS-Klassen</b>	<b>285</b>
A.1	Eingabe direkt im Feld	285
A.2	Eingabe im Markup	288
<b>B</b>	<b>Troubleshooting</b>	<b>291</b>
B.1	WYSIWYG-Editor im Frontend	291
B.2	Merkwürdige Zeichen im Frontend	292
B.3	SQL-Fehlerbeschreibungen im Frontend	292
B.4	Fehlermeldung: Übergeordnete Kategorie	292
B.5	Das Bild im Frontend ist nicht anklickbar	292
B.6	Template existiert nicht	293
B.7	Anzahl der sichtbaren Felder beschränkt?	293
B.8	Nach dem Joomla!- und SEBLOD-Update	293
B.9	Mehrere Startseitenmenüs bei Multisites	294
B.10	Inhaltstyp wird nicht angezeigt	294
<b>C</b>	<b>Weiterführendes</b>	<b>295</b>
C.1	Deutschsprachige SEBLOD-Community	295
C.2	Tutorials	296
C.3	Videos	296
C.4	SEBLOD-Erweiterungen	296
	<b>Stichwortverzeichnis</b>	<b>297</b>

# Teil I - Einstieg in SEBLOD

In diesem ersten Teil finden Sie mehr als nur eine bloße Übersicht über die SEBLOD-Struktur. Ein erstes praktisches Beispiel wird erstellt, in dem die Ein- und Ausgabe eines Joomla!-Beitrags mit SEBLOD gezeigt wird. Der erste Teil lässt bereits erkennen, wo die Stärken von SEBLOD liegen und wie einfach im Grunde die Arbeit damit ist.

Sie finden in diesem ersten Teil auch alles rund um die Speicherung. Die hat es durchaus in sich und kann zu ungeahnten Fehlern führen. Der Einstieg ist also mehr als ein bloßer Einstieg. Teilweise wird hier ordentlich hinter die Kulissen geschaut und so der Grundstock für die weitere Arbeit mit SEBLOD gelegt.





## Was ist SEBLOD?

SEBLOD ist ein Content Construction Kit, kurz CCK, mit dem Dinge erstellt werden, die bislang nur Entwicklern vorbehalten waren. So lassen sich komplexe Erweiterungen mit wenig Aufwand und damit auch deutlich weniger Zeit erstellen als mit der Programmierung.

### 1.1 Was ist ein CCK?

CCK steht für *Content Construction Kit* und bedeutet, dass die Ein- und Ausgabemaschinen für den Inhalt selbst gestaltet werden können, sowohl die Anordnung oder Anzeige von Feldern als auch das Erstellen neuer Felder, die nach Belieben eingefügt werden können. Manche CCKs fassen Inhalte sehr eng und beziehen sich meist auf die Gestaltung oder Erweiterung von Beiträgen.

Eine beliebige Erweiterung unter den Komponenten besteht stets aus Formularen, in denen etwas eingegeben werden kann oder in denen bereits etwas steht. Und im Frontend wird letztlich dieses Formular bzw. dessen Inhalt ausgegeben. Eigentlich würde es naheliegen, den Begriff »Inhalt« auch auf Erweiterungen auszudehnen. Das machen allerdings nur sehr wenige CCKs. Die wenigen, die das können, sind denn auch nicht so einfach zu durchschauen. Schon deswegen nicht, weil sie tatsächlich sehr viele unterschiedliche Bereiche und Einstellungen abdecken müssen. SEBLOD kann natürlich nicht auf bereits vorhandene externe Erweiterungen zugreifen, aber es

ist durchaus möglich, sich mit SEBLOD eigene Erweiterungen zu kreieren, ohne dass Programmierkenntnisse erforderlich sind.

Joomla! kann sehr viel. Dort, wo Joomla! von Haus aus nicht weiterkommt, helfen die über 10.000 Erweiterungen für Joomla!. Dennoch stößt man ab und an auf Grenzen oder Umständlichkeiten, zumal Erweiterungen oft genug den Nachteil haben, dass entweder ein entscheidendes kleines Detail fehlt oder aber viel zu viel dabei ist, was eigentlich gar nicht benötigt wird. Oft sind sie auch umständlich zu bedienen. Wer professionelle Webseiten erstellt, wird stets bemüht sein, seinen Kunden einen einfachen Zugang zu geben.

Selber Erweiterungen zu programmieren könnte eine Lösung sein. Aber nicht jeder kann programmieren, und eigene Erweiterungen in Joomla! zu programmieren ist nicht ganz so einfach, denn man braucht eine weitere Erweiterung – und Zeit. Ein CCK kann die Lösung für solche Probleme sein, aber auch ein CCK hat Grenzen. Es gibt Erweiterungen, die sich nicht mit einem CCK umsetzen lassen oder für die ein CCK einfach nicht geeignet ist.

SEBLOD ist eines dieser großen und mächtigen CCKs. Experten gehen davon aus, mit dem Einsatz von SEBLOD auf ca. 70 % aller Erweiterungen verzichten zu können. Jede Erweiterung, die Sie installieren, benötigt nicht nur Festplatten- und Datenbankspeicherplatz, sondern belastet auch die Performance Ihrer Website. Gerade bei größeren Websites könnte SEBLOD da eine interessante Alternative sein.

## 1.2 Was ist mit SEBLOD möglich?

SEBLOD wird gern als Eier legende Wollmilchsau bezeichnet, weil es eine Vielzahl von Möglichkeiten bietet und in seiner Funktionalität kaum eingeschränkt ist. Im Gegensatz zu den meisten anderen CCKs stülpt SEBLOD Joomla! nicht etwas über, sondern integriert sich nahezu komplett in Joomla!. Der Vorteil ist, dass alles, was neu in Joomla! angelegt wird, ein Beitrag, eine Kategorie, eine Benutzergruppe oder ein neuer Benutzer ist. Das sind zumindest die vorgegebenen SEBLOD-Möglichkeiten.

Durch diese Integration ist auch weiterhin alles möglich, was Joomla! von Haus aus bietet. Beispielsweise kann die Rechteverwaltung oder die Navigation komplett auf das CCK wirken, ohne dass das CCK hier selber etwas mitbringen müsste. Da also alles sehr gut weiterverwendet werden kann und lediglich etwas Neues dazukommt, muss sich aber auch niemand Gedanken darüber machen, wie etwas zur Anzeige gebracht werden kann. Ein einfacher Menüverweis reicht in aller Regel.

Ich selber habe mir ein Portfolio mit SEBLOD erstellt. Ich war mit den vorhandenen Portfoliokomponenten nicht zufrieden und hatte meine eigenen Vorstellungen, wie meine Referenzliste aussehen sollte. Das, was ich schließlich mit SEBLOD entwickelt habe, hat mich ungefähr einen halben Tag Arbeit gekostet – einschließlich umfangreicher CSS-Formatierungen.

Hätte ich das Gleiche als Komponente programmiert, hätte ich vermutlich ein paar Tage länger benötigt. Und ich habe jetzt den zusätzlichen Vorteil, dass ich jederzeit mein Portfolio umbauen oder erweitern kann, mal eben schnell ein neues Feld hinzufügen kann, ohne dass ich danach erst umständlich ein Update meiner Komponente einspielen müsste. Auch brauche ich mir keine Gedanken über neue Joomla!-Versionen und die Anpassung meiner mit SEBLOD erstellten Erweiterungen zu machen. Es reicht, neben Joomla! auch SEBLOD upzudaten, und alles ist sauber und sicher.

Da SEBLOD es ebenfalls gestattet, eigene Datenbanktabellen zu benutzen, sind in SEBLOD nur wenige Grenzen gesetzt. Ein Großteil der gängigen Joomla!-Erweiterungen ließe sich relativ problemlos in SEBLOD abbilden. Kompliziert wird es erst, wenn umfangreiche Verwaltungsaufgaben entstehen. Beispielsweise ein komplexeres Mailing- oder PM-System ist meines Erachtens in einer eigenständigen Komponente nach wie vor besser aufgehoben.

#### SEBLOD-Forum

Auf den Seiten von SEBLOD selber ist beispielsweise auch das Forum mit SEBLOD gemacht. Generell funktioniert das zwar, jedoch ist das Forum meines Erachtens sehr unübersichtlich, und es ist auch nicht so einfach, dort zu finden, was zusammengehört (siehe URL [www.seblod.com/community/forums](http://www.seblod.com/community/forums)).

Übrigens - das deutschsprachige SEBLOD-Forum finden Sie unter <http://seblod.time4mambo.de>.

Es gibt in Joomla! einige mächtige Erweiterungen, die über die Jahre entstanden sind, stetig weiterentwickelt werden und längst von diversen Kinderkrankheiten befreit sind. Ich würde stets auf diese Erweiterungen zurückgreifen und nicht mit SEBLOD versuchen, das Rad neu zu erfinden. Aber dort, wo diese Erweiterungen nicht reichen, ist SEBLOD stets eine Überlegung wert.

Dennoch lassen sich in SEBLOD nicht nur kleine Lösungen bequem umsetzen. Es gibt beispielsweise eine SEBLOD-Erweiterung für eine Immobilienseite, die sehr umfangreich ist, sowie Dokumentationen und Download-Seiten, die mit SEBLOD erstellt sind. Ausgeklügelte Redaktionssysteme, Dokumentationen, Kundenerfahrungsberichte, Affiliateseiten in Selbstverwaltung, mehrsprachige Kataloge und Eventseiten sind nur ein paar Beispiele dafür, was mit SEBLOD möglich ist.

Dazu kommt, dass es mit SEBLOD möglich ist, Joomla! multidomainfähig zu machen. Ein Joomla! und viele Domains und Subdomains greifen auf eine »Mastersite« zu. Bislang war das nur mit (teils kostenpflichtigen) umfangreichen Erweiterungen möglich - nun ist das auch recht einfach mit SEBLOD möglich.

## 1.3 Versionen

SEBLOD selber gibt es bereits seit Joomla! 1.5. Die erste SEBLOD-Version hieß jSEBLOD und änderte mit der SEBLOD-Version 2 ihren Namen zu SEBLOD, die auf

Joomla! 2.5x lief. Mit Joomla! 3x kam SEBLOD 3 auf den Markt, die teilweise auch in Joomla! 2.5x noch lief. Meist wird SEBLOD zeitnah zu neuen Joomla!-Versionen ebenfalls upgedatet. Ab Joomla! 3.2 gilt der gleiche Versionsstrang von SEBLOD. Die jeweils aktuelle SEBLOD-Version finden Sie auf der SEBLOD-Site<sup>1</sup>.

**Joomla! 3.x und SEBLOD 3.x**

Während dieses Buch entstand, wechselte Joomla! seine Versionspolitik. Das hatte zur Folge, dass während der Erstellung dieses Buches verschiedene Versionen zum Einsatz kamen. Entscheidend für dieses Buch ist die Joomla!-Version 3.3.6. Alles was Sie hier lesen, läuft auf dieser Joomla!-Version. Lassen Sie sich also bitte nicht von einem Screenshot erschrecken, der eine für Sie alte Versionsnummer trägt – er hat dennoch weiterhin seine Gültigkeit.

Da SEBLOD in aller Regel erst dann eingesetzt wird, wenn in Joomla! an Grenzen gerüttelt wird, setze ich voraus, dass Sie zumindest rudimentäre Kenntnisse in Joomla! haben. Auf die Handhabung und die Funktionalitäten in Joomla! gehe ich in diesem Buch nicht ein, sofern es der Umgang mit SEBLOD nicht erforderlich macht.

---

<sup>1</sup> <http://www.seblod.com>

# 2

## Der schnelle Überblick

### 2.1 Installation

Die Installation von SEBLOD ist sehr einfach. Wie in Joomla! gewohnt, wird das gezippte SEBLOD-Package direkt über *Erweiterungen: Installieren* in Joomla! integriert.

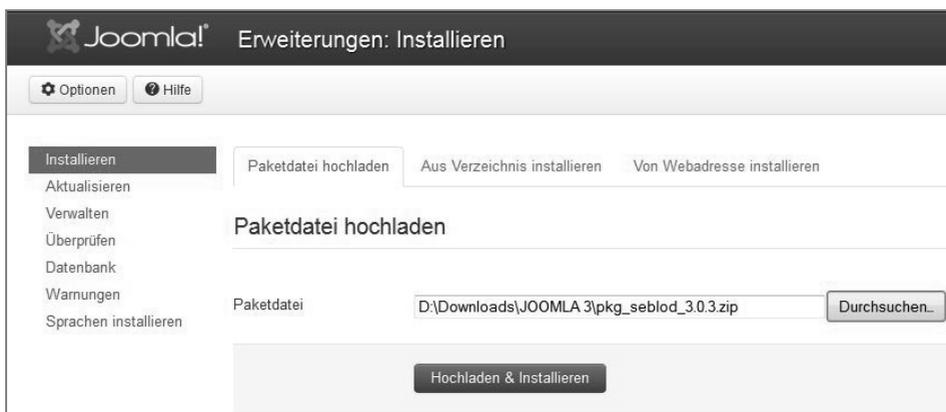


Bild 2.1: Die Installation läuft nach dem gewohnten Schema ab.

Der Installationsvorgang dauert ein wenig. Je nach Internetverbindung kann das schon einmal zwei, drei Minuten Zeit in Anspruch nehmen. Dafür legt SEBLOD 19 neue Datenbanktabellen an und installiert einige neue Dateien. In seltenen Fällen kann es dabei auch zu einem Time-out kommen.

- Eine mögliche Abhilfe besteht darin, dass Sie den Link des Downloads auf SEBLOD.com kopieren und ihn direkt im Register *Von Webadresse installieren* unter *Erweiterungen: Installieren* eingeben.
- Oder Sie laden das Paket per FTP in das *tmp*-Verzeichnis und installieren es vom Register *Aus Verzeichnis installieren* aus.
- Sollte das auch keine Abhilfe bringen, gibt es eventuell noch die Möglichkeit, unter *Erweiterungen: Installieren* auf der linken Seite die *Warnungen* auszuwählen und über die angebotenen *Optionen* (Schaltfläche oben links) den *Aktualisierungszwischenspeicher* nach oben zu setzen.

## 2.2 Apps downloaden und installieren

Wenn Sie fertige Apps installieren möchten (vom SEBLOD-Team werden beispielsweise einige fertige Apps angeboten), installieren Sie sie ebenfalls über *Erweiterungen: Installieren*, also so, wie Sie es kennen.

Ihre eigenen Apps mit allen Feldern und Formularen können Sie sehr einfach downloaden, indem Sie im Apps-Verzeichnis den blauen, nach unten weisenden Pfeil anklicken.



**Bild 2.2:** Die eigene App downloaden.

Zum einen können Sie so sehr einfach eine Sicherheitskopie Ihrer selbst erstellten App anlegen, zum anderen können Sie so natürlich eine App auch weiterreichen. Sie haben auch die Möglichkeit, lokal in aller Ruhe eine App zu entwickeln und sie erst, wenn alles fertig und getestet ist, in Ihr Live-System zu laden.

Sie speichern mit dem Download-Button Ihre komplette App, also auch alle Einstellungen, die Sie bei den Feldern und Formularen gemacht haben, und natürlich auch alle zugehörigen Formulare und Suchmasken. Wenn Ihre App weitere Untergruppierungen hat, werden diese ebenfalls gespeichert. Wenn Sie nicht die ganze App, sondern nur Teilgruppen der App sichern möchten, klicken Sie nicht den größeren blauen Pfeil, sondern den in der Zeile stehenden kleineren Pfeil an. Damit können Sie Ihre

App beispielsweise lokal weiterentwickeln und anschließend Ihre bestehende App einfach erweitern.

Was es mit den einzelnen Gruppierungen auf sich hat, werden Sie im weiteren Verlauf des Buches erfahren.

## 2.3 SEBLOD spricht deutsch

SEBLOD spricht auch deutsch. Es gibt zwei voneinander unabhängige Sprachversionen. Die eine kann nach einigem Suchen im SEBLOD-Forum gefunden werden und wird von Pierre Corell<sup>2</sup> entwickelt. Leider stagniert die Sprachversion seit Monaten (Stand: Oktober 2014).

Des Weiteren gibt es eine wirklich sehr gute und sehr umfangreiche Übersetzung von Paul Brand<sup>3</sup>. Paul hat nicht nur die Sprachdateien von SEBLOD, sondern auch die allermeisten Plug-ins in die deutsche Sprache übersetzt. Zudem lässt sich die jeweils aktuelle deutsche Sprachdatei einfach über das Kontrollzentrum in Joomla! aktualisieren.

In diesem Buch finden sich Screenshots von beiden Übersetzungen. Das liegt daran, dass Paul erst etwas später mit der Übersetzung begonnen hat.

## 2.4 Grundsätzliche Funktionsweise

Als ich das erste Mal SEBLOD installiert hatte, habe ich einige Male mehr oder weniger unkontrolliert hin- und hergeklickt und dann das Ganze wieder deinstalliert, weil ich nicht wirklich durchgestiegen bin. Tatsächlich präsentiert sich SEBLOD zunächst nicht unbedingt innovativ und auf den ersten Blick etwas unübersichtlich. Beginnt man aber, etwas Ein- und Überblick zu bekommen, wird die Struktur stetig nicht nur übersichtlicher, sondern auch logisch.

Das Faszinierendste ist für mich dabei die Erkenntnis gewesen, dass SEBLOD im Grunde genommen stets der Joomla!-Mimik folgt, also nicht etwas gänzlich Eigenes macht und Joomla! überstülpt, sondern mit dem Joomla!-System selber arbeitet und somit eine problemlose Zusammenarbeit der hauseigenen Funktionen von Joomla! mit und innerhalb von SEBLOD ermöglicht.

---

<sup>2</sup> <http://joomla-praxis.de> oder <http://pierre-corell.net/>

<sup>3</sup> <http://seblod.inetserv.net>

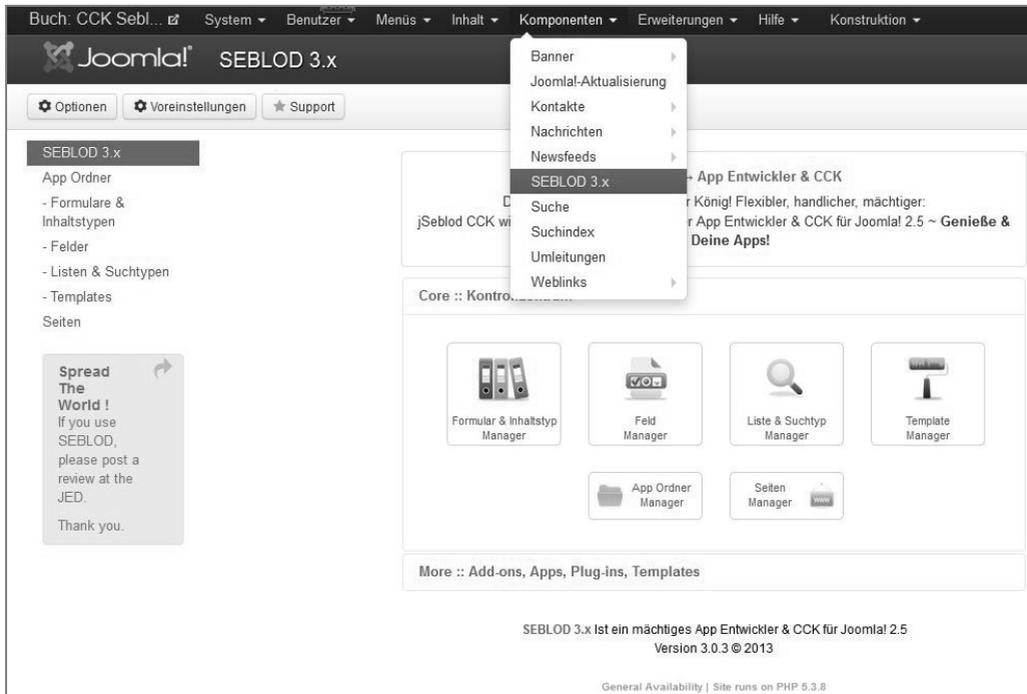


Bild 2.3: SEBLOD 3.x im Überblick.

### SEBLOD legt neue Beiträge an.

Auf diesen lapidaren Satz könnte ich SEBLOD beinahe reduzieren. Bleiben wir zunächst bei dieser vereinfachten Feststellung, dann bemerken wir, dass wir zwar jede Menge eigener Erweiterungen mit SEBLOD machen können, die Verwaltung dieser eigenen Inhalte aber am Ende nichts weiter als Beiträge sind. Beiträge wiederum können auf vielfältige Weise per Menü angesprochen und gestaltet werden. Diverse Module greifen direkt auf Beiträge zu, die Suche sowieso, und die Rechteverwaltung von Joomla! weiß einiges mit Beiträgen und davon ableitend der Menüführung anzufangen.

Dadurch benötigt SEBLOD keine eigene Rechteverwaltung, weil SEBLOD einfach die vorhandene benutzt – ebenso wie die Navigation und vieles andere mehr.

SEBLOD ist komplett in Joomla! integriert. Um es durchschaubarer zu machen, werfe ich einen schnellen Blick in die einzelnen Bereiche.

#### 2.4.1 App-Folder

»App« ist eigentlich etwas irreführend oder zumindest recht verwirrend, weil jeder Apps im Zusammenhang mit mobilen Endgeräten kennt und der Programmierer aus

der »App« noch die »Applikation« zaubert. Ich persönlich neige dazu, den »App-Folder« eher mit »Struktur« gleichzusetzen, der Struktur unserer eigenen Erweiterung in SEBLOD, zu der dann Felder und Formulare gehören. Andererseits lässt aber das Wortkürzel »App« auch erkennen, was SEBLOD genau macht: eine eigene Applikation, eine eigene Erweiterung.

Im einfachsten Fall besteht unsere SEBLOD-Erweiterung aus einem einzigen Bereich, also einer sehr klaren und ausgesprochen überschaubaren Struktur.

Weiter unten in diesem Buch bauen wir uns eine einfache News-App zusammen. Diese Newserweiterung wird aus einem Titel, etwas Text, einem Bild und einem Link bestehen. Im App-Folder bietet sich als Bezeichnung *News* an. Die vier Felder werden wir dieser Struktur zuweisen und am Ende dafür sorgen, dass das auch vernünftig ausgegeben wird.

Zusammenfassend: Der App-Folder gibt die grobe Struktur unserer SEBLOD-Erweiterung vor. Fügen wir Felder und Formulare hinzu, entsteht am Ende eine App.

## 2.4.2 Formulare und Inhaltstypen

An dieser Stelle sorgen Sie endlich dafür, dass es auch Formulare zum Ein- und Ausgeben von Daten gibt. Sie legen selbst fest, ob Inhalte nur im Backend vom Admin oder auch von berechtigten Benutzern im Frontend eingegeben werden können. Sie entscheiden, wie Ihre Inhalte ausgegeben werden – sowohl was das Layout als auch den Inhalt und Form betrifft. So steht ein Intro zur Verfügung für den schnellen inhaltlichen Überblick und die detaillierte Sicht. Wie genau Sie was definieren, bleibt selbstverständlich vollkommen Ihnen überlassen.

## 2.4.3 Felder

Wie bereits erwähnt, werden in diesem Register die Felder unserer App erstellt. In meinem Beispiel mit den News erstelle ich die vier Felder *Titel*, *Text*, *Bild* und *Verweis* und weise sie meinem App-Folder zu.

Möglich ist an dieser Stelle natürlich auch, statt einer eigenen App eine bereits vorhandene zu erweitern, beispielsweise die Joomla!-Beiträge, wie wir sie bislang kennen. Sie können zwar auch Felder direkt bei den Formularen und Inhaltstypen zuweisen, aber hier sind die Übersicht und die Handhabung etwas besser.

## 2.4.4 Listen & Suchtypen

Ich kann die News in meinem Beispiel auflisten, indem ich nur den Titel anzeige, oder ich kann die Auflistung als Blog mit dem Text und dem Verweis darstellen oder – wie Sie es möchten. Hier bestimmen Sie, wie solche Listen aussehen sollen.

Und auch die Suche können Sie selber bestimmen. Da Sie auch Abhängigkeiten definieren können – beispielsweise *Dieses Feld wird erst angezeigt, wenn in dem Feld dieses und jenes steht* –, können Sie bereits erahnen, dass Sie vielfältige Suchmuster erschaffen können. Sie entscheiden, was gesucht und wie es ausgegeben wird.

## 2.4.5 Templates

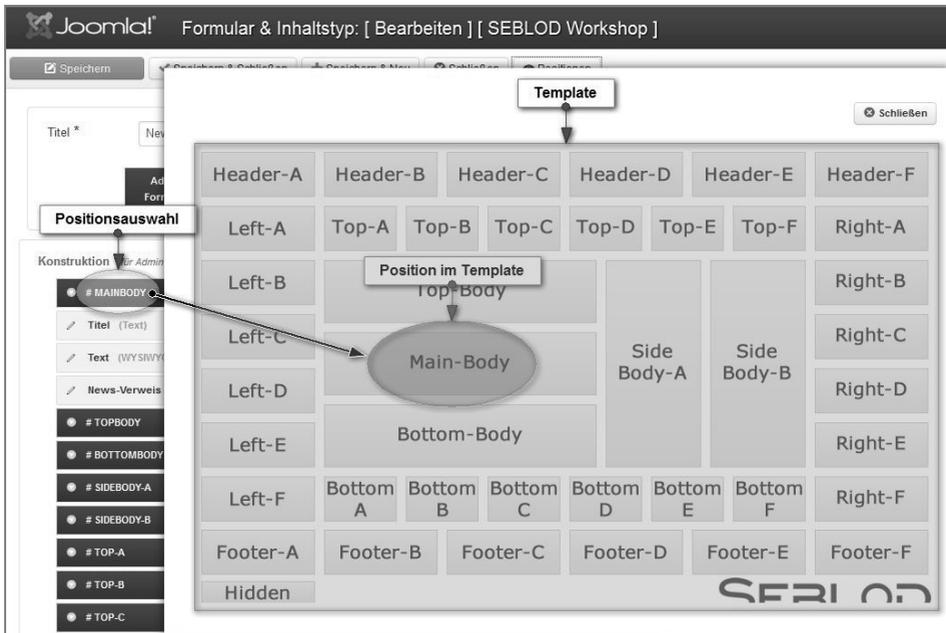
Das hat mich anfangs sehr verwirrt: Unter Templates versteht man in Joomla! stets Design und Layout einer Website. Tatsächlich finden sich auch die mitgelieferten SEBLOD-Templates in der Template-Übersicht wieder – allerdings mit dem strikten Hinweis, dass sie sich nicht zur Seitenanzeige eignen. Probieren Sie es dennoch aus, passiert nichts Sinnvolles auf Ihrer Webseite.

### Template und Template

SEBLOD unterscheidet zwischen – Template und Template. Falls Sie gerade die Augen verdrehen sollten – ich bringe gleich Licht in diese Merkwürdigkeit.

Wenn Sie einen Beitrag in Joomla! eingeben, dann sehen Sie oben zunächst den Titel, daneben die Kategorieauswahl, darunter den Editor, darunter unter anderem die Bilder und Links, rechts neben dem Editor ein paar Details zum Auswählen und weitere Tabs mit verschiedenen Einstellungen. Dieser ganze Bereich wird in SEBLOD Template genannt.

Zudem gibt es das Template nur und ausschließlich für die Eingabe des Beitrags. Dessen Aussehen können Sie umgestalten, indem Sie bestimmen, an welcher Stelle was abgebildet wird, indem Sie die einzelnen Felder einfach anderen Positionen zuordnen oder indem Sie ein ganz neues eigenes Template entwickeln.



**Bild 2.4:** Template-Positionen in SEBLOD.

Zusammengefasst bedeutet das: Das SEBLOD-Template umfasst nur den Inhaltsbereich, nicht das Gesamtaussehen der Website. Dieser ganze Ablauf ist am Ende ein einzelner Beitrag, bestehend aus Struktur - Feldern - Form und gegebenenfalls eigenen Auflistungen und Suchen.

Was ich eben erklärt habe, gilt ebenso für Kategorien, die Sie umgestalten können, wie auch für Benutzer und Benutzergruppen, die Sie so nach Ihrem eigenen Gusto anlegen können.



Und alles läuft stets wie gehabt innerhalb von Joomla!. Mit SEBLOD erstellen Sie sich zwar Ihre eigenen Ein- und Ausgabemasken, handhaben anschließend aber alles im normalen Joomla!-Kontext.

Meine News-App (die im nächsten Kapitel angelegt wird) sieht zwar komplett anders aus als ein Joomla!-Beitrag, wie wir ihn bisher kennen, aber ich kann diese selbst erstellten News ganz normal, und wie wir es bisher kennen, im Menü ansprechen – als Kategorieblog und Liste, als Einzelverlinkung oder als Archiv. Und ebenso kann ich mit der Rechteverwaltung auf diese Beiträge zugreifen, und mir stehen somit alle Joomla!-Möglichkeiten offen.

Bei anderen CCKs haben Sie hingegen meist das Problem, dass Sie das dort Erstellte nur so verarbeiten können, wie es das jeweilige CCK vorgibt. Die gewonnene Freiheit des CCK wird oft mit Einschränkungen der Weiterverarbeitung wieder zunichtegemacht. Nicht so in SEBLOD.

# 3

## Der detaillierte Blick

Ich will ganz ehrlich zu Ihnen sein: Als ich anfang, mich mit SEBLOD zu beschäftigen, ein paar Einführungsvideos gesehen und das eine oder andere spärliche Tutorial gelesen hatte, wusste ich nicht wirklich, was ich nun tun sollte. Zwar hatte ich den generellen Zusammenhang schnell erfasst, aber wie man damit arbeitet und was am Ende dabei herauskommt, habe ich erst begriffen, als ich meinen ersten Versuch einer konkreten App gestartet habe. Darum möchte ich auch in diesem Buch erst einmal ganz vorsichtig anfangen. Ich denke, spätestens nach diesem Kapitel werden Sie die Möglichkeiten, die SEBLOD bietet, besser einschätzen können.

Für diesen detaillierten Blick habe ich eine einfache Newsausgabe gewählt, bestehend aus einem Titel, einem Bild, etwas Text und einem Link.

Lernziele in diesem Kapitel:

- Grundsätzlicher Aufbau einer App.
- Erste Beispiele mit den Details in den Formularen und Inhaltstypen.
- Abhängigkeiten zwischen zwei Feldern.
- Nutzung bereits vorhandener Felder.

Wir steigen mit dieser ersten App ein, bereits tiefer in die Materie ein. Dennoch »übersehe« ich großzügig einige Bereiche in dieser ersten App. SEBLOD ist sehr vielschichtig und bietet Ihnen einiges an Einstellungen und Möglichkeiten. Nach diesem

Kapitel werden Sie wesentlich sicherer mit SEBLOD umgehen können und werden wissen, wo Sie was finden.

#### Download der App

Sie können diese App mit dem Namen *app\_cck\_news.zip* von meiner Website downloaden.<sup>4</sup>

Die verwendeten Icons sind aus verschiedenen Iconsammlungen und dürfen frei benutzt werden. Ich habe die wichtigsten zusammengepackt, und sie können an der gleichen Adresse als *images.zip* heruntergeladen werden. Diese Icons laden Sie bitte in das *images*-Verzeichnis Ihrer Joomla!-Installation.

Sie können SEBLOD auf unterschiedliche Weise aufrufen, zum einen über das Komponentenmenü und *SEBLOD 3.x*, womit Sie stets in die Übersicht von SEBLOD gelangen. Dort können Sie sowohl das Menü links als auch im Kontrollzentrum die einzelnen Bereiche von SEBLOD ansteuern (siehe Bild 2.3 auf Seite 20).

Oder Sie rufen SEBLOD gezielt direkt über das neu angelegte Menü *Konstruktion* auf, das Sie in der Menüleiste ganz rechts nach der Installation finden. Dort finden Sie auch diverse Verlinkungen zu Dokumentationen seitens des SEBLOD-Teams. Die Links führen allerdings meist zu englisch- und französischsprachigen Seiten. Über dieses Menü kommen Sie auch nicht direkt zu den Optionen und Voreinstellungen. Sie müssen erst auf die »Übersicht« klicken: *SEBLOD 3.x*.

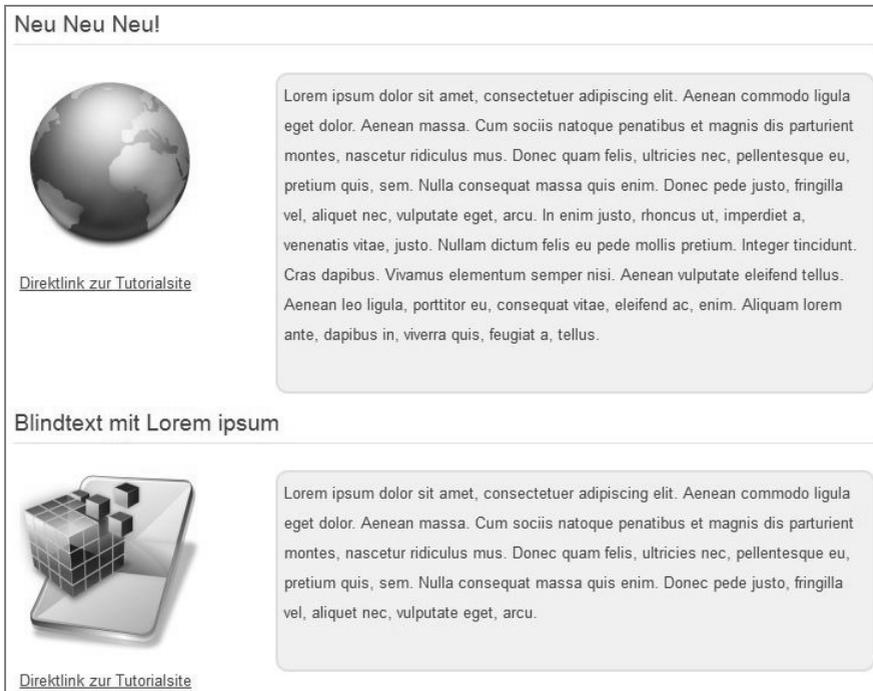
## 3.1 Die erste App: Newsanzeige

Der erste Schritt unterscheidet sich in der Ausgabe letztlich kaum von einem Joomla!-Beitrag. Dennoch ist ein eigener Beitrag mit eigenen Eingabemasken erstellt worden. Sie können dieses Beispiel auch sehr gut als Grundlage für ein eigenes Redaktionssystem nehmen, in dem Sie die Eingabe von Beiträgen für Redakteure vereinfachen. Um einen Eindruck zu bekommen, wie es aussehen soll – was also in den nachfolgenden Seiten gemacht wird –, zeigt der folgende Screenshot die fertigen News.

Mit nur sehr einfachen Formatierungsangaben in SEBLOD lassen sich die News ein wenig aufpeppen und unterscheiden sich dann auch deutlich vom herkömmlichen Joomla!-Beitrag. Mit Joomla! können Sie das gleiche Ergebnis erreichen, müssen jedoch bei der Eingabe sehr auf die richtige Form achten, müssen vielleicht sogar Tabellen benutzen und sollten sich auch detaillierter in CSS auskennen. Und Sie müssen es bei jedem Newsbeitrag erneut in dieser Form eingeben. SEBLOD hingegen stellt sehr einfache Mechanismen zur Verfügung, um es so aussehen zu lassen und ohne dass Sie sich bei der Eingabe Gedanken über das spätere Aussehen machen müssen.

---

<sup>4</sup> <http://www.time4mambo.de>



**Bild 3.1:** Unsere News im Frontend.

Je nach verwendetem Template kann das bei Ihnen etwas anders aussehen. Sie sehen hier das Standard-Template Beez 3 in Joomla! 3.3.6. Zur Erinnerung: SEBLOD integriert sich komplett in Joomla!.

### 3.1.1 Einfache Anlage im App-Verzeichnis

Im App-Folder liegen zunächst die vorhandenen Apps. Das sind die Beiträge, Kategorien, Benutzer und Benutzergruppen, also die Joomla!-Standards. Diese Apps können Sie nach Belieben ändern und Ihrem Geschmack anpassen. Sie brauchen auch keine Angst zu haben, dass Sie damit die grundsätzlichen Joomla!-Bereiche zerstören. SEBLOD hat Ihnen, nach Standardvorlage, einen eigenen Satz Formulare zur Verfügung gestellt. Wichtig für uns an dieser Stelle ist, dass wir natürlich mit allem, was schon da ist, auch weiterarbeiten können bzw. all dieses in unseren eigenen Apps nutzen können.

Erstellen wir nun unsere News-App. Dazu einfach beherzt oben links auf *Neu* klicken. In das Feld *Titel* schreiben Sie den Namen Ihrer App, hier *News*.

Im Listenfeld *Übergeordneter* wählen Sie den Ordner *Top* aus, weil wir eine eigene App erstellen wollen und keine neue Gruppe, die in eine bestehende App integriert ist.

Sie können hier sehr schön sehen, dass ein Joomla!-Beitrag (als Beispiel genannt) aus vielen Untergruppen besteht. Und wenn Sie sich die Eingabe eines Joomla!-Beitrags in Erinnerung rufen oder nachschauen, werden Sie die einzelnen Strukturen dort auch wiederfinden.

The screenshot shows the Joomla! News App configuration interface. At the top, there are buttons for 'Speichern', 'Speichern & Schließen', '+ Speichern & Neu', 'Als Kopie speichern', and 'Schließen'. Below these, the 'Titel \*' field contains 'News', and the 'Übergeordneter' dropdown is set to 'Top'. The 'Beschreibung' field has an 'Editor' button, and the 'Status' is checked. The 'Optionen' section includes:

- App Root (1. Ebene):** Radio buttons for 'Nein' and 'Ja (1.Ebene der Applikation (App))', with 'Ja' selected.
- Haupteintrag:** Radio buttons for 'Nein' and 'Ja (Gekennzeichneter Ordner kann als Gerüst für neuen Inhaltstyp ausgewählt werden)', with 'Ja' selected.
- Symbol:** A text field containing 'images/zeitung.png'.
- Elemente:** A table with checkboxes and input fields:
 

<input checked="" type="checkbox"/> Formular & Inhaltstyp	Farbe	<input type="text" value="#878787"/>
<input checked="" type="checkbox"/> Feld	Zeichen	<input type="text" value="N"/>
<input type="checkbox"/> Liste & Suchtyp		
<input type="checkbox"/> Template	Zeichenfarbe	<input type="text" value="#ffffff"/>

**Bild 3.2:** Einstellungen der *News*-App.

Klicken Sie auf den Button *Editor*, öffnet sich ein WYSIWYG-Editor, in dem Sie eine Anmerkung/Beschreibung eingeben können. Das ist besonders dann sinnvoll, wenn Sie eine umfangreichere App erstellen und auch noch Monate später nachlesen möchten, was Sie sich heute dabei alles so gedacht haben.

Der *Status* zeigt Ihnen an, ob die App veröffentlicht oder gesperrt ist, was Sie aber erst nach dem ersten Speichern sehen.

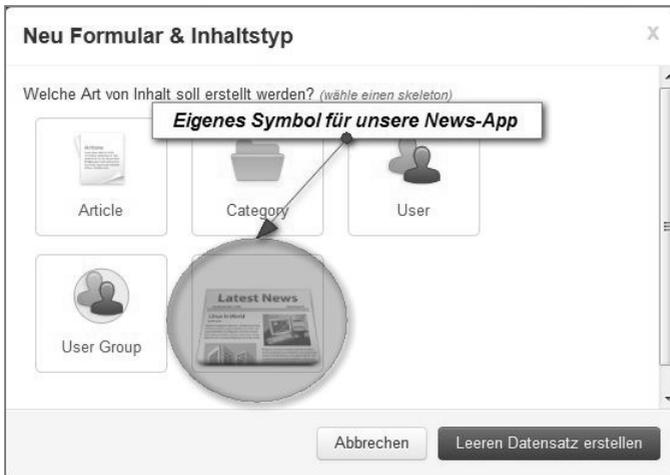
*App Root (1. Ebene)* zeigt an, ob es sich um die Hauptebene Ihrer App handelt oder um eine Unterstruktur. In unserem Fall haben wir nur eine Ebene, und das ist natürlich auch die erste Ebene. Hier wählen Sie *Ja*.

#### Hinweis

Es ist denkbar, dass Sie innerhalb einer bestehenden App eine neue App erstellen, weshalb Abweichungen vom übergeordneten Verzeichnis und der ersten Ebene auftreten können. In unserem Beispiel liegt beides auf der ersten Ebene.

Der *Haupteintrag* ist sehr wichtig. An dieser Stelle legen Sie fest, dass diese App bzw. diese Struktur später ausgewählt werden kann.

Sie können ein eigenes *Symbol* angeben. Dieses Symbol wird überall dort angezeigt, wo Sie die App auswählen können. Also beispielsweise wenn Sie einen neuen Inhaltstyp anlegen möchten. Lassen Sie dieses Feld leer, wird das Standardsymbol genommen.



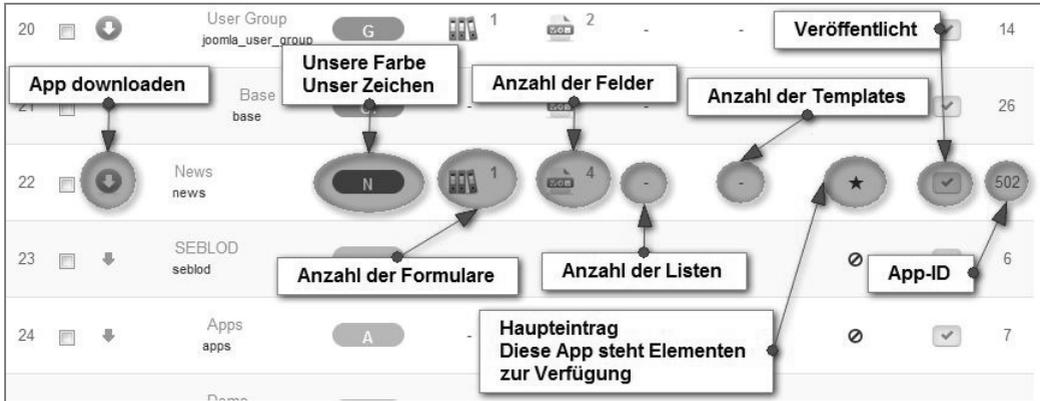
**Bild 3.3:** Ein eigenes Symbol für unsere App.

Die *Elemente* sind sehr interessant. Sie legen an dieser Stelle fest, für welche Elemente Sie Ihre App verwenden möchten. Es ist gut denkbar, dass Sie eine App nur für Suchtypen oder nur für die Felder als Untergruppe anlegen möchten. Vielleicht brauchen Sie auch eine App, weil Sie ein Modul erstellen möchten, aber mit der App selber nicht weiterarbeiten wollen. Wie Sie weiter unten im Buch sehen werden, lassen sich sehr unterschiedliche Strukturen erstellen.

In unserem Beispiel muss die App für Formulare, Inhaltstypen und für die Felder zur Verfügung stehen. Die *Elemente* dienen der weiteren Bearbeitung der App und kennzeichnen auch, was mit der App möglich ist. Weiter unten im Buch werden wir Apps in der App erstellen. Da das verwirrend klingt, bevorzuge ich den Begriff »Struktur« statt »App«. Eine Unterstruktur in einer Hauptstruktur klingt angenehmer.

Die *Farbe* dient nur der besseren Übersicht. Überall, wo die App angezeigt wird, erscheint ein kleines Symbol (Kreis oder Oval) mit Ihrer Farbe in der Zeile Ihrer App. Darin stehen der oder die Buchstaben, die Sie in das Feld *Zeichen* eingeben. Sie können maximal zwei Buchstaben eingeben. In meinem Beispiel kennzeichne ich die News passenderweise mit einem »N« und einer grauen Farbe (was diesem Buch geschuldet ist). Die *Zeichenfarbe* gibt die Schriftfarbe in Ihrem Farbkreis an.

Nach dem Speichern taucht die *News*-App in der Liste auf. Sie haben hier stets den Überblick über die Inhalte Ihrer App. Momentan wird bei Ihnen noch nichts stehen, aber zum Ende dieses ersten Beispiels werden Sie in der Zeile einige Informationen erhalten.



**Bild 3.4:** Die App im Überblick.

### 3.1.2 Direkt zu den Feldern

Vier Felder benötigen wir für unsere *News*-App. Wählen Sie im SEBLOD-Menü die *Felder* aus und erstellen Sie hier die nachfolgenden Felder:

- Titel
- Text
- Bild
- Verweis

#### Feldtyp: Text

Beginnen wir zunächst mit dem Feld für den Titel. Nachdem Sie in der Übersicht auf *Neu* geklickt haben, ist als einziges Pflichtfeld das Titelfeld gekennzeichnet, und wenn Sie dort etwas eingetragen haben, auch das Namensfeld. Ich nenne mein Newstitelfeld auch genau so: *Titel*.

Hinter *Name* verbirgt sich der Alias. Den Alias (Namen) können Sie frei lassen. SEBLOD beschreibt, genauso wie Joomla!, dieses Feld aus dem Titel selbstständig. Leerzeichen werden mit Unterstrichen ersetzt und Umlaute sowie Sonderzeichen gelöscht. Normalerweise reicht es, SEBLOD den Alias zu überlassen. Erst wenn Sie Felder mit gleichen Namen haben, ist es erforderlich, den Alias anzupassen. Der Alias wird sowohl intern gebraucht als auch bei Abhängigkeiten, wie sie weiter unten beispielsweise beim *Feld X* beschrieben sind. Eventuell greifen Sie auch in Programmie-

rungen auf den Alias zu. Sie können den Alias vor dem ersten Speichern anpassen, danach ist er festgeschrieben und kann nicht mehr verändert werden.

Als *App Ordner* müssen Sie nun *News* auswählen. Dieser stehen etwas weiter unten in der langen Liste. Auch hier können Sie wieder einen Beschreibungstext mit dem Editor eingeben und sehen nach dem Speichern, ob das Feld veröffentlicht ist.

Alles, was in dem Bereich *Konstruktion* steht, ergibt sich aus der Auswahl hinter *Typ*. Der *Typ* ist quasi eines der großen Geheimnisse von SEBLOD und lässt bereits in der reinen Auswahlliste die vielfältigen Möglichkeiten von SEBLOD erahnen. Für unser Titelfeld ist *Text* die richtige Auswahl.

Im Bereich *Konstruktion* sehen wir das *Bezeichnungs-Feld*. Die Eingabe hier wird vor dem Eingabefeld später in der Eingabemaske und, wenn Sie möchten, auch bei der Ausgabe angezeigt. Im *Standard-Wert* können Sie einen Titel vorgeben, müssen dann aber bedenken, dass der Alias jeweils angepasst werden muss, da es in Joomla! keine zwei Beiträge mit gleichen Alias geben kann.

*Mindest-* und *Maximallänge* beziehen sich auf den einzugebenden Text.

#### Minimallänge statt 0

Sie könnten statt der 0 auch eine Minimallänge angeben, um eine Eingabe zu erzwingen. SEBLOD bietet aber an anderer Stelle eine einfachere Möglichkeit, ein Feld als Pflichtfeld zu kennzeichnen.

Die *Größe* bezieht sich auf den Eingabebereich des Feldes später in den Formularen. Wird der Wert zu klein gewählt, kann unter Umständen nicht das ganze Textfeld beschrieben werden. Deswegen scheint es sinnvoll zu sein, die vorgegebenen Werte einfach zu übernehmen.

Kommen wir zum Bereich *Speicher*. Viele Fehler, Ungereimtheiten und Verzweiflungsfläche haben ihren Ursprung an dieser Stelle. Weiter unten im Buch rücke ich dem Speicherbereich näher auf die Pelle, bis dahin folgen Sie einfach meinen Vorschlägen.

Im *Speicher*-Bereich wird festgelegt, wie und wo SEBLOD den Inhalt unseres Feldes in der Datenbank speichern soll. Wie ich ja schon schrieb, erzeugt SEBLOD am Ende einen neuen Beitrag, der das Aussehen unserer App hat. Ein Beitrag wird in Joomla! mit der Artikel-ID einwandfrei gekennzeichnet. SEBLOD vereinfacht uns diesen Vorgang etwas, indem wir nur unseren Titel in das Artikelfeld der vorhandenen Artikel-tabelle einzufügen brauchen.

Wir teilen also SEBLOD an dieser Stelle mit, dass unser Newstitle identisch mit dem Titel eines Beitrags ist. Dazu wählen wir zunächst als *Format Standard* aus. Wir teilen SEBLOD mit, dass es sich um ein Feld handelt, das in einem *Article* verarbeitet wird. Und schließlich greifen Sie direkt auf die Tabellenspalte *title* zu, indem Sie entweder auf die kleinen unscheinbaren blauen Pfeile klicken und die Spalte auswählen oder den Namen *title* per Hand in das Feld eintragen. Besser ist es aber, die Auswahl zu benutzen, um die Fehlerquelle »falsche Eingabe« von vornherein zu umgehen.

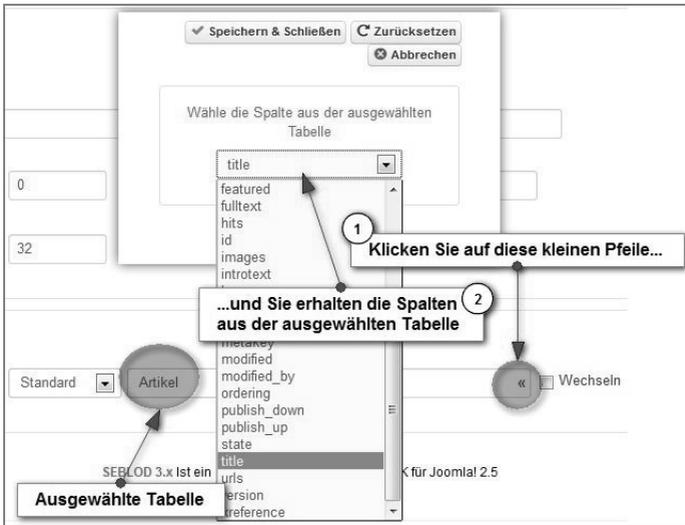


Bild 3.5: Auswahl der Datenbanktabelle.

Danach öffnet sich ein kleines Fenster, in dem Sie alle Tabellenspalten sehen, in diesem Fall die der Datenbanktabelle für *Artikel: #\_\_content*.

#### Platzhalter für das Datenbankpräfix

Die Zeichenkombination `#__` wird in Joomla! als Platzhalter für das Datenbankpräfix benutzt. Jede Joomla!-Datenbank hat ein eigenes Präfix, was ein guter Sicherheitsschutz ist. Denn kennt man das Präfix nicht, kann man nicht auf die Datenbanktabellen und deren Inhalte zugreifen. Meine Tabelle heißt `bdayo_content`. Ihre hat ein anderes Präfix, und damit Sie verstehen, wovon ich spreche, wird dieses Präfix als `#__` geschrieben. Die Tabelle selbst heißt dann `__content` (`PREFIX_content`). Zusammen mit dem Platzhalter ergeben sich zwei Unterstriche: `#__content`.

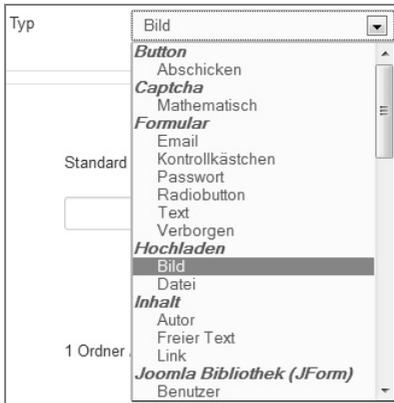
id	asset_id	title	alias	introtext	fulltext	state	catid	created
4	82	Blindtext mit Lorem ipsum	blindtext-mit-lorem-ipsum	<code>::cck::4::cck::&lt;br /&gt;::newsverweis:: ("link":"http...</code>	<code>&lt;br /&gt;::text:: &lt;p&gt;Lorem ipsum dolor sit amet, conse...</code>	1	2	2013-04-23
5	83	Neu Neu Neu!	neu-neu-neu	<code>::cck::5::cck::&lt;br /&gt;::newsverweis:: ("link":"http...</code>	<code>&lt;br /&gt;::text:: &lt;p&gt;Lorem ipsum dolor sit amet, conse...</code>	1	2	2013-04-23

Bild 3.6: Ein kleiner Auszug aus `#__content`. Hier finden wir später auch unsere News wieder.

Damit können SEBLOD und Joomla! später unsere News einwandfrei zuordnen.

## Feldtyp: Bild

Am besten nun auf *Speichern & Neu* klicken. Das nächste Feld soll ein Bild sein, und zwar eins, das wir hochladen können. Dazu wählen Sie bei *Typ* unter dem Abschnitt *Upload (Hochladen)* das Feld *Image (Bild)* aus.



**Bild 3.7:** Typauswahl: ein Bild zum Hochladen.

Der Bereich *Konstruktion* verändert sich nun immens im Vergleich zum Textfeld. Auch kommt ein neuer Bereich dazu: *Prozesse*.

Die *Bezeichnung* ist identisch mit dem Textfeld und gibt das Labelfeld/die Bezeichnung des Feldes an. Auch hier können Sie einen Standardwert eingeben: dazu einfach Pfad und Bildname eingeben, beispielsweise *images/bild.jpg*.

Im *Ordner* geben Sie an, wo SEBLOD die hochgeladenen Bilder ablegen soll. Sie können nun vorgeben, dass pro Benutzer ein eigenes Verzeichnis angelegt werden soll. Das ist sehr interessant, wenn Sie beispielsweise ein Redaktionssystem mit mehreren Autoren einrichten und jeder seinen eigenen Ordner mit seinen Bildern erhalten soll. Des Weiteren können Sie auch vorgeben, dass jeder Inhalt – also jeder Beitrag – einen eigenen Ordner erhalten oder alles in einem Verzeichnis abgelegt werden soll. Erwarten Sie viele Bilder, ist eine entsprechende Planung durchaus sinnvoll.

Die *Maximalgröße* gilt für das Hochladen eines Bildes. Machen Sie sich bewusst, dass alles, was in einem Beitrag angezeigt wird, zunächst in den Browsercache des Besuchers Ihrer Website geladen werden muss. Große Bilddateien nehmen Festplattenplatz beim Besucher weg und sorgen eventuell für lange Ladezeiten Ihrer Website. Besser ist es daher, die Grafiken vor dem Upload auf die richtige Größe anzupassen.

Das *Zeige Löschen*-Kästchen ist eine unscheinbare Checkbox, die bei Auswahl *Zeige* neben dem Bild-Upload angezeigt wird, wenn bei den News ein Bild hochgeladen wurde, also frühestens nach dem ersten Speichern. Aktivieren Sie das Kästchen, löschen Sie das vorhandene Bild.

Die *Größe* bezieht sich auch hier auf den Eingabebereich des Feldes. Es kann also sein, dass der korrekte Pfad nicht angezeigt wird, wenn Sie hier einen zu kleinen Wert auswählen. Die vorgegebenen 32 funktionieren aber bisher zuverlässig.

Bei den *erlaubten Erweiterungen* lässt sich einiges in den Optionen einstellen. Das voreingestellte *Image* bezieht sich auf alle Bilddateien. Details dazu werden weiter unten im Buch behandelt.

*Zeige Vorschau* ist sehr interessant. Damit ist gemeint, wie das Bild im Backend nach dem Editieren der News angezeigt werden soll. Sie können es ganz verbergen, lediglich den Bildnamen anzeigen (der bei Anklicken das Bild in einer Lightbox öffnet), ein Bildsymbol oder ein Bild in einer beliebigen Größe. Die Auswahl *Bild* stellt zunächst die Originalgröße des Bildes dar, sofern Sie das weiter unten bei den *Prozessen* nicht verändern.

Sollten Sie ein sehr großes Bild hochgeladen haben, wird die Gesamtgröße auch angezeigt – weshalb es sinnvoll sein kann, entweder das Bild auf eine Einheitsgröße zu bringen oder eines der Miniaturbilder auszuwählen. Bei den *Prozessen* können Sie bis zu zehn verschiedene Miniaturbilder angeben und die Größe selbst bestimmen. Es bleibt auch Ihnen überlassen, wie das Originalbild angepasst werden soll.

Bei der dynamischen Anpassung erscheint mir die Bildqualität ein wenig besser. Aber ich habe nicht das absolute grafische Auge, weshalb ich dafür nicht meine Hand ins berüchtigte Feuer legen mag. Auch sind einige Techniken der Bildanpassung in der Auswirkung gleich.



**Bild 3.8:** Das Original: 260 x 430.



**Bild 3.9:** *Zuschneiden Mitte:* 100 x 100.

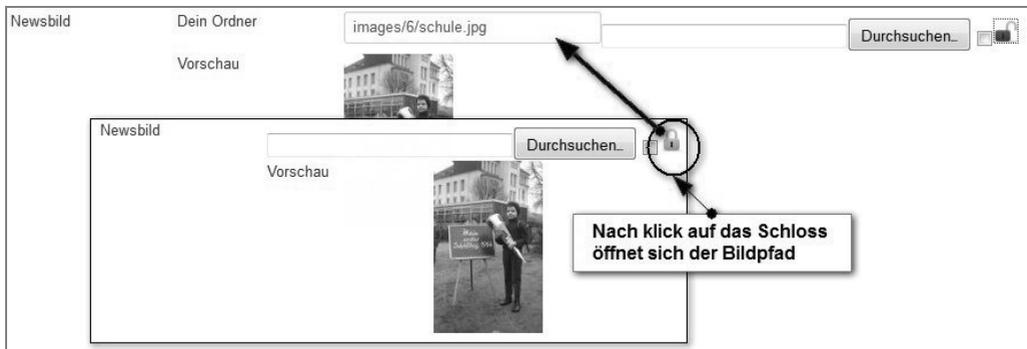


**Bild 3.10:** *Schrumpfen, Strecken, Dynamisch*  
*Schrumpfen, Dynamisch Strecken:* 100 x 100.



**Bild 3.11:** *Dynamisch: Max Anpassung* 100 x 100.

Wenn Sie *Zeige spezifischen Pfad* auf *Zeigen* stellen, wird Ihnen nach dem ersten Speichern der Pfad und der Bildname im Backend ausgegeben. Sie können ein Bezeichnungsfeld beschriften. In meinem Beispiel habe ich *Dein Ordner* hineingeschrieben.



**Bild 3.12:** *Zeige spezifischen Pfad.*

Der *erweiterte Modus* bietet unter anderem das Alternativattribut (*alt*-Attribut im HTML beim *img*-Element). Neben dem alternativen Text können Sie auch einen Titel beim Bild angeben. Zusätzlich zu eventuellen SEO<sup>5</sup>-Gründen ist das *alt*-Attribut im Zusammenhang mit Barrierefreiheit sehr wichtig.

#### Immer mit Alternativtext

Braillegeräte für Blinde können logischerweise keine Bilder vorlesen – sehr wohl aber den Alternativtext, weshalb dieser bei Grafiken, die nicht dem Design dienen, immer angegeben werden sollte.

Bei beiden Möglichkeiten können Sie noch einmal eine Unterauswahl treffen – also ob der Titel und oder der alternative Text angegeben werden soll – und dazu ein eigenes Beschriftungsfeld angeben.

Bei den *Prozessen* geben Sie zunächst an, wann das Miniaturbild erstellt werden soll. *Immer wenn nicht vorhanden* prüft zunächst, ob das Miniaturbild schon vorhanden ist. Das ist dann sinnvoll, wenn Sie mehrere Miniaturbildeinstellungen haben, die alle die gleiche Größe haben. Das Miniaturbild ist letztlich immer gleich, wird aber entsprechend angepasst.

Im obigen Beispiel meiner Einschulung ist das Miniaturbild immer 100 x 100 Pixel groß und wird durch Strecken, Schrumpfen etc. lediglich etwas anders angezeigt. Stelle ich hingegen *Immer* ein, wird stets ein neues Miniaturbild angelegt, egal ob schon vorhanden oder nicht. *Beim Hochladen* bietet nicht nur die beste Leistung, sondern erzeugt auch nur ein Miniaturbild. Allerdings muss die Größe dann immer gleich sein. Wird im Nachhinein die Größe verändert, muss das Bild erneut hochgeladen werden. Zumindest war das so bei meinen Tests.

*Anzeige als Standard* gibt an, welche Bildeinstellung für das Frontend verwendet werden soll.

Sie können *Bild* so stehen lassen und haben dann immer eine Einstellung für das Originalbild. Aber Sie können natürlich auch diese Größe anpassen und somit maximal elf individuelle Einstellungen für das immer gleiche Bild vornehmen. Der Vorteil liegt darin, dass Sie sich später eventuell für eine andere Größe entscheiden oder bei den verschiedenen Formularen jeweils andere Bildauflösungen angeben möchten. Vielleicht möchten Sie aber mit Abhängigkeiten oder Programmierungen auf verschiedene Bildauflösungen zugreifen. Die Miniaturbilder werden immer jeweils passend angelegt.

Im Bereich *Speicher* muss zunächst wieder *Standard* und auch unsere *Artikel-Tabelle* ausgewählt werden. Klicken Sie auf die beiden blauen Pfeile und wählen Sie die Spalte *images* aus. Alles speichern, und dann geht's an das nächste Feld für unsere *News-App*.

---

<sup>5</sup> Search Engine Optimization – Suchmaschinenoptimierung

### Feldtyp: WYSIWYG-Editor

Dieses Feld ist etwas überschaubarer als das *Bild*-Feld. Aber ein paar interessante Dinge können wir auch hier einstellen. Unter *Typ* finden Sie den *WYSIWYG<sup>6</sup>-Editor* unter dem Abschnitt *Textbereich*.

Der *Standard Wert* ist hier schon vorgegeben – mit dem Editor. Den können Sie sich in der ganzen Pracht anschauen, wenn Sie auf den *Editor*-Button klicken. Welchen Editor Sie benutzen möchten, können Sie direkt darunter auswählen. Standardmäßig hat Joomla! den TinyMCE und den Code Mirror installiert. Haben Sie andere Editoren installiert, werden diese hier auch angezeigt. Alternativ können Sie *Kein* auswählen. Allerdings hat dann der Feldtyp nicht so viel Sinn, zumal es als Feldtyp auch einen *Textbereich* gibt, der keinen Editor hat.

Unter *Ort* legen Sie fest, wie der Editor und somit der Eingabebereich angezeigt werden: entweder als modales Fenster – das ist gleichbedeutend mit dem Button *Editor* oben rechts, wo Sie eine Beschreibung zu dem Feld eingeben können, oder Sie wählen *Standard* aus – das ist so, wie Sie es kennen bei Joomla!: Eingabe direkt.

Bei *Breite/Höhe* können Sie die Größe des Editors angeben.

Der *Importierer* gibt Ihnen die Möglichkeit, festzulegen, ob immer der *Volltext* angezeigt wird oder ein *Intro<sup>7</sup>/Teaser<sup>8</sup>* und erst nach Anklicken des *Weiterlesen*-Buttons der ganze Text. Das setzt natürlich voraus, dass Sie Ihren Text entsprechend gestalten.

*Speichern* kennen Sie schon, als Format wählen Sie *Standard*, in der *Artikel-Tabelle* und mittels der Spaltenauswahl finden Sie die Spalte *fulltext*.

#### Joomla!-Editor JCE

Mit dem kostenlosen Joomla!-Editor JCE<sup>9</sup> können Sie verschiedene Profile anlegen und Ihren Autoren sehr individuelle Editoroberflächen zur Verfügung stellen.

### Feldtyp: Link

Den Feldtyp *Link* finden Sie im Abschnitt *Inhalt*, der relativ weit oben steht. Im Backend geben Sie bei den News erst den Verweis ein, dann, mit welchem Text der Verweis dargestellt werden soll, und wählen dann aus, wie der Verweis aufgerufen werden soll.

<sup>6</sup> WYSIWYG, englische Abkürzung für »What You See Is What You Get« (Übersetzung: Was du siehst, ist, was du bekommst)

<sup>7</sup> Intro = kurzer Einführungstext

<sup>8</sup> Teaser = Anreißertext

<sup>9</sup> <http://www.joomlacontenteditor.net>

Newsverweis

Link

Verweisname

Ziel

Vorschau time4mambo

**Nach dem ersten Speichern wird eine Vorschau angezeigt**

Bild 3.13:  
Linkeingabe im Backend.

Unter *Zeige Link* geben Sie die Bezeichnung für das Linkeingabefeld ein. Sie können den Link so anzeigen, wie er eingegeben wird – also als vollständige URL oder stattdessen mit einem Verweisnamen. In *Zeige Text* wählen Sie aus, ob der Verweisname benutzt werden soll (*Zeige*) oder die URL (*Verberge*). Darunter können Sie eine Bezeichnung für dieses Feld eingeben, sofern Sie *Zeige* ausgewählt haben. Auch können Sie dann einen Verweisnamen als *Standard Text* vorgeben. Ebenso können Sie hier einen Standardlink vorgeben.

Mit *Zeige Ziel* ist *target*<sup>10</sup> gemeint, also wie sich der Link verhält, wenn er angeklickt wird.

- *\_blank* öffnet ein neues Browserfenster.
- *\_self* öffnet den Verweis im aktuellen Fenster.
- *\_parent* und *\_top* wurden in Framesets, die heute normalerweise keine Anwendung mehr finden, benutzt, um das ganze oder Teile des Framesets zu »sprengen«. Sie bewirken außerhalb von Framesets das Gleiche wie *\_self*.

Wenn Sie *Zeige Ziel* auf *Zeige* gesetzt haben, können Sie später das Ziel ändern. Andernfalls wird das *Standard Ziel* benutzt.

Sie können bei den Links eine spezielle CSS-Klasse angeben. Die CSS-Klasse können Sie dann in der CSS-Datei des Templates formatieren. Angenommen, Sie haben die Klasse *schatten* vergeben, dann können Sie in der CSS-Datei den Link folgendermaßen ansprechen:

```
001 a.schatten { ... }
002 a.schatten:link { ... }
003 a.schatten:visited { ... }
004 a.schatten:hover { ... }
005 a.schatten:active { ... }
```

<sup>10</sup> *target* ist ein Attribut im HTML-Element *a*: `<a href="..." target="_blank">`.

Sie können sich das Feld für die Klasse auch anzeigen lassen (*Zeige Klasse*), um für bestimmte News einem Verweis eine eigene Klasse zuzuordnen, den Sie dann mittels CSS individuell gestalten können.

Unter *Zeige Vorschau* können Sie einstellen, ob eine Vorschau angezeigt wird oder nicht. Die Vorschau ist natürlich erst nach dem ersten Speichern sichtbar.

*Größe*, *Mindestlänge* und *Maximallänge* sind analog zum Textfeld.

Beim *Speichern* weichen wir nun aber ein wenig vom Standard ab. Als Erstes wählen wir hier das Format *Benutzerdefiniert*. Wir bleiben allerdings in der *Artikel-Tabelle*. Wenn Sie nun die Spaltenauswahl öffnen, werden Sie keine Spalte *Links* finden. SEBLOD benutzt für alles, was nicht sauber zuzuordnen ist, meist die Spalte *introtex*, die Bestandteil der *Artikel-Tabelle* ist. Das wählen wir auch aus.

### Wichtig!

Wenn Sie hier nicht *Benutzerdefiniert*, sondern *Standard* auswählen, werden die Links nicht gespeichert.

## 3.1.3 Die Felder im Überblick

Das waren die vier Felder, die durchaus schon ein wenig gezeigt haben, was in SEBLOD möglich ist. Hier noch einmal ein Überblick über unsere Felder:

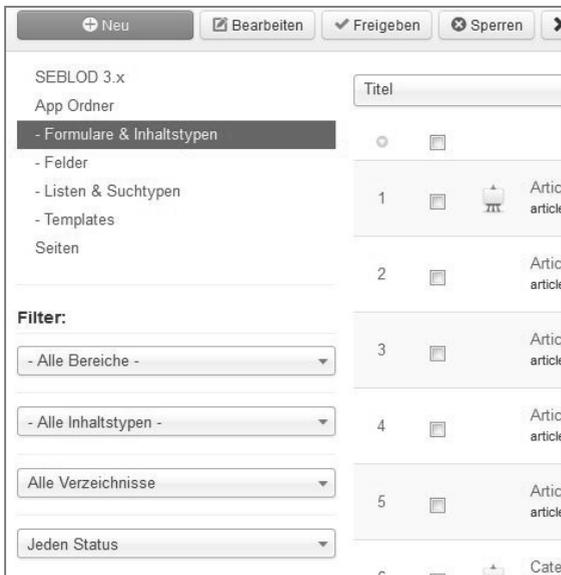
		Titel		App Ordner	# Speicher	Typ
1		News-Verweis newsverweis		News TOP	Custom	Link
2		Newsbild newsbild		News TOP	Standard	Hochladen - Bild
3		Text text		News TOP	Standard	WYSIWYG-Editor
4		Titel titel		News TOP	Standard	Text

**Bild 3.14:** Unsere vier Felder für unsere News im Überblick.

Nun erstellen wir etwas zum Eingeben und Ausgeben.

## 3.1.4 Eine Form muss her

Gehen Sie nun zu *Formulare & Inhaltstypen*. Es wird Zeit, dass wir auch sehen können, was wir angelegt haben. In der Übersicht sehen Sie die vorangelegten Joomla!-/SEBLOD-Inhaltstypen *Article*, *Category*, *User* und *User Group*. Noch stehen unsere *News* nicht in der Auflistung. Das werden wir aber gleich ändern.



**Bild 3.15:** Die Filterung in der Übersicht der Inhaltstypen findet und offenbart die News.

Wenn Sie später viele Apps in der Auflistung haben, können Sie an dieser Stelle links die Übersicht filtern. Sie können auch eine Menge Apps finden, die Sie bisher eventuell noch gar nicht wirklich wahrgenommen haben. Die vielen Apps im *App Ordner* finden Sie hier wieder – je nachdem, wie sie angelegt sind und welche Aufgabe sie haben. Ordnung in dieses Verwirrspiel wird weiter unten gebracht. An dieser Stelle empfehle ich Ihnen, den *Status* im letzten Filterfeld auf *An* zu setzen und im vorletzten Filter auf *Alle Verzeichnisse*. Eigentlich sollte das so voreingestellt sein.

Klicken Sie nun oben links auf *Neu*, damit Sie Ihren eigenen Inhaltstyp anlegen, und wählen Sie dann *News* aus. Wenn Sie beim Erstellen der App ein eigenes Symbol angegeben haben, erscheint das hier zur Auswahl, ansonsten sehen Sie den Standardbutton. Nach der Auswahl benötigt SEBLOD stets ein paar Sekunden, bis das Fenster komplett aufgebaut ist. Oben rechts sehen Sie einen leicht grauen Ladebalken, der aber wirklich schwer zu erkennen ist.

Was Sie in *Titel* eintragen, ist das, was Sie später bei der Beitragsübersicht angezeigt bekommen, wenn Sie eine neue News eingeben wollen. In meinem Beispiel habe ich auch *News* eingetragen.

Sie müssen nun, obwohl Sie bei der Auswahl eindeutig *News* angeklickt haben, im *App Ordner* explizit die *News* auswählen. Erst dann werden die Formulare, die wir jetzt erstellen werden, auch unserer App eindeutig zugeordnet. Es wäre auch denkbar, dass Sie ein Formular mit den Newsfeldern in einer anderen App erstellen, weshalb Sie hier die zugehörige App auswählen müssen. In unserem Fall ist das auch unsere *News-App*.

Darunter sehen Sie im linken Teil vier Formulärmöglichkeiten zur Ein- und Ausgabe: *Admin Formular*, *Seitenformular*, *Intro* und *Inhalt*. Die drei rechten Buttons (*Konfiguration*, *Felder* und *Template*) sehen wir uns etwas später an. Standardmäßig sollte *Felder* ausgewählt sein. Zunächst beschäftigen wir uns mit den Formularen. Bevor wir dort tiefer einsteigen, zunächst ein kleines Kapitel über die Handhabung dieser Ansicht.

### Handhabung der Formulare & Inhaltstypen

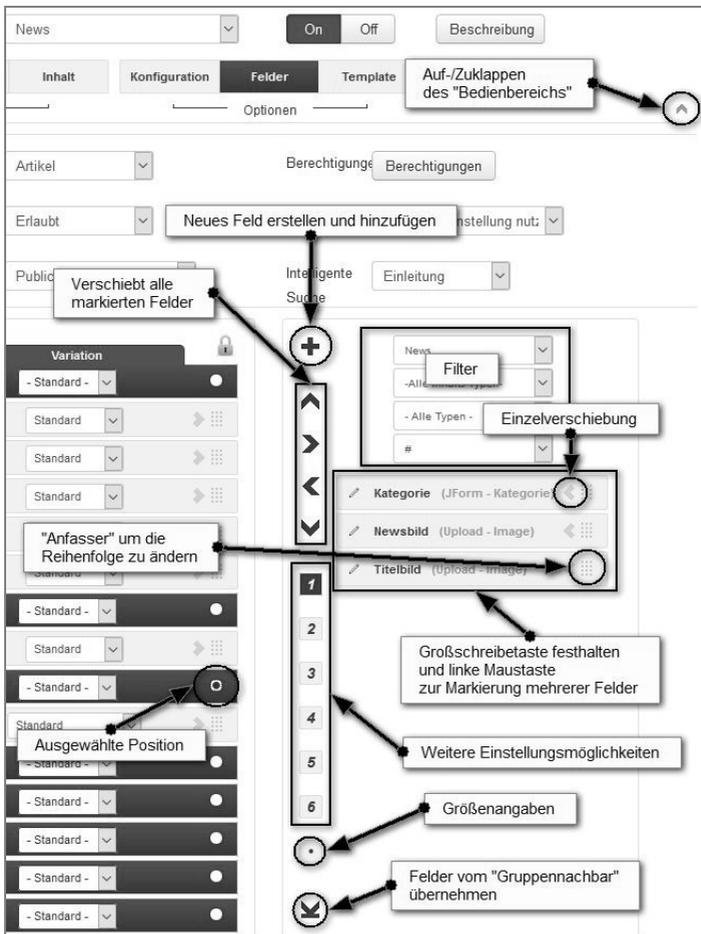
Sie werden einen großen Teil Ihrer Zeit mit SEBLOD an dieser Stelle zubringen, weshalb Sie diesen Bereich auch einwandfrei bedienen können sollten. Es gibt hier einige sehr spannende Dinge zu entdecken.

Bei den Feldern ist Ihnen vielleicht schon der kleine, orangefarbene, nach unten weisende Pfeil aufgefallen, wie Sie ihn im *Speicher*-Bereich finden. Bei den Inhaltstypen haben Sie ihn weiter oben und können hier Einstellungen zu Bedienung, Art und Suche des Inhaltstyps vornehmen. Hier finden Sie auch den *Alias*. Im rechten Bereich der *Konstruktion* finden Sie verschiedene Filter.

- Mit dem Filter, der standardmäßig mit *Alle Typen* angezeigt wird, können Sie gezielt bestimmte Typen anzeigen lassen.
- Mit dem Rautenfilter (#) listen Sie alle Felder auf, die mit dem ausgewählten Anfangsbuchstaben beginnen.
- Sie können auch gezielt in den bereits vorhandenen *Formularen & Inhaltstypen* suchen.

Voreingestellt ist bei uns im vierten Filterfeld die *News*, weshalb vermutlich alle anderen Filter derzeit keinen nennenswerten Erfolg haben werden. Sie können natürlich die einzelnen Filter beliebig miteinander kombinieren. Ich verspreche Ihnen, dass Sie dies einige Male gebrauchen werden.

Um Felder zu verschieben, gibt es verschiedene Möglichkeiten. Sie können mehrere Felder markieren, indem Sie die Großschreibtaste festhalten und mit der linken Maustaste die einzelnen Felder anklicken. Danach können Sie mit den Pfeilbuttons diese Felder entsprechend verschieben. Nach *rechts* werden die Felder aus dem Formular in die Übersicht verschoben, nach *links* die Felder aus der Übersicht in den Bereich des Formulars, den Sie mit einem kleinen Punkt markiert haben. Beim *#MAINBODY* beispielsweise ist ganz rechts ein kleiner Radiobutton, so auch in den anderen Positionsbereichen. Markierte Felder, die Sie nach unten oder oben bewegen möchten, werden dorthin verschoben, wo Sie den Punkt gesetzt haben.



**Bild 3.16:** Schieberei bei den Formularen.

Sie können aber auch die Felder einzeln verschieben, wenn Sie den Pfeil im Feld anklicken. Der ist zunächst unscheinbar grau, füllt sich aber mit Farbe, wenn Sie die Maus darüber bewegen. Direkt daneben haben Sie einen Anfasser, um das Feld bei gedrückter linker Maustaste frei in der Reihenfolge zu bewegen. Allerdings können Sie so kein Feld von links nach rechts bewegen, wohl aber von der Übersicht in das Formular.

Mit dem Plusbutton können Sie jederzeit ein neues Feld anlegen. Wir hätten also unsere vier Felder auch an dieser Stelle anlegen können. Allerdings ist gerade bei größeren Apps mit vielen Feldern die Übersicht im Menü *Felder* besser als an dieser Stelle. Außerdem müssen Sie hier stets ein wenig im Modalfenster scrollen, und mir ist es auch schon passiert, dass einmal etwas abgeschnitten wurde. Aber wenn Sie bemerken, Sie haben ein Feld vergessen oder möchten Ihr Formular um ein, zwei Felder erweitern, dann ist das hier durchaus eine praktikable Lösung.

Die Nummern bieten verschiedenste Einstellungsmöglichkeiten und verändern jeweils die Formularübersicht links etwas. Mit der *Nummer 3* können Sie beispielsweise angeben, ob das Feld *erforderlich* oder *optional* ist.

Mit dem Punkt am Ende der Zahlenreihe (1 bis 6) können Sie die Größe der einzelnen Positionen ändern und zusätzliche Angaben zu den Formularbereichen machen.

Mit dem Pfeil nach unten übernehmen Sie das „Vorformular“. Im Seitenformular können Sie so alle Felder aus dem Adminformular übernehmen und müssen nicht noch einmal alles neu anlegen. Sie können aber dann natürlich das Seitenformular individuell gestalten. Umgekehrt können Sie im Adminformular die Felder aus dem Seitenformular übernehmen, allerdings jeweils nur die Felder und nicht bereits angelegte Strukturen. Gleiches gilt für Einleitung und Inhalt. Der Pfeil nach unten betrachtet die Bereiche als zwei Zweiergruppen, die untereinander in beide Richtungen ihre Felder weitergeben, aber nicht die beiden Gruppen. Also können Sie keine Felder aus dem Eingabebereich auf den Ausgabebereich übertragen.

Sie können natürlich auch die einzelnen Felder überarbeiten. Dazu einfach den Stift ganz vorn im jeweiligen Feld anklicken.



**Bild 3.17:** Felder aus dem Formular heraus editieren.

### Das Geheimnis des Schlosses

Das Schloss ist nur und ausschließlich im Zusammenhang mit Feldern, die Sie aus dieser Ansicht heraus neu anlegen, interessant. Auf bereits vorhandene Felder hat das Schloss keinen Einfluss.



**Bild 3.18:** Geschlossenes Schloss und neues Feld ergeben ein exklusives Feld.

Wenn Sie ein neues Feld mit geschlossenem Schloss anlegen und speichern, erhalten Sie quasi ein exklusives Feld, das nur und ausschließlich in der geöffneten Inhaltsform sichtbar ist.



**Bild 3.19:** Das exklusive Feld ist nur in diesem Inhaltstyp sichtbar.

Wenn Sie also einen anderen Inhaltstyp anlegen, können Sie auf dieses Feld nicht so ohne Weiteres zugreifen. Es wird nicht aufgelistet, egal wie Sie den Filter einstellen. Öffnen Sie hingegen das Schloss und legen dann ein neues Feld an, können Sie in jedem Inhaltstyp auf dieses Feld zugreifen.



**Bild 3.20:** Offenes Schloss = neues Feld für alle Inhaltstypen.

### Die Formulararten im Überblick

- Das *Admin Formular* ist die Eingabemaske im Backend, die Sie standardmäßig ab der Rechtegruppe *Moderator* erreichen können, also die Maske, die Sie normalerweise von den Joomla!-Beiträgen im Backend kennen.
- Das *Seitenformular* rufen Sie im Frontend auf, wenn Sie eine News editieren oder als Autor einen neuen Beitrag/eine neue News einreichen.
- *Intro* ist das, was zum Beispiel bei Blogbeiträgen angezeigt wird, bevor Sie auf *Weiterlesen* klicken. Der einzelne Blogbeitrag ist identisch mit dem *Intro*.
- Der ganze Beitrag mit dem *Inhalt*.

### Eingabe der News: Admin Formular

Links sind die Positionen des Templates. Ganz oben in der horizontalen Buttonzeile haben Sie einen Button mit der Beschriftung *Positionen*. Sie sehen unter dem Button Positionen, die nicht angezeigt werden, beispielsweise alle *#LEFT-* oder *#HEADER-*Positionen. Wie wir an die herankommen, erkläre ich etwas später, wenn wir die Ausgabe unserer News im Frontend aufhübschen.

Sie erstellen hier das Formular zur Eingabe Ihrer News im Backend, was Sie dann unter *Beiträge* auswählen können. Die Erstellung ist denkbar einfach: Sie schieben einfach die Felder, die Sie angezeigt haben möchten, an die verschiedenen Positionen. Wichtig ist dabei lediglich, dass Sie mindestens ein Feld im *#MAINBODY* stehen haben, da sonst nichts angezeigt wird.

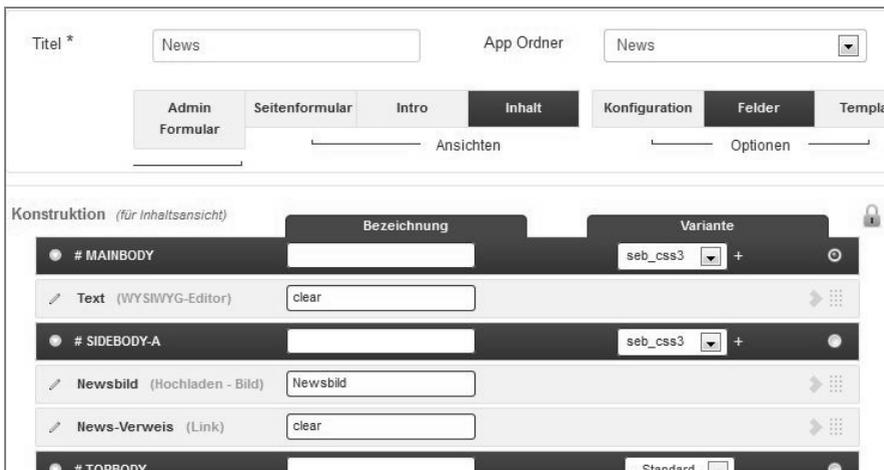
Wenn Sie einen neuen Inhaltstyp anlegen, werden vermutlich bereits alle Felder im *#MAINBODY* stehen. Sie können aber diese Felder beliebig anders anordnen und »herumschieben«. Andere Positionen bedeuten eine andere Anordnung. Beispielsweise können Sie die Felder nebeneinander ausgeben, wenn Sie die oberen oder

unteren Positionen entsprechend belegen. Später werden Sie sehen, dass Sie die einzelnen Positionen auf unterschiedliche Art arrangieren und so gerade auch bei größeren Inhaltstypen alles ein wenig übersichtlicher gestalten können.

Das war's an dieser Stelle zunächst. In dieser ersten App belasse ich es beim *Admin Formular*.

### Ausgabe im Frontend: Inhalt

Zur Anzeige im Frontend wechseln Sie oben auf *Inhalt*. Diesen Inhalt sehen Sie, wenn Sie den Menüverweis aufrufen, der die News anzeigt.



**Bild 3.21:** Frontend-Ausgabe.

Für die Ausgabe benötigen wir nicht alle Felder. Es reichen *Textfeld*, *Newsbild* und *News-Verweis*. Den *Text* positioniere ich im *#MAINBODY* und die beiden anderen Felder im *#SIDEBODY-A*. Wenn Sie das jetzt aufrufen, stehen der Verweis und das Bild rechts neben dem Text. Ich möchte es aber links vom Text haben. Damit das klappt, klicken Sie oben rechts in den *Optionen* auf *Template*.



**Bild 3.22:** Um die Positionen im Template einzustellen, zuerst das *Template* auswählen ...

Hier gibt es eine Menge zum Einstellen. Für uns ist zunächst im Bereich *Positionen* die *Linke Spalte (Breite)* interessant. Tragen Sie dort *200 px* ein. Zwei Zeilen darunter geben Sie an, dass der *Seitenteil-a* links stehen soll. Für die *Rechte Spalte (Breite)* können Sie eine *0* eingeben. Für unsere Ausgabe reicht das so.

⇒ Positionen

Standard Variante	seb_css3	Rand	8 px
Kopfzeile	Tiefste	Kopfzeilenvariante	
Linke Spalte (Breite)	200 px	Linke Spalte Variante	
Oberste Zeile	Nein	Oberste Zeilenvariante	
Seitenteil-a	Links	Seitenteil-b	Rechts
Unterste Zeile	Automatisch	Unterste Zeilenvariante	
Rechte Spalte (Breite)	0 px	Rechte Spaltenvariante	
Fußzeile	Nein	Fußzeilenvariante	

**Bild 3.23:** ... und dann die Positionen im Detail ansprechen.

Wenn Sie Werte für die *Kopfzeile* oder die *Fußzeile* eingeben, sehen Sie in der Formularübersicht auch die Positionen für *#HEADER* und *#BOTTOM*.

Wechseln Sie wieder zu den *Feldern* zurück, sehen Sie auch die Positionen für *#LEFT*. Wenn Sie in der *Rechten Spalte (Breite)* eine *0* eingetragen haben, sehen Sie die Positionen für *#RIGHT* nicht mehr. Andernfalls wird Ihre Positionenliste entsprechend länger.

Ein paar Kleinigkeiten fehlen noch: Momentan würden in der Ausgabe auch die Label-/Bezeichnungstexte angezeigt werden. Um sie auszuschalten, gibt es zwei Möglichkeiten.

- 1 Wollen Sie nur einzelne Beschreibungstexte nicht anzeigen, geben Sie im jeweiligen Bezeichnungsfeld ein *clear* ein.

Newsbild (Hochladen - Bild)	Newsbild
News-Verweis (Link)	clear

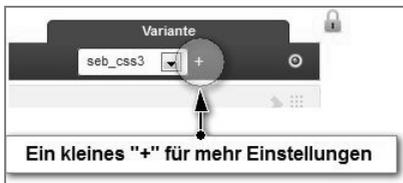
**Bild 3.24:** Einmal mit und einmal ohne Ausgabe des Bezeichnungsfeldes.

- 2 Sie wechseln noch einmal in die *Template-Optionen* und setzen im Bereich *Felder* die *Zeige Bezeichnung* auf *Verberge*. Damit setzen Sie alle Bezeichnungsfelder auf nicht anzeigen und brauchen nicht überall ein *clear* einzugeben.



**Bild 3.25:** Bezeichnungsfelder werden nicht angezeigt.

Wenn Sie einen Rahmen oder farbliche Absetzungen etc. in die Ausgabe einarbeiten möchten, setzen Sie zuerst bei den Feldern in der gewünschten Position unter der Spalte die *Variante seb\_css3* und klicken dann auf das kleine Pluszeichen, das daneben erscheint.



**Bild 3.26:** Mit dem Pluszeichen das Aussehen im Frontend gestalten.

Hier finden sich allerlei interessante Einstellungsmöglichkeiten. So können Sie die Hintergrundfarbe für den Positionsbereich ebenso bestimmen wie Rahmenfarbe, -stärke und -radius. Haben Sie mehrere Positionen nebeneinander, können Sie den Abstand der einzelnen Rahmen auch mit angeben.

Sie können hier auch explizit das Anzeigen der *Bezeichnung* für einen bestimmten Bereich aktivieren, wenn Sie im *Template* die *Bezeichnungen* deaktiviert haben. Das stellen Sie entsprechend im Bereich *Felder* ein: *Zeige Bezeichnung. Nutze Allgemein* ist die globale Einstellung beim *Template*. Was man hier noch einstellen kann – dazu später mehr.

### Speichern & Schließen

Klicken Sie jetzt auf *Speichern & Schließen*. Wichtig ist dabei, das ganze Formular zu speichern. Erst dann wird all das übernommen, was Sie in den einzelnen Einstellungsfenstern eingestellt haben.

In der Übersicht steht nun auch unsere App. Die einzelnen Spalten geben Ihnen unter anderem Informationen darüber, wie viele Felder Sie für die verschiedenen Formulare verwendet haben.

			Titel	App Ordner	# Admin Formular	# Seitenformular	# Intro	# Inhalt	Status	ID
1			Article article	 Article JOOMLA	15	14	1	2	 	1
2			Category category	 Category JOOMLA	12	10	1	1		5
3			News news	 News TOP	4	-	-	3	 	501
4			User user	 User JOOMLA	12	6	-	-		8
5			User Group user_group	 User Group JOOMLA	3	-	-	-		11

**Bild 3.27:** Die News in der Übersicht *Formulare & Inhaltstypen*.

### 3.1.5 Ein- und Ausgabe einer News

Jetzt wird es Zeit, dass wir auch sehen, was wir die ganze Zeit erstellt haben. Dazu gehen Sie bitte in die Beitragsübersicht. Dort klicken Sie auf *Neu* oben links. In dem erscheinenden Auswahlfenster sehen Sie standardmäßig immer die *Article-Auswahl*. Allerdings handelt es sich hier bereits um eine von SEBLOD angelegte Form, die zwar inhaltlich identisch mit der Joomla!-eigenen Artikeleingabe ist, aber anders aussieht.

Wenn Sie zur Joomla!-eigenen Beitragseingabe zurückkehren möchten, dann klicken Sie einfach diesen Link an:

**001** oder hier klicken zum Erstellen eines "Standard Joomla! Inhalts".

Da wir aber eine News eingeben wollen, klicken Sie das Symbol mit den News an.

The screenshot shows the Joomla! administration interface for adding a new news item. At the top, the Joomla! logo and the title 'Neues hinzufügen News' are visible. Below the title are four buttons: 'Speichern', 'Speichern & Schließen', 'Speichern & Neu', and 'Abbrechen'. The form contains several input fields: 'Newstitel', 'Newsbild' (with a 'Durchsuchen...' button), and a large 'News' field with a rich text editor toolbar. Below the editor is a 'Pfad: p' field. At the bottom, there are buttons for 'Beiträge' and 'Bild', and a table for 'Newsverweis' with fields for 'Link', 'Verweisname', 'Klasse', and 'Ziel'.

**Bild 3.28:** Das Adminformular zum Eingeben einer News.

Sie sehen nun das Formular, das Sie zuvor bei den *Formularen & Inhaltstypen* für das *Admin Formular* gestaltet haben. Geben Sie ein wenig Text und alles Weitere ein und speichern Sie Ihre News. In der Beitragsübersicht sehen Sie dann Ihre News mit Ihrem Titel. Geben Sie an, dass es sich um einen Hauptbeitrag handelt, indem Sie aus dem weißen Stern einen orangefarbenen machen.



**Bild 3.29:** Hauptbeitrag erzeugen.

Rufen Sie nun das Frontend auf, sehen Sie im Startmenü Ihre News, vorausgesetzt, Sie haben die Startseite auf *Hauptbeiträge anzeigen* gesetzt. Andernfalls müssen Sie ein entsprechendes Menü erzeugen.

## 3.2 Die News erweitern

Nun haben wir einen kleinen detaillierten Einblick bekommen. Aber wenn wir ehrlich sind, dann lassen unsere News doch so einiges vermissen. Zum Beispiel kann es nicht angehen, dass wir die News nur als Haupteintrag abbilden und auch erst in der Übersicht selbstigen auswählen können. Also liegt es nahe, einige Erweiterungen vorzunehmen.

### 3.2.1 Haupteintrag und Kategorie

Diesmal machen wir es uns einfach. Da SEBLOD uns alle Felder zur Verfügung stellt, die in einem Beitrag vorhanden sind, können wir diese Felder für unsere News benutzen.

Editieren Sie die News unter *Formulare & Inhaltstypen*. Bei den Filtern stellen Sie an dritter Stelle *Article* ein und im letzten Feld, wo standardmäßig unsere News angezeigt werden, wählen Sie bitte *Alle Verzeichnisse* aus. Sie erhalten nun eine Liste aller Felder für die Beiträge. An dieser Stelle müssen Sie die englischen Bezeichnungen kennen. Die Hauptbeiträge werden im Englischen als *Article Featured* bezeichnet. Genau dieses Feld schieben wir nach links in unser Formular und editieren es dort.

Wie Sie sehen, handelt es sich um ein einfaches Auswahlfeld, das aber entsprechend in der Datenbank an der entscheidenden Stelle gespeichert wird. Bei den *Optionen* sehen Sie, dass dem *No* eine *0* und dem *Yes* eine *1* zugeordnet ist. Da wir standardmäßig den *Haupteintrag* aktiviert haben möchten, tragen wir in das Feld *Standard Wert* eine *1* ein.

Wenn wir dann eine neue News anlegen, ist der *Hauptbeitrag* bereits auf *Ja* gesetzt – das kann aber geändert werden. Natürlich können Sie aus *Yes* und *No* auch *Ja* und *Nein* machen. Wenn Sie einen Text in *Bezeichnung* eingeben, wird Ihnen über der *Ja-/Nein-Auswahl* noch ein Text im Auswahlbutton angezeigt. Die *Anordnung* ergibt bei uns keinen Sinn. Bei längeren eigenen Listen können Sie dort aber einstellen, wie die Auswahlliste sortiert werden soll.

Da wir gerade in unserer *Admin Form* sind, fügen wir auch gleich ein Kategoriefeld hinzu. Auch hier können Sie wieder das entsprechende Feld aus der *Article-Liste* auswählen. Editieren Sie bitte das Feld, nachdem Sie einen Blick in Ihre Kategorien geworfen haben. Falls Sie nicht ohnehin bereits eine Kategorie namens *News* angelegt haben, tragen Sie bitte diese Kategorie ein und merken sich die ID der Kategorie.

Sie sehen, dass der Feldtyp *Kategorie aus der Joomla Bibliothek* ist. Auch hier können Sie wieder einen *Standard Wert* vergeben. Allerdings hat das an dieser Stelle eine

kleine Tücke. Denn wenn Sie den Standardwert mit der Kategorie-ID versehen (bei mir hat die Kategorie *News* die ID 10), würde das fortan für alle Kategorien gelten, wo Sie das Kategoriefeld einfügen. Überall wird also die Kategorie *News* voreingestellt sein.

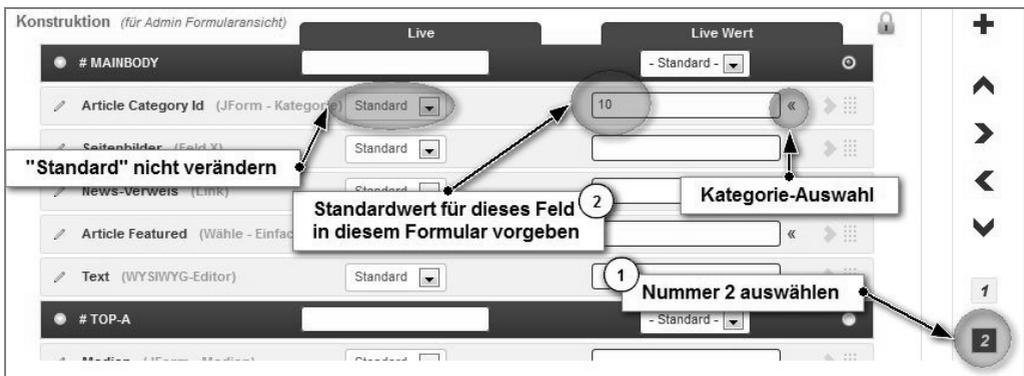
In unserem Fall wäre das nicht sinnvoll, weshalb wir hier den *Standardwert* leer lassen und stattdessen nach dem *Speichern & Schließen* auf die *Nummer 2* klicken: *Live + Live Wert*. In der Zeile unseres Kategoriefeldes (*Article Category Id*) lassen Sie das *Standard* stehen und schreiben in das leere Feld rechts die Kategorie-ID. Bei mir ist das die 10.

Sie können aber auch die Kategorie mittels des kleinen blauen Pfeilbuttons direkt auswählen. Damit haben Sie als *Standard* die *News-Kategorie* vorgegeben, aber nur für dieses eine konkrete Feld. Immer wenn Sie also im Backend eine neue News anlegen, erscheint die hier eingetragene Vorauswahl. Benutzen Sie hingegen das Kategoriefeld in einem anderen Inhaltstyp, gilt dieser *Live Wert* nicht mehr.

### Mehrfachauswahl erstellen

Im Kategoriefeld gibt es zwar die Möglichkeit, eine Mehrfachauswahl einzustellen, doch leider funktionierte das bis zum Erscheinen dieses Buches nicht. Der Versuch einer Mehrfacheinstellung führte immer zu Fehlern. Aber es ist denkbar, dass es in einer späteren Version von SEBLOD möglich ist.

Das englische *Category* können Sie natürlich sprachlich anpassen.



**Bild 3.30:** Kategorie-ID als Standardwert nur in diesem Inhaltstyp vorgeben.

Im Formular für den *Inhalt* brauchen wir nichts zu ändern, da die Kategorie und der Haupteintrag nur für die Verarbeitung unserer News wichtig sind, aber keinerlei zusätzliche Ausgabe im Frontend erzeugen.

Eine Anmerkung noch zu den Feldern aus der *Article*-App: Obwohl Sie die Felder unserer *News*-App hinzugefügt haben, bleiben die Felder selber den *Articles* zuge-

ordnet. Entscheidend ist am Ende, dass die Daten, die Sie dort beim Erstellen einer neuen News eingeben, innerhalb unseres Beitrags in der Datenbank gespeichert werden, was Sie dann auch tun. Brauchen Sie später die Felder noch einmal an anderer Stelle, finden Sie sie problemlos in der *Article*-App wieder.

### 3.2.2 Bilder sagen mehr als tausend Worte

Ergo sollten wir diesbezüglich auch mehr anbieten. Zwar können wir generell die Bildfunktionen des WYSIWYG-Editors nutzen, aber wir möchten vielleicht mehr Bilder in die linke Spalte bringen, und ein kleines Bild direkt links neben der Überschrift wäre vielleicht auch eine interessante Idee.



**Bild 3.31:** Die News erweitern.

Beginnen wir zunächst mit der Überschrift. Das bisher verwendete Bilderfeld hat den Nachteil, dass wir stets eine Grafik von unserer Festplatte hochladen müssen. Ich möchte in diesem Fall aber auf das *images*-Verzeichnis von Joomla! zugreifen, da ich mit der Überschrift noch etwas mehr vorhabe, wie Sie gleich sehen werden.

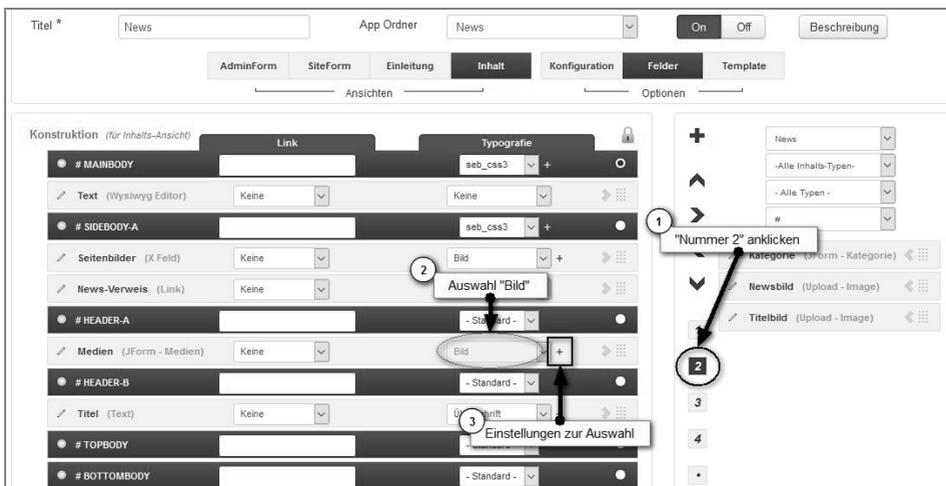
Der Einfachheit sollten Sie jetzt direkt in *Formulare & Inhaltstypen* arbeiten, indem Sie das *Pluszeichen (+)* für neue Felder anklicken, denn wir müssen das eine oder andere gleich in den Formularen einstellen.

Legen Sie also bitte im *Admin Formular* ein neues Feld vom Typ *Medien* an. Medien finden Sie in der Joomla!-Bibliothek (JForms). Allzu viel gibt es hier nicht einzustellen. Den *Standard Wert* lassen Sie leer. Speichern müssen Sie das Feld als *Standard* in den *Artikeln* und auch als *images*.

Das Bild soll am Anfang der Überschrift stehen und muss in der Größe angepasst werden. Leider geht das nicht direkt beim Feld, so wie Sie es bisher gewohnt sind.

Wechseln Sie in das Formular für den *Inhalt* und klicken die *Nummer 2* an. Sie können jetzt die Typografie für die Ausgabe genauer einstellen.

Wählen Sie dazu bei dem Feld *Medien* in der Spalte *Typografie Bild* aus. Es erscheint sofort ein kleines Pluszeichen (+) neben der Auswahl, worauf Sie bitte klicken. In dem Fenster, das sich nun öffnet, können Sie die Bildgröße einstellen. Wir wählen als *Standard-Anzeige Miniaturbild 1* und geben eine feste Größe von *55x55px* vor. Darunter können Sie noch einstellen, ob beim Anklicken des Bildes eine Modalbox aufgehen soll oder nicht. Da es sich um ein kleineres Bild neben der Überschrift handeln soll, ist ein *Kein* in der Modalboxauswahl sinnvoll.



**Bild 3.32:** Neben der Bildgröße ist es auch wichtig, SEBLOD mitzuteilen, dass es sich bei der *Medien*-Auswahl um ein *Bild* handelt. Denn sonst wird zwar der ausgewählte Grafikpfad im Frontend angezeigt, jedoch nicht das Bild selbst.

Bei unserem *Titel* haben wir momentan noch das Problem, dass er als einfacher Text über dem Beitrag angezeigt wird. Sinnvoller ist es, eine ordnungsgemäße HTML-Überschrift zu erstellen. Wählen Sie analog die *Typografie Überschrift* aus und klicken Sie auf das kleine Pluszeichen.

Dort wählen Sie als Überschrift *h1* oder *h2* – das hängt ganz davon ab, ob Sie die Menütitel anzeigen lassen oder ob Sie ein Template verwenden, das das Logo unnötigerweise mit einem *h1*-Element belegt. In meinem Beispiel habe ich das *h1*-Element gewählt. In der *Klasse* tragen Sie eine CSS-Klasse ein, die Sie später formatieren können, wenn Sie möchten oder es benötigen. Den *Anker* benötigen wir hier nicht.

### Überschriften für die Website-Struktur

Die *h*-Elemente, also die Überschriften in HTML-Syntax, sind wichtig für die Struktur der Website, damit auch für Google und für die Suchmaschinenoptimierung. *h1*-Überschriften werden von Google als wichtiger angesehen als *h2*-Überschriften, die wiederum wichtiger sind als *h3*-Überschriften und so weiter. Allerdings sollte die *h1*-Überschrift nur einmal vorkommen. (Wobei dies von den SEO-Spezialisten heiß diskutiert wird. Einige meinen, man könne die *h1*-Überschrift auch mehrmals abbilden.) Ein Logo (viele Templates formatieren das so) ist dafür Verschwendung.

Es ist wichtig, dass wir das Titelbild und die Überschrift für das Frontend anders positionieren als bisher. Beides muss in den Header, das Titelbild auf *#HEADER-A* und die Überschrift auf *#HEADER-B*. Das ist deshalb notwendig, da bei allen anderen Positionierungen entweder die Überschrift über dem Bild liegt oder sie nicht sauber nebeneinander angezeit werden.

Wenn Sie die Positionen für den *#HEADER* nicht haben, rufen Sie oben rechts bei den *Formular-Optionen* das *Template* auf. Im Bereich *Positionen* wählen Sie bei der *Kopfzeile* *Automatisch* oder *Tiefste* aus. Einmal auf *speichern* klicken, und Sie sind wieder in der *Formular-Ansicht* und haben nun auch die *#HEADER*-Positionen.

Klicken Sie nun auf den Punkt unter der Zahlenreihe, und Sie können die Größenangaben zu den einzelnen Positionen machen. Für das Bild neben der Überschrift geben wir *65 px* in der Breite vor. Die Höhe lassen wir leer. Für die Überschrift sollten Sie auch eine Breite vorgeben. Mir ist es ein paarmal passiert, dass die Überschrift über zwei Zeilen angezeigt wurde, obwohl eigentlich genügend Platz da gewesen wäre. Geben Sie hier eine Breite von *500 px* oder mehr an, passiert dieser Fehler nicht mehr.



Bild 3.33: Größenangaben festlegen.

Wenn Sie möchten, können Sie nun auch ein wenig per CSS eingreifen. Ich habe beim *beez3*-Template nachfolgende Zeilen an das Ende der *layout.css*<sup>11</sup> angefügt:

```
001 div.cck-line-header {
002 border-bottom:1px solid #dddddd;
003 height:5em;
004 margin-bottom:2em;
005 }
```

Damit habe ich den Unterstrich über die gesamte Zeilenbreite erzeugt und den Abstand zum nachfolgenden Text eingestellt.

Wenn Sie mit dem Standard-Template von Joomla! arbeiten, bedenken Sie bitte, dass bei einem Update eventuell Ihre Eingaben überschrieben werden. Sie können sich auch eine eigene CSS-Datei erstellen und diese entsprechend ins Template einbinden. Wenn Sie in CSS nicht so fit sind, empfehle ich Ihnen die Seiten von *de.selfhtml.org*<sup>12</sup>.

### 3.2.3 Mehr Bilder mit Feld X

Bislang können wir für das Seitenbild nur ein Bild auswählen. Schöner wäre es, wenn wir so viele Bilder für die Seite auswählen könnten, wie wir möchten. Genau das geht in SEBLOD.

Dazu legen wir uns ein neues Feld an, das wir sowohl in das *Admin Formular* als auch in das Formular für den *Inhalt* einbauen müssen. Eigentlich brauchen Sie dieses Feld nur gegen das bisherige Seitenbildfeld auszutauschen. Allerdings dürfen Sie das bisherige Bildfeld nicht löschen, denn wir brauchen es noch.

Legen Sie ein neues Feld an und wählen Sie unter *Typ* im Bereich *Sammlung* das *Feld X* aus. Ich habe dieses Feld in meinem Beispiel *Seitenbilder* genannt. Wichtig ist, dass Sie einen Bezug zu dem »Hochlade-Bild-Feld« legen. Das ist das Feld, das Sie eben aus dem Formular entfernt haben, um es durch dieses neue *Feld X*-Feld zu ersetzen.

Tragen Sie in *Feld (Formular)* den Namen des »Hochlade-Bild-Feldes« ein. Ich hatte es *Newsbild* genannt und trage eben diesen Namen hier ein. Das Ganze funktioniert so, dass eine Referenz auf dieses Feld gemacht wird, das beliebig oft im *Admin Formular* vervielfältigt werden kann. Im *Admin Formular* greift das *Feld X*, aber eigentlich ist es immer wieder das *Newsbild*-Feld, das angezeigt wird und unser ausgewähltes Bild ordnungsgemäß in die Datenbank überträgt.

In der linken Spalte geben Sie an, wie viele Bilder mindestens und maximal hochgeladen werden können, und in der rechten Spalte, dass Sie Bilder hinzufügen (+), löschen (-) und die Reihenfolge verschieben können. Im Backend erhalten Sie dann diese Ausgabe:

<sup>11</sup> Sie finden die *layout.css* im Template-Verzeichnis von *beez3*: *templates/beez3/css/layout.css*

<sup>12</sup> <http://wiki.selfhtml.org>

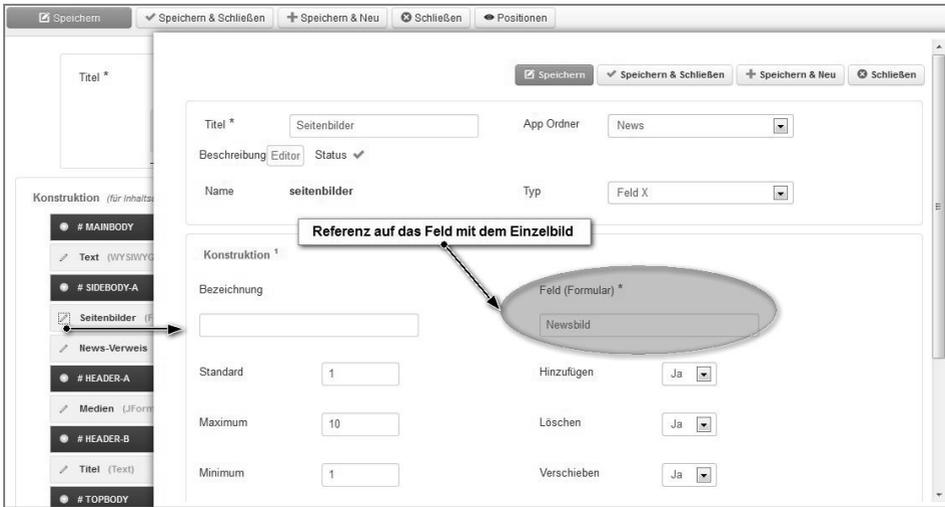


Bild 3.33: Feld X und seine Referenz.



Bild 3.34: So viele Bilder im Backend eingeben, wie Sie möchten.

Hier noch eine kleine Einstellung zum Abschluss dieses Bereichs: Sie müssen die *Typografie* (Nummer 2 im *Inhalt*-Formular anklicken) für *Feld X* auf *Bild* stellen, und zwar deswegen, weil SEBLOD sonst eine Aufzählung aus den Bildern macht. Die Bilder werden zwar auch angezeigt, aber mit einem Aufzählungszeichen. So aber werden die Bilder ohne Aufzählung untereinander angezeigt.

### 3.2.4 Abhängige Bilder

Wer mit regelmäßigen News arbeitet, möchte diese News vielleicht mit festen Bildern, die einer Kategorie zugeordnet sind, anzeigen. Angenommen, wir haben regelmäßig News zu Kultur, Politik und Wirtschaft und alle nicht eindeutig zuzuordnenden News werden allgemein unter der Kategorie *News* gespeichert. Toll wäre es, wenn wir zu jeder Kategorie ein festes Bild haben, das automatisch nach Auswahl der Kategorie neben der Überschrift angezeigt wird.

In SEBLOD ist das kein Problem und ohne Programmierung möglich.

Wechseln Sie in das Formular *Admin Formular* und klicken Sie auf *Nummer 5*. Wir wollen das Medienfeld abhängig von den Kategorien machen. Also klicken wir im *Medienfeld* auf das kleine Wort *Hinzufügen*, das in den Feldern aufgetaucht ist (wenn Sie bereits etwas eingetragen haben, erscheint *<Bearbeiten />*).

Unter *Status* tragen wir unseren Wert ein und unter *Auslöser* die Bedingung.

Fangen wir mit dem Auslöser an: Zuerst wählen wir das Feld aus, das für die Bedingung zuständig ist, hier das Feld *Article Category Id*. Ist nun in diesem Feld die Kategorie *News* (*Gleich 10*, 10 steht für die *Id* der Kategorie) ausgewählt, soll dieses Feld ausgefüllt werden mit dem Wert *images/news.png*. Sollten Sie die *Category Id* nicht kennen, so können Sie diese über den kleinen orangefarbenen Pfeil auswählen. Das Wort *#form#* müssen Sie im vorgegebenen Feld stehen lassen und den Haken bei der *Checkbox* entfernen.

Damit haben wir die erste Bedingung eingetragen. Das müssen wir nun pro Kategorie (Kultur, Politik, Wirtschaft – falls noch nicht geschehen, müssen Sie diese Kategorien natürlich noch in Joomla! anlegen) wiederholen und jeweils ein entsprechendes Bild zuordnen. Um mehrere Bedingungen einzutragen, klicken Sie oben auf das kleine grüne Pluszeichen.

**Bild 3.35:** Abhängige Zustände.

Die Bedingungen im Überblick:

- Dieses Feld – ist ausgefüllt – images/news.png – #form# – WENN – Article Category Id – Gleich – 10
- Dieses Feld – ist ausgefüllt – images/kultur.png – #form# – WENN – Article Category Id – Gleich – 11
- Dieses Feld – ist ausgefüllt – images/politik.png – #form# – WENN – Article Category Id – Gleich – 12
- Dieses Feld – ist ausgefüllt – images/wirtschaft.png – #form# – WENN – Article Category Id – Gleich – 13

Im *images*-Verzeichnis müssen natürlich die genannten Bilder liegen. Sie können dazu gern die Iconsammlung benutzen, die Sie sich an der angegebenen Stelle downloaden können.

Wenn Sie nun eine neue News anlegen, wird im *Medien*-Feld automatisch das zugehörige Bild ausgewählt, wenn Sie die Kategorie wechseln. Sehen können Sie das, indem Sie mit der Maus über das kleine Auge beim *Medien*-Feld fahren. Aber Sie können auch weiterhin ein individuelles Bild zuordnen, indem Sie im *Medien*-Feld einfach ein neues Bild laden.

# 4

## Die Speicherung und ihre Hürden

Plötzlich gibt es eine weiße Seite, oder Daten werden nicht gespeichert, oder es werden merkwürdige Wörter ausgegeben – und meist ist es ein Fehler im Speicherbereich von SEBLOD.

*Benutzerdefiniert, Standard, Json* – in einer vorhandenen Tabelle oder einer neuen oder ganz anders – SEBLOD gibt da einige Möglichkeiten vor, und jede hat ihre eigene Auswirkung.

Der Speicherbereich ist sehr wichtig und bisweilen gleichermaßen mysteriös. Fragt man jemanden, der sich mit SEBLOD ein wenig auskennt, bekommt man fast immer die Antwort: »Der Speicherbereich ist ein Kapitel für sich.« Wie aber dieses »Kapitel« genau funktioniert, weiß kaum jemand. Leider ist auch die Dokumentation gerade in diesem so wichtigen Bereich auf [www.seblod.com](http://www.seblod.com) sehr dürftig.

Bringen wir also Licht ins Dunkel.

### 4.1 Ein Blick hinter die Kulissen: MySQL

In den meisten Fällen arbeitet Joomla! mit einer MySQL-Datenbank. Aber auch, wenn es eine andere Datenbank ist, beispielsweise ein Microsoft-Server, handelt es sich stets um eine SQL-Datenbank. SQL<sup>13</sup> ist nichts weiter als eine Datenbanksprache.

---

<sup>13</sup> SQL = Structured Query Language

Die Sprache ist recht einfach gehalten. SQL macht Datenbankabfragen und erhält so einzelne Datensätze, die nach verschiedenen Kriterien ermittelt und zusammengefasst werden. Diese Ergebnisse wiederum werden in der Joomla!-Programmierung dafür benutzt, Daten anzuzeigen bzw. sie erst einmal entsprechend zu speichern.

### Die Datenbank

Die Datenbank ist ein System zur Datenspeicherung. Die Daten werden dafür in verschiedenen Tabellen abgelegt, die teilweise miteinander verknüpft sind. Wenn Informationen gespeichert werden, entstehen Daten. Wenn Sie beispielsweise einen Text in den Editor eingeben, ist er wieder weg, wenn Sie vorher nicht speichern. Erst durch die Speicherung wird Ihr Text zu Daten. Das klingt logisch. In einer Datenbank stehen also gespeicherte Informationen. Und Informationen werden zuvor eingegeben, damit sie gespeichert werden können.

SEBLOD legt von vornherein bereits einige Datenbanktabellen an, die zur Verwaltung der Erweiterung benötigt werden.

Eine Tabelle besteht aus Spalten. Diese Spalten wiederum haben Namen, und das sind exakt die Namen, die Sie angeben müssen (neben der Datenbanktabelle, in der diese Spalte steht). Damit Joomla! und SEBLOD das hinterher auch alles einwandfrei zuordnen können, gibt es die IDs, denen Sie an verschiedenen Stellen immer wieder begegnen. Eine ganz bestimmte ID kennzeichnet einen einzelnen Datensatz. Wenn Sie in diesem Datensatz eine bestimmte Spalte namentlich ansprechen, haben Sie den Inhalt eines einzelnen Feldes. Genannt wird das »Datenzelle«.

id	asset_id FK to the #_assets table.	title	alias	introtext	fulltext
13	95	Kultur vom Feinsten	kultur-vom-feinsten	...cck:19:/:cck: :seitenbilder:2: consetctuer a /seitenb...	
14	97	Kulturnews	kulturnews	...cck:20:/:cck: :newsverweis: consetctuer a /newsverwe...	
15	99	Neues aus Berlin	neues-aus-berlin	...cck:21:/:cck: :seitenbilder:2: consetctuer a /seitenb...	
16	102	Eine neue News	eine-neue-news	...cck:22:/:cck:chr	

Bild 4.1: Ein kleiner Auszug aus der Datenbanktabelle #\_content.

Wenn Sie nun Informationen über Ihre App und Ihre Felder eingeben, speichert SEBLOD diese in eigens von SEBLOD benutzten und zuvor erstellten Datenbanktabellen. Wenn Sie selber Inhalte (Informationen) in Ihre Felder, beispielsweise News, eingeben, müssen diese Informationen natürlich auch gespeichert werden. Dafür nehmen Sie entweder Tabellen, die von Joomla! zuvor erstellt wurden, oder welche, die SEBLOD angelegt hat. Wenn Sie sich mit Datenbanken und SQL etwas auskennen, können Sie auch eigene Tabellen erstellen, auf die Sie mit SEBLOD zugreifen können.

Gerade diese letzte Gegebenheit bietet eine Menge Möglichkeiten. Vielleicht haben Sie eine eigene Datenbanktabelle erstellt oder eine aus einem anderen Programm übernommen. In Joomla! können Sie in der Regel auf diese Daten nur mittels Programmierung zugreifen – in SEBLOD reicht die einfache Auswahl der Tabelle. SEBLOD bietet auch die Möglichkeit, CSV-Daten einzubinden. Damit sind weitere großartige Möglichkeiten geschaffen worden.

Microsoft-Excel-Tabellen lassen sich beispielsweise im CSV-Format abspeichern und mithilfe von SEBLOD in die Datenbank laden. Dort wird eine neue Tabelle angelegt, auf die Sie wiederum mit von Ihnen angelegten SEBLOD-Feldern zugreifen können.

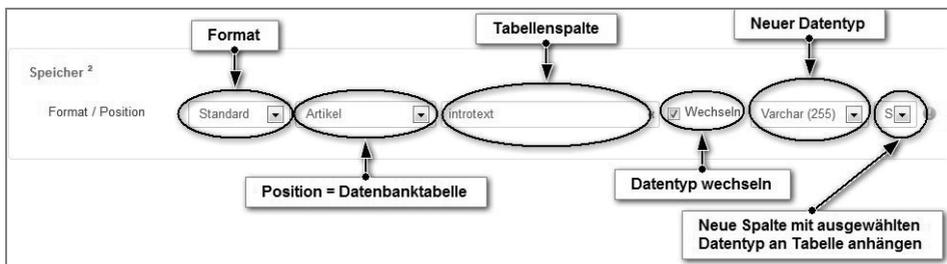


Bild 4.2: Der Speicherbereich eines Feldes.

Um den Inhalt eines Feldes abzuspeichern, sind verschiedene Angaben notwendig. Zuerst das Format, in dem gespeichert werden soll. Was es damit auf sich hat, erfahren Sie in den nächsten Kapiteln. Danach wählen Sie die Tabelle aus, in der die Daten abgelegt werden sollen, und dazu das entsprechende Feld – den Spaltennamen. Je nachdem, was Sie zuvor ausgewählt haben, erzeugen Sie eigene Spalten und hängen sie an bestehende Tabellen an – erweitern also bestehende Tabellen oder erzeugen lediglich in einer ganz bestimmten SEBLOD-Tabelle eine neue Spalte. Auf den nachfolgenden Seiten wird das alles ein wenig entknotet.

## 4.2 Speicherformat: Standard

Als Beispiel erstelle ich ein neues Formular zu unserer News-App mit zunächst einem einzigen Feld, das ich im *Admin Formular* anzeigen lasse.

Titel *	<input type="text" value="Ein Feld zum testen"/>	App Ordner	<input type="text" value="News"/>
Name	<b>ein_feld_zum_testen</b>	Typ	<input type="text" value="Text"/>
<b>Konstruktion <sup>1</sup></b>			
Bezeichnung	<input type="text" value="Ein Feld zum testen:"/>	Standard Wert	<input type="text"/>
Mindestlänge	<input type="text" value="0"/>	Maximallänge	<input type="text" value="255"/>
Größe	<input type="text" value="32"/>		
<b>Speicher <sup>2</sup></b>			
Format / Position	<input type="text" value="Standard"/>	<input type="text" value="Artikel"/>	<input type="text" value="fulltext"/> « <input type="button" value="Wechseln"/>

Bild 4.3: Das Feld ...

Titel *	<input type="text" value="Datenbankspeicherung"/>	App Ordner	<input type="text" value="News"/>
<div style="display: flex; justify-content: space-between;"> <div style="display: flex; gap: 5px;"> <div style="background-color: #333; color: white; padding: 2px 5px;">Admin Formular</div> <div style="padding: 2px 5px;">Seitenformular</div> <div style="padding: 2px 5px;">Intro</div> <div style="padding: 2px 5px;">Inhalt</div> </div> <div style="display: flex; gap: 5px;"> <div style="padding: 2px 5px;">Konfiguration</div> <div style="background-color: #333; color: white; padding: 2px 5px;">Felder</div> <div style="padding: 2px 5px;">Tem</div> </div> </div>			
Ansichten <span style="margin-left: 100px;">Optionen</span>			
<b>Konstruktion (für Admin Formularansicht)</b>			
	Bezeichnung	Variante	
<input type="radio"/>	<input type="text" value="# MAINBODY"/>	<input type="text" value="- Standard -"/>	<input type="button" value="↻"/>
<input type="radio"/>	<input type="text" value="Ein Feld zum testen (Text)"/>	<input type="text" value="Ein Feld zum testen"/>	<input type="text" value="Standard"/>
<input type="radio"/>	<input type="text" value="# TOPBODY"/>	<input type="text" value="Standard"/>	<input type="button" value="↻"/>

Bild 4.4: ... eingebaut im Admin Formular.

Erzeuge ich nun einen neuen Beitrag in unserem Datenbanktest, der nur das eine Feld hat, und schreibe dort »Hochwichtiger Text« hinein, wird dieser Text genau so in der Datenbank abgespeichert.



**Bild 4.5:** Im Bearbeitungsmodus des Datensatzes ist unser »Hochwichtiger Text« einwandfrei zu erkennen.

Da ich zuvor angegeben habe, dass die *Artikel-Tabelle* benutzt werden soll und dort die Spalte *fulltext*, wird unsere Information genau an dieser Stelle gespeichert.

Für das Format *Standard* gelten vier Bedingungen:

Es muss eine der vorgegebenen Tabellen benutzt werden.

Es handelt sich nicht um ein Sammelfeld/eine Spalte (siehe dazu das nächste Kapitel »Speicherformat: Benutzerdefiniert«).

Es ist eine Tabellenspalte in den vorgegebenen Tabellen vorhanden.

Oder es wird ein Wechsel des Datentyps vorgenommen und eine freie Tabelle ausgewählt (Näheres weiter unten).

Vorgegebene Tabellen sind solche, die Sie bei den *Positionen* im Pull-down-Button unter dem Bereich *Joomla!* auswählen können.



**Bild 4.6:** Eine der vorgegebenen Tabellen.

Es gibt eine weitere vorgegebene Tabelle, die Sie erreichen, wenn Sie den Checkbutton *Wechseln* aktivieren. Dazu mehr etwas weiter unten.

## 4.3 Speicherformat: Benutzerdefiniert

Wenn Sie unter *Benutzerdefiniert* abspeichern, ändert sich das Speicherformat dahin gehend, dass Ihre Informationen vom Aliasnamen Ihres Feldes eingefasst werden:

```
001 <br />:ein_feld_zum_testen::Hochwichtiger Text::/ein_feld_zum_testen::
```

Es wird zunächst ein HTML-Zeilenumbruch eingefügt. Danach wird die eingegebene Information mit dem Aliasnamen des Feldes eingefasst. Wie bei HTML-Elementen üblich, wird der Aliasname am Ende mit einem führenden Slash begonnen, dieser

signalisiert den Abschluss. Statt der spitzen Klammern, wie Sie es aus HTML kennen, werden in dieser Syntax doppelte Doppelpunkte verwendet.

Der Vorteil ist, dass Sie auf diese Art mehrere Inhalte in eine Tabellenspalte ablegen können und SEBLOD tatsächlich nur die Daten, die zum entsprechenden Aliasfeld gehören, anzeigt.

Damit wir das überprüfen können, legen wir uns kurzerhand ein zweites Textfeld an und speichern es auch *Benutzerdefiniert* in der *Artikel-Tabelle* unter *fulltext*:



Bild 4.7: Beide Felder werden gleich abgespeichert ...



Bild 4.8: ... und in das Admin Formular eingebaut.

Rufen Sie nun das Formular bei den Beiträgen auf und geben entsprechenden Text ein, wird in die Datenbank beides mit dem jeweiligen Alias eingetragen.



Bild 4.9: Beitrag eingeben.

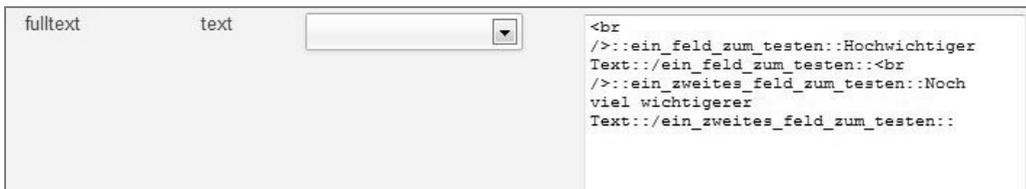


Bild 4.10: So wird es dann in der Datenbank gespeichert.

Der große Vorteil dieser Art der Speicherung ist, dass Sie mehrere unterschiedliche Datenfelder in einer Tabellenspalte speichern und es SEBLOD überlassen können, alles anschließend wieder der richtigen Stelle zuzuordnen. Diese Art der Speicherung ist besonders auch dann sinnvoll, wenn Sie Felder kreieren, zu denen es keine Entsprechung in der *Artikel-Tabelle* (`#__content`) gibt. Ganz so, wie wir es in unserer ersten News hatten, als wir das Feld für die Links angelegt haben. Dort haben wir diese Technik benutzt und den Inhalt des Feldes *Benutzerdefiniert* in dem Feld *introtext* gespeichert. SEBLOD benutzt für diese »Sammelspeicherungen« sehr gern die Tabellenspalte *introtext*.

Wenn Sie ein Feld auf den *fulltext*, in unserem Beispiel auf *Standard* setzen würden, würde Ihnen alles, was dort steht, angezeigt, also der gesamte Text mit den SEBLOD-Formatierungen, wie er im Bild 4.10 abgebildet ist. Denn der *Standard* unterscheidet nicht die einzelnen Aliasnamen, sondern zeigt es so an, wie die Daten darin stehen.

## 4.4 Speicherformat: Json

Das Speicherformat *Json*<sup>14</sup> ist fast identisch mit dem Speicherformat *Benutzerdefiniert*. Der einzige Unterschied ist das etwas andere Format:

```
001 {"ein_feld_zum_testen":"Hochwichtiger Text","ein_zweites_feld_zum_
testen":"Noch viel wichtigerer Text"}
```

Es gibt keine doppelten Doppelpunkte und keinen Anfang und kein Ende. Es ist mehr eine Auflistung, mit Kommata getrennt, bei der die einzelnen Inhalte in Anführungsstriche eingefasst sind. Der ganze Block ist von geschweiften Klammern umgeben.

Rein inhaltlich ändert sich nichts im Vergleich zur benutzerdefinierten Speicherung. Auch hier gilt all das, was ich im Kapitel vorher geschrieben habe. Der Unterschied besteht darin, dass Sie mittels Programmierung leichter auf die einzelnen Felder zugreifen können.

JSON lässt sich sowohl mit der Programmiersprache JavaScript als auch mit PHP ansprechen. Sie haben bei diesem Format also entsprechende Möglichkeiten in der Weiterverarbeitung Ihrer Daten.

Wenn Sie jedoch nicht programmieren, hat dieses Speicherformat vermutlich nicht allzu viel Sinn. Denkbar wäre allenfalls, dass Sie ein Plug-in oder eine andere Erweiterung benutzen, die dieses Speicherformat voraussetzt. Allerdings sind mir derzeit keine solchen Erweiterungen im Zusammenhang mit SEBLOD bekannt.

---

<sup>14</sup> JSON steht für JavaScript Object Notation.

## 4.5 Speicherformat: Kein

Sogar das gibt es: dass in keiner Datenbanktabelle etwas gespeichert wird. Zum Beispiel ist das so, wenn Sie Buttons benutzen, die ein Formular abschicken sollen oder eine E-Mail. Da braucht nichts gespeichert zu werden, weil keine Informationen entstehen. Dort wird einfach eine Aktion, eine Funktion, eingeleitet.

## 4.6 Speicherposition

Die *Speicherposition* bezieht sich auf die Datenbanktabelle, in der die Informationen gespeichert werden sollen. Vorgegeben sind unter dem Bereich *Joomla!* die Standardbereiche von SEBLOD: *Artikel*, *Benutzer*, *Benutzergruppe*, *Kategorie*.

Jedem dieser Bereiche ist eine bestimmte Datenbanktabelle zugeordnet:

- Artikel = `#__content`
- Kategorie = `#__categories`
- Benutzer = `#__users`
- Benutzergruppe = `#__usergroups`

Wählen Sie beispielsweise die *Artikel* unter *Position* aus, werden Ihre Daten automatisch in der Datenbanktabelle `#__content` gespeichert. Das `#_` steht auch hier für ein beliebiges Präfix. Siehe dazu auch Seite 34 (Kasten: Platzhalter für das Datenbankpräfix)

Wenn Sie eine andere Tabelle benutzen möchten, wählen Sie *Frei* aus. Danach erhalten Sie ein zusätzliches Drop-down-Feld, in dem Sie die Tabelle auswählen können. Es werden Ihnen alle Tabellen angezeigt, die in der Datenbank angelegt wurden.

Sie sehen, dass das nicht gerade wenige sind. Wenn Sie keine Tabelle aus dem Joomla!-Bereich nehmen möchten, gibt Ihnen SEBLOD Lösungen für Tabellen.

- `#__cck_store_item_`
- `#__cck_store_form_`
- `#__cck_store_item_users`

Am Ende dieser Vorgabe fehlt allerdings noch eine Bezeichnung – bis auf die Tabelle für die *users*, die als einzige vorgegeben ist. Wenn Sie die News angelegt haben, könnte es sein, dass Sie nun in der Datenbank folgende Tabelle stehen haben:

- `#__cck_store_form_news`

Das hängt davon ab, wie Sie die einzelnen Felder abgespeichert haben. Wenn Sie dort Feldnamen nehmen, die es in der vorgegebenen *Artikel-Tabelle* nicht gibt, legt SEBLOD

eine neue Tabelle mit dem Namen `#__cck_store_item_IHR-FORMULARNAME` an und erzeugt dort eine Spalte mit den »unbekannten« Namen.

Als Beispiel habe ich ein kleines Formular mit dem Namen *Datenbankspeicherung* erstellt. Es gibt in diesem Formular nur ein Textfeld, das ich der *Artikel*-Tabelle mit dem Spaltennamen *unbekannte\_spalte* zugeordnet habe.



**Bild 4.11:** Zuordnung zu einer nicht vorhandenen Spalte.

In der Datenbank wird nun die Tabelle `#__cck_store_item_datenbankspeicherung` angelegt, und dort finden wir auch unsere Spalte wieder und natürlich auch unseren Inhalt.



**Bild 4.12:** Unsere Daten in der Datenbank.

Am Rande sei angemerkt, dass es natürlich egal ist, welches Format Sie dafür auswählen. Sowohl *Standard* als auch *Benutzerdefiniert* sind dafür benutzbar. Das Format hat Einfluss auf das »Wie« - und nicht auf das »Wo«.

Der Vorteil ist, dass Sie damit die Joomla!-eigenen Datenbanktabellen in Ruhe lassen können und auf eigene Tabellen ausweichen, die SEBLOD problemlos verarbeiten kann. Nicht umsonst heißt es immer wieder, dass die Core-Dateien nicht angefasst werden sollen. Und was für die Dateien gilt, gilt bei der Datenbank natürlich erst recht. Es kann durchaus vorkommen, dass bei einem Update die Tabellenstruktur verändert wird, oder Sie müssen später eine Migration durchführen, wo sich vielleicht

die Tabellenstruktur verändert hat. Dann können Sie aber problemlos die SEBLOD-Tabellen mitnehmen.

## 4.7 Datentyp wechseln

Standardmäßig werden neue Daten so abgespeichert, wie sie vom Feldtyp vorgegeben sind. Ein *Link*-Feld speichert eine URL, ein *eMail*-Feld eine E-Mail, ein *Bild*-Feld einen *image*-Pfad und ein *Text*-Feld einen *Varchar* (255). So weit zur Vorgabe. Das heißt aber nicht, dass Sie das so übernehmen müssen. Insbesondere bei Feldern, die bestimmte Funktionen haben, die vom Standard abweichen, kann es sinnvoll sein, den Datentyp den eigenen Bedürfnissen anzupassen.

So könnte es vielleicht sein, dass Sie sich für ein Textfeld entschieden haben, in das Sie aber ein Datum eingeben. Dann könnte es sinnvoll sein, es auch gleich als solches abzuspeichern, indem Sie *Datetime* auswählen. Oder Sie geben über das Textfeld Zahlen ein und können dann es auch als Zahlenfeld abspeichern. Dafür stehen Ihnen einige Datentypen zur Verfügung.

Jeder dieser Datentypen stellt in der Datenbank einen bestimmten Speicherplatz zur Verfügung. Die Programmierer unter Ihnen wissen, dass ein *Varchar* (2048) erheblich mehr Speicherplatz benötigt als ein *Varchar* (7). Das Problem ist, dass dieser Speicherplatz immer pro Datensatz zur Verfügung gestellt wird – auch wenn gar nichts drinsteht. Wer also auf Nummer sicher gehen will und überall das höchste auswählt, erzeugt eine große Datenbank mit hoher Speicheranforderung, die erst einmal geladen werden muss, wenn entsprechende Anfragen kommen.

Darum rate ich, zu übernehmen, was SEBLOD anbietet. Wenn Sie genau wissen möchten, was dort passiert, empfehle ich einen Blick in die wirklich sehr gute MySQL-Dokumentation, die in deutscher Sprache vorliegt.<sup>15</sup>

Wenn Sie den Datentyp wechseln, haben Sie auch die Möglichkeit, auszuwählen, ob die Spalte an die ausgewählte Datentabelle angehängt werden soll. Das funktioniert bei allen Tabellen, auch bei den Joomla!-Core-Tabellen. Aus oben genannten Gründen sollten Sie das jedoch nicht tun. Wenn Sie aber eigene Tabellen erzeugen oder benutzen, ist es vielleicht eine Option. Dann haben Sie Ihre Daten immer zusammen in einer Tabelle, die Sie nach Belieben erweitern können, allerdings nicht verkleinern, das funktioniert nur in der Datenbank.

## 4.8 Eigene Tabelle erstellen

Eine neue und eigene Datentabelle in der Datenbank zu erstellen ist sehr einfach. Dennoch machen Sie bitte vorher eine Sicherung der Datenbank. Man kann schließlich nie wissen.

---

<sup>15</sup> <http://dev.mysql.com/doc/refman/5.1/de/data-types.html>

In die Datenbank gelangen Sie lokal über *phpMyAdmin* oder beim Hoster über die Domainverwaltung, die bei den einzelnen Anbietern abweichen kann. Lokal ist es bei mir der Aufruf <http://localhost/phpmyadmin>.

Nachdem Sie die passende Datenbank ausgewählt haben, finden Sie ganz unten in der linken Tabellenaufistung einen Button mit der Aufschrift *Erzeuge Tabelle*. Dort einfach mutig draufgeklickt!

Oben geben Sie einen Tabellennamen ein. Es ist sehr wichtig, dass Sie das Präfix mit angeben! Wenn Sie das nicht machen, wird Ihnen später der Versuch, die Tabellenspalten aufzurufen, die Seite zum Einfrieren bringen. Zumindest ist mir das passiert.

<b>Tabellenname:</b>	
<input type="text" value="bdayo_meine_tabelle"/>	
<b>Struktur</b>	
<b>Spalte</b>	<input type="text" value="id"/> <input type="text" value="irgendeine_spalte"/>
<b>Typ</b>	<input type="text" value="INT"/> <input type="text" value="TEXT"/>
<b>Länge/Set</b>	<input type="text" value="10"/> <input type="text"/>
<b>Standard</b>	<input type="text" value="Kein(e)"/> <input type="text" value="Kein(e)"/>
<b>Kollation</b>	<input type="text" value="utf8_general_ci"/> <input type="text" value="utf8_general_ci"/>
<b>Attribute</b>	<input type="text"/> <input type="text"/>
<b>Null</b>	<input type="checkbox"/> <input type="checkbox"/>
<b>Index</b>	<input type="text" value="UNIQUE"/> <input type="text" value="---"/>
<b>AUTO_INCREMENT</b>	<input checked="" type="checkbox"/> <input type="checkbox"/>
<b>Kommentare</b>	<input type="text"/> <input type="text"/>
<b>MIME-Typ</b>	<input type="text"/> <input type="text"/>
<b>Darstellungsumwandlung</b>	<input type="text"/> <input type="text"/>
<b>Umwandlungsoptionen</b>	<input type="text"/> <input type="text"/>
<b>Tabellen-Kommentar:</b>	<input type="text"/>
<b>Tabellenformat:</b>	<input type="text" value="InnoDB"/>
<b>Kollation:</b>	<input type="text"/>
<b>PARTITION Definition:</b>	<input type="text"/>

Bild 4.13: Eine neue Tabelle in der Datenbank erzeugen.

In der Struktur haben Sie Platz, zwei Spalten vorzugeben, wobei die erste vorzugsweise für die ID reserviert werden sollte. Die ID ist wichtig, damit jeder Datensatz einwandfrei erkannt werden kann. Eine ID wird denn auch einmalig vergeben. Dafür tragen Sie in *Spalte* am besten auch den Namen *id* ein. Als *Typ* eignet sich *INT*. *INT* steht für Integer und hat Platz für eine recht große Zahl (Details finden Sie in der MySQL-Dokumentation).

*Standard* lassen Sie auf *Kein(e)* stehen. In der *Kollation* wählen Sie die Zeichencodierung aus. In unserem Fall ist das *utf8\_general\_ci*. Sie finden sie im Auswahlfeld ziemlich weit unten im vorletzten Block.

Im *Index* tragen Sie noch *UNIQUE* ein. Und aktivieren Sie das Kästchen für den *AUTO\_INCREMENT*. Damit haben Sie eine Spalte definiert, die einen Wert nur einmal enthalten kann (*UNIQUE*) und bei der automatisch bei jedem neuen Datensatz hochgezählt wird.

In der rechten Spalte definieren Sie zur Übung erst einmal ein einfaches Textfeld: Unter *Spalte* steht bei mir *irgendeine\_Spalte*, was Sie aber gern durch einen fantasiereicheren Namen ersetzen können.

Dabei beachten Sie bitte: keine Sonderzeichen, keine Umlaute und keine Leerzeichen. Statt Leerzeichen benutzen Sie die Unterstriche. Als *Typ* wählen Sie den *TEXT* aus, *Standard* bleibt auf *Kein(e)*, in die *Kollation* kommt wieder unser *utf8\_general\_ci*.

Wenn Sie möchten, geben Sie einen Kommentar zu Ihrer Tabelle ein. Das *Tabellenformat* lassen Sie bitte auf *InnoDB16* stehen.

Das war's auch schon. Jetzt noch ganz unten links auf *Speichern* drücken, und schon haben Sie eine neue Datenbanktabelle angelegt.

#### Von der Datenbankübersicht zur Eingabemaske

Sie kommen zu dieser Eingabemaske auch, wenn Sie sich in der Datenbankübersicht befinden – also dann, wenn in der rechten Hälfte auch alle Datenbanktabellen aufgelistet sind –, dann gibt es ebenfalls ganz unten *Neue Tabelle in Datenbank [DATENBANKNAME] erstellen*. Dort geben Sie den Namen ein, wie eben beschrieben, die Anzahl der Spalten – im obigen Beispiel waren es zwei – und klicken dann auf *OK*.

Ihre neu erstellte Tabelle können Sie wie im Kapitel 4.6 beschrieben ansprechen. Erstellen Sie ein neues Feld, wählen Sie ein *Format* aus und als *Position* *Frei*. Dann können Sie Ihre Tabelle auswählen und eine Ihrer angelegten Spalten oder mit *Wechseln* eine neue Spalte an Ihre bestehende Tabelle anfügen.

<sup>16</sup> <http://dev.mysql.com/doc/refman/5.1/de/innodb.html>



**Bild 4.14:** Die neue Tabelle in SEBLOD benutzen.

## 4.9 CSV-Import

In SEBLOD 2 gab es lange Zeit eine Betaversion eines CSV-Importers. In SEBLOD 3 ist er verschwunden, er findet sich aber auf der Extensionsseite direkt auf der SEBLOD-Seite wieder. Dort kommen Sie hin, wenn Sie auf der SEBLOD-Startseite *Download* anklicken und dann im Menü *Products* auswählen. Dort finden Sie den *SEBLOD Importer*<sup>17</sup>.

Sie müssen sich registrieren bzw. anmelden zum Download. Nachdem Sie das Add-on wie üblich über den Installer in Joomla! installiert haben, finden Sie es direkt im Komponentenmenü als Untermenüeintrag bei *SEBLOD3.x*.

Zuvor habe ich mir eine kleine Excel-Tabelle angelegt mit drei Büchern und Informationen zu Titel, Autor und Verlag. Diese Excel-Tabelle habe ich als CSV-Datei abgespeichert.

	A	B	C	
1	<b>Buchtitel</b>	<b>Autor</b>	<b>Verlag</b>	
2	Die Elfen	Bernhard Hennen	Heyne	
3	Stiller	Max Frisch	Suhrkamp	
4	Gevatter Tod	Terry Pratchett	Piper	
5				

**Bild 4.15:** Eine Excel-Tabelle, die im CSV-Format abgespeichert wird.

Ich möchte diese Excel-Tabelle als Beitrag anzeigen lassen. Also wähle ich im CSV-Importer das *Inhaltsobjekt Artikel Standard* aus und ebenso im *Inhaltstyp*. Im *Inhaltstyp* könnten Sie beispielsweise auch die News auswählen.

<sup>17</sup> <http://www.SELOD™.com/v3/products/7>

**SEBLOD Importer → Add-on for SEBLOD 3.x**  
 Would you like to import Content from CSV Files or Third-party Components ?  
 Here is **SEBLOD Importer** ~ Enjoy !

---

**Import :: from CSV File**

Inhaltsobjekt \*

Inhaltstyp \*

File to Import \*

Trenner

Force UTF-8

**Standardwerte**

Autor

Kategorie

Status

WIE STELLT MAN DIESE ERWEITERUNG EIN?

---

**Import :: from Third-party Component**

**Bild 4.16:** Die CSV-Datei importieren.

Mit *Import from File* wähle ich meine CSV-Datei aus.

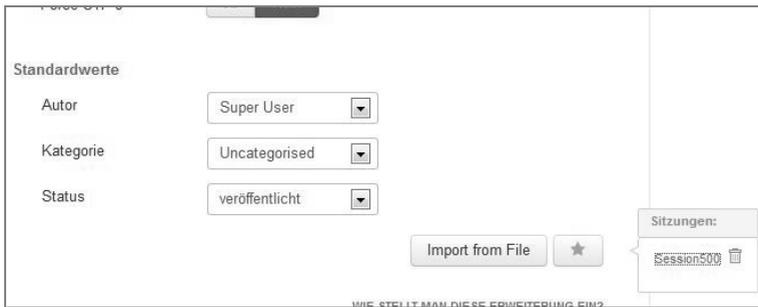
#### Trenner für die CSV-Datei

Wichtig ist der Trenner. MS-Excel nimmt standardmäßig das Semikolon als Trenner zwischen den einzelnen Daten. Wenn Sie sich unsicher sind, können Sie eine CSV-Datei auch mit einem beliebigen Texteditor aufrufen und einfach nachsehen. Ich stelle den Trenner auf das Semikolon um.

Unsere Datenbank ist ohnehin auf *utf-8\_general\_ci* eingestellt. Aber Sie können hier auch noch einmal darauf bestehen, dass es wirklich so gemacht wird. Interessant ist das eigentlich erst, wenn Ihre Zeichencodierung generell abweicht.

Bei *Autor*, *Kategorie* und *Status* können Sie Joomla!-spezifische Dinge einstellen – also kundtun, wer der *Autor* der Beiträge sein soll, in welcher *Kategorie* die Beiträge erscheinen und welchen *Status* sie haben sollen.

Wenn Sie Ihre Einstellungen gemacht haben und sie öfter benutzen möchten, können Sie sie über den orangefarbenen Stern unten rechts speichern. Sie können dann beim nächsten Aufruf des Importers durch Überfahren des Sterns die Voreinstellung erneut abrufen oder eine andere durch Anklicken speichern. Allerdings wird die ausgewählte Datei nicht mitgespeichert. Diese müssen Sie mittels *Durchsuchen* jedes Mal erneut auswählen.



**Bild 4.17:** Einstellungen merken.

Wenn Sie nun auf den Button *Import from File* klicken, passiert Folgendes:

SEBLOD legt zunächst eine neue Datenbanktabelle (`#__cck_store_form_article`) an, sofern sie noch nicht existiert, und hängt dort die drei neuen Felder nebst Inhalt an:

	id	Buchtitel	Autor	Verlag
<input type="checkbox"/> <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	1	Die Elfen	Bernhard Hennen	Heyne
<input type="checkbox"/> <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	2	Stiller	Max Frisch	Suhrkamp
<input type="checkbox"/> <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	3	Gevatter Tod	Terry Pratchett	Piper

**Bild 4.18:** Drei neue Datensätze in der `#__cck_store_form_article`.

Gleichzeitig werden drei neue Beiträge erzeugt (weil wir den Importer angewiesen haben, die CSV-Daten in den Inhaltstyp *Artikel* zu schreiben), die per ID mit der neuen Tabelle verknüpft sind. SEBLOD zeigt die drei neuen Felder jeweils in den Beiträgen an. Die drei neuen Beiträge haben als Titel das aktuelle Datum und die Uhrzeit erhalten.

	Status	Titel	Zugriffsebene	Autor	Sprache	Datum	ID
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	2013-05-09-13-52-59 Kategorie: Uncategorised	Öffentlich	Super User	Alle	09.05.2013	1
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	2013-05-09-13-52-59 Kategorie: Uncategorised	Öffentlich	Super User	Alle	09.05.2013	2
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	2013-05-09-13-52-59 Kategorie: Uncategorised	Öffentlich	Super User	Alle	09.05.2013	3

**Bild 4.19:** Drei neue Beiträge.

Die drei neuen Felder erscheinen am Ende des Beitrags:

The screenshot shows a text editor window titled "Introtext". At the top is a rich text toolbar with various icons for bold, italic, underline, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, insert image, and print. Below the toolbar is a large empty text area. Underneath the text area is a path field labeled "Pfad: p". Below the path field are two buttons: "Beiträge" and "Bild". Below these buttons are four form fields: "Fulltext" with an "Editor" button, "Buchtitel" with the text "Die Elfen", "Autor" with the text "Bernhard Hennen", and "Verlag" with the text "Heyne".

**Bild 4.20:** Excel-Inhalte in einem Beitrag.

Was Sie nun noch machen müssen, ist, ein Formular zu erstellen, das die Beiträge so ausgibt, wie Sie sich das vorstellen, damit so Ihre CSV-Daten angezeigt werden. Sie können natürlich auch vor dem Import einen passenden Inhaltstyp erstellen, den Sie dann dem Importer zuweisen. Was Sie dabei beachten müssen, ist, dass die Felder auch so benannt sind wie die Excel-Spalten. In meinem Beispiel hätte ich also einen Inhaltstyp mit den drei Textfeldern *Autor*, *Verlag* und *Buchtitel* anlegen müssen.

# **Teil 2 - Die Schaltzentrale**

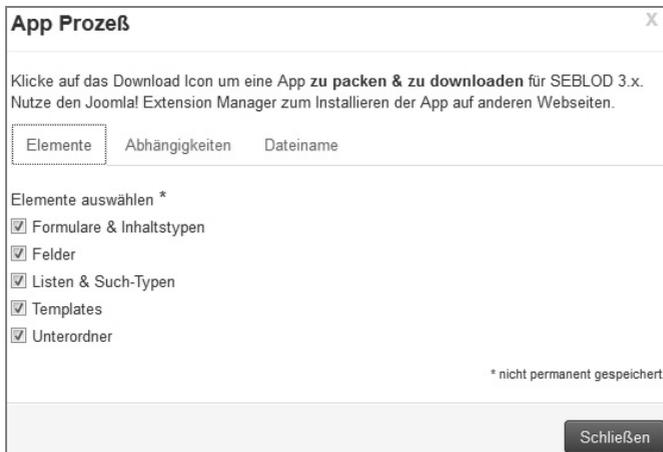
SEBLOD neigt angesichts der Fülle von Einstellungsmöglichkeiten zu Unübersichtlichkeit. In diesem Kapitel möchte ich daher weniger erklären, was welche Auswirkung hat, sondern einen Überblick darüber geben, wo was eingestellt werden kann.



# 5

## App-Optionen

Befinden Sie sich in der Übersicht der Apps, finden Sie oben links in der Buttonleiste die *App Optionen*.



The screenshot shows a dialog box titled "App Prozeß" with a close button (X) in the top right corner. Below the title bar, there is a text area containing instructions: "Klicke auf das Download Icon um eine App zu packen & zu downloaden für SEBLOD 3.x. Nutze den Joomla! Extension Manager zum Installieren der App auf anderen Webseiten." Below this text are three tabs: "Elemente" (which is selected and highlighted with a dotted border), "Abhängigkeiten", and "Dateiname". Under the "Elemente" tab, there is a section titled "Elemente auswählen \*" followed by a list of five items, each with a checked checkbox: "Formulare & Inhaltstypen", "Felder", "Listen & Such-Typen", "Templates", and "Unterordner". At the bottom right of the dialog, there is a small note: "\* nicht permanent gespeichert." and a "Schließen" button.

**Bild 5.1:** Die *App Optionen* regeln das Speichern der Apps.

Mit diesen Optionen regeln Sie, was alles gespeichert wird, wenn Sie die App downloaden und demzufolge woanders installieren. Sie können also lediglich bestimmte Strukturen Ihrer App speichern (downloaden) oder alles, was Sie dazu erstellt haben.

Im *App Prozeß* wählen Sie also aus, ob alle mit Ihrer App erstellten *Formulare & Inhaltstypen, Felder, Listen & Suchtypen*, die eigens dafür erstellten *Templates* und auch »Unter-Apps« - Unterordner - oder nur die »Haupt-App« auf der ersten Ebene gespeichert werden sollen.

Wenn Sie in Ihrer App, Ihrem Inhaltstyp, Abhängigkeiten zu *Kategorien* erstellt haben, dann haben Sie die Möglichkeit, diese Abhängigkeiten auch in Ihrer App zu speichern oder eben nicht, abhängig davon, ob der Anwender frei in der *Kategorieauswahl* ist oder nicht.

Ebenso verhält es sich beim Menü. Haben Sie entsprechende Abhängigkeiten in Ihre App eingebaut, können sie auch mit abgespeichert werden. Zur Auswahl stehen natürlich nur die in Ihrer Installation vorhandenen Menüs.

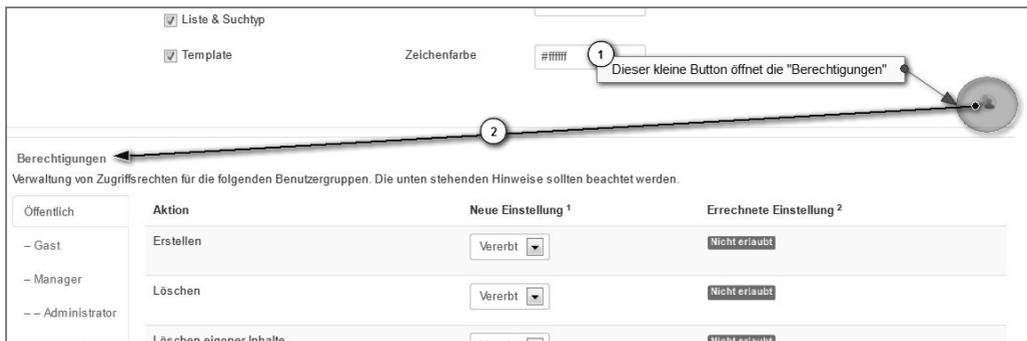
Wenn Sie die App für sich abspeichern - weil Sie sie beispielsweise lokal erstellt haben und nun auf Ihrer Website verwenden möchten -, kennen Sie natürlich Ihre *Kategorien* und *Menüs*. Wollen Sie die App aber der Allgemeinheit weitergeben, können Sie nicht davon ausgehen, dass *Kategorien* und *Menüs* auch so vorhanden sind.

Speichern Sie die App, können Sie im Tab *Dateinamen* angeben, ob das aktuelle Datum und/oder die Versionsnummer mit angehängt wird. Wenn Sie regelmäßig Apps erweitern und speichern, kann eine entsprechende Speicherung sinnvoll sein.

# 6

## ACL in der APP

Sie können natürlich eine differenzierte Rechteverwaltung in den Apps vornehmen. Die ACL<sup>18</sup> von SEBLOD lehnt sich an die von Joomla! an, sodass Sie vermutlich keine Schwierigkeiten haben werden. Sie finden die *Berechtigungen*, wenn Sie direkt die App editieren und dann den kleinen Button unten rechts anklicken.



Verwaltung von Zugriffsrechten für die folgenden Benutzergruppen. Die unten stehenden Hinweise sollten beachtet werden.

Öffentlich	Aktion	Neue Einstellung <sup>1</sup>	Errechnete Einstellung <sup>2</sup>
- Gast	Erstellen	Vererbt	Nicht erlaubt
- Manager	Löschen	Vererbt	Nicht erlaubt
-- Administrator	Löschen eigener Inhalte	Vererbt	Nicht erlaubt

Bild 6.1: Berechtigungen in der App.

<sup>18</sup> ACL = Access Control List - auf Deutsch: Rechteverwaltung

Sie können zunächst auswählen, für welche Benutzergruppe Sie die Berechtigungen anpassen möchten. Neben den Standardbenutzergruppen, die von Joomla! angelegt wurden, finden Sie hier auch etwaige selbst angelegte Benutzergruppen. Neben der *Aktion* können Sie zwischen *Erlaubt*, *Verweigert* und *Vererbt* auswählen. Den Unterschied zwischen *Erlaubt* und *Verweigert* brauche ich sicherlich nicht zu erklären. *Vererbt* bedeutet, dass die Einstellung von der darüberliegenden Gruppe übernommen wurde.

Ihre eigene App wird vermutlich mindestens unter der *Top*-App liegen. Somit erben Sie die Einstellungen von *Top*. Die *Top*-App wiederum orientiert sich an den Einstellungen in der Joomla!-Konfiguration, wo sozusagen alles seinen Ursprung hat. Um die Berechtigungen für Ihre App zu ändern, müssen Sie nicht an den globalen Joomla!-Berechtigungen herumschrauben. Es reicht völlig, wenn Sie das in Ihrer App tun und aus dem *Vererbt* ein *Erlaubt* oder *Verweigert* machen. Dann gelten die Berechtigungen nur für Ihre App und nicht für andere Apps.

*Vererbt* bedeutet stets, dass die Rechteeinstellung von der darüberliegenden Struktur (App) übernommen wird. Steht da auch ein *Vererbt*, gilt die darüberliegende Struktur und so weiter. Es kann also sehr wohl so sein, dass die tatsächlichen *Erlaubt*-/*Verweigert*-Berechtigungen sehr weit oben in der Struktur gesetzt und lediglich nach unten weitervererbt wurden. Um den Überblick zu behalten, sehen Sie aber in der rechten Spalte, was genau für die einzelne Berechtigung eingestellt wurde.

Berechtigungen		Stets den Überblick behalten	
Verwaltung von Zugriffsrechten für die folgenden Benutzergruppen. Die unten stehenden Hinweise sollten beachtet werden.			
Öffentlich	Aktion	Neue Einstellung <sup>1</sup>	Errechnete Einstellung <sup>2</sup>
- Gast	Erstellen	Vererbt ▼	Erlaubt
- Manager	Löschen	Vererbt ▼	Nicht erlaubt
-- Administrator	Löschen eigener Inhalte	Vererbt ▼	Nicht erlaubt
- Registriert	Bearbeiten	Vererbt ▼	Nicht erlaubt
-- Autor	Status bearbeiten	Vererbt ▼	Nicht erlaubt
--- Editor	Eigene Inhalte bearbeiten	Vererbt ▼	Erlaubt
---- Publisher			
- Super Benutzer			

**Bild 6.2:** Stets den Überblick bewahren.

Die einzelnen Berechtigungen von oben nach unten:

- *Erstellen* - Wenn Sie möchten, dass eine Benutzergruppe einen Beitrag (je nach App und Inhaltstyp) erstellen können soll, müssen Sie stets *Erstellen* auf *Erlaubt*

setzen. Es reicht nicht, eine der nachfolgenden Berechtigungen auf *Erlaubt* zu setzen. Wenn *Erstellen* nicht explizit erlaubt ist, kann kein Beitrag geschrieben und gespeichert werden.

- *Löschen* – Bedeutet lediglich die Erlaubnis, einen Beitrag in den Papierkorb zu verschieben. So wie Sie es vielleicht von einem MS-Windows-Rechner her kennen, landen gelöschte Dateien im Papierkorb und können dort wiederhergestellt oder endgültig gelöscht werden. In Joomla! funktioniert das genauso. Ein Beitrag kann erst im Backend endgültig gelöscht werden. Und auch nur dort kann er wiederhergestellt werden, falls er doch nicht gelöscht werden soll.

*Löschen* bezieht sich auf alle Beiträge – *Löschen eigener Inhalte* hingegen nur auf die selbst erstellten Beiträge. Ist *Löschen* auf *Verweigert* und *Löschen eigener Inhalte* auf *Erlaubt* gesetzt, kann der Autor nur seinen eigenen Beitrag löschen, nicht jedoch Beiträge, die er nicht erstellt hat.

- *Bearbeiten* – Bezieht sich auf das Editieren und Verändern von eigenen und fremden Beiträgen.
- *Status bearbeiten* – Damit ist gemeint, dass Sie einen Beitrag auf *Veröffentlicht*, *Gesperrt* oder *Archiviert* setzen und – und wenn Sie die Berechtigung zum *Löschen* haben – auch *Papierkorb* auswählen können. *Verweigern* Sie den Status, wird dieser gar nicht erst angezeigt.
- *Eigene Inhalte bearbeiten* – Damit erlauben (oder verweigern) Sie, dass vom Autor erstellte Beiträge bearbeitet werden können. Interessant ist diese Einstellung, wenn Sie verhindern möchten, dass Autoren einmal veröffentlichte Beiträge weiterhin ändern dürfen.

Sie können jederzeit – unabhängig von der Hierarchie und der daraus resultierenden Vererbung – Berechtigungen neu setzen. Nachfolgende Unterordner (Apps) haben standardmäßig ein *Vererbt* eingestellt und übernehmen die gesetzten Berechtigungen, die Sie aber natürlich in einer »Unter-App« wieder ändern können.

## 6.1 Der ACL-Generalschlüssel

Sie können Ihre individuell eingestellten Rechteeinstellungen mit dem Button in der oberen Leiste – *Leere ACL* – komplett löschen und wieder auf die Defaulteinstellungen zurücksetzen. Dazu einfach die App, deren Rechteeinstellungen auf den Ursprung zurückgesetzt werden sollen, markieren, indem Sie das Häkchen am Anfang setzen und dann den Button anklicken.



**Bild 6.3:** Alles wieder zurück auf Start.



# Optionen

Wenn Sie im Pull-down-Menü<sup>19</sup> die *Komponenten* auswählen und dann auf SEBLOD 3.x oder wenn Sie im linken Menü auf SEBLOD 3.x klicken, dann können Sie oben links die *Optionen* und *Voreinstellungen* auswählen.

Die *Optionen* bieten einige sehr interessante Dinge, die nachfolgend erläutert werden.

## 7.1 Allgemein

Zunächst können Sie einstellen, ob während der Arbeit an den Formularen eine *Automatische Speicherung* erfolgen soll und, wenn ja, in welchem *Intervall*. Eine überaus praktische Einstellung. Gerade wenn Sie in einem Büro arbeiten, das Telefon klingelt, ein Kollege von Ihnen etwas möchte oder Ihnen einfällt, dass Sie den Tee wieder einmal in der Küche stehen gelassen haben – schnell vergisst man, ob man etwas speichern wollte oder es schon hat oder doch nicht. Löschen ist schneller, als einmal das Speichern zu vergessen und wieder von vorn anzufangen. Ein *Intervall* von 10 Minuten erscheint mir persönlich recht sinnvoll, die Einstellung sollte sich aber nach Ihren Bedürfnissen richten.

*Ausgabe der Formate* ist wegen des einstellbaren *JText* vielleicht ein wenig verwirrend.

---

<sup>19</sup> Damit ist das obere horizontale Menü gemeint. Wenn Sie auf einen Hauptmenüverweis klicken, dann »pullt« das Menü »nach unten« (pull down).

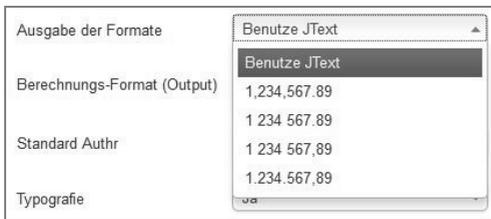


Bild 7.1: Einstellung des Zahlenformats.

In Deutschland trennen wir die einzelnen Tausenderblöcke mit einem Punkt und die Dezimalstellen mit einem Komma. Sie können aber auch den Punkt als Trenner für die Dezimalzahlen auswählen und die Tausenderblöcke wahlweise mit einem Leerzeichen, einem Punkt oder einem Komma trennen, wenn Ihnen das besser gefällt. Oder Sie überlassen das der Einstellung von *JText*. *JText* ist eine Joomla!-Klasse, die überall dort angegeben wird, wo Sprachschlüssel benutzt werden. Damit Joomla! in den vielen Sprachen funktioniert, stehen im Quellcode Sprachschlüssel, die erst zur Laufzeit (also wenn Joomla! ausgeführt wird) von der ausgewählten Sprache ersetzt werden.

Manchmal fehlt eine Übersetzung, und es wird dann der Sprachschlüssel ausgegeben. Beispielsweise könnte das so aussehen:

```
001 LIB_CCK_INTEGRATION_MULTILANGUAGE
```

In der PHP-Datei könnte dann so etwas stehen:

```
001 JText::printf('LIB_CCK_INTEGRATION_MULTILANGUAGE');
```

Im Prinzip handelt es sich um eine Art Platzhalter. Der Sprachschlüssel wird mit der Übersetzung ausgetauscht.

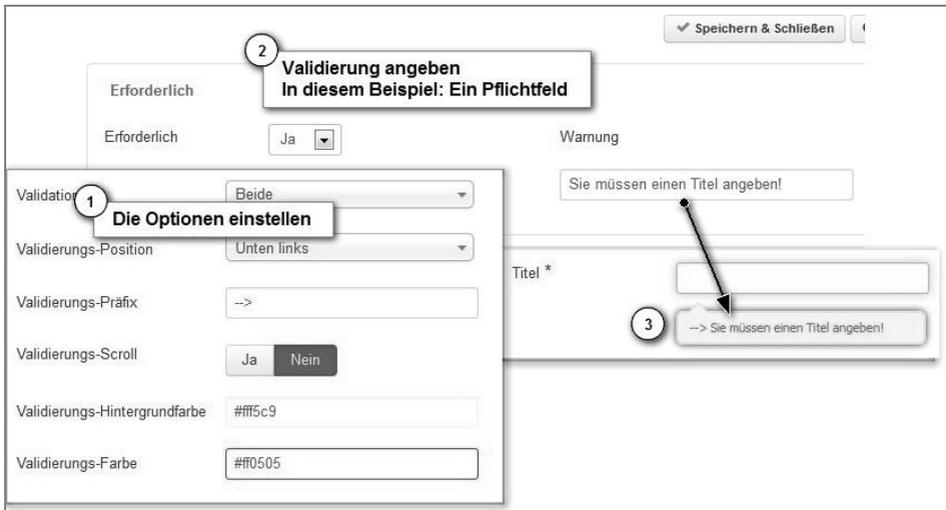
Mit der Einstellung *JText* überlassen Sie also einer Übersetzung die Ausgabe des Zahlenformats. In Deutschland ist es üblicherweise dieses: *1.234.567,89*.

*Berechnungsformat (Output)*: Laut SEBLOD-Team wird hier die Gültigkeit des Berechnungsformats nach der Kalkulation bestimmt. Leider hat sich mir das trotz der Erklärung nicht erschlossen, und ich muss Sie an dieser Stelle bitten, eigene Experimente durchzuführen.

Mit dem *Standard Autor* können Sie festlegen, wer schwerpunktmäßig neue Beiträge schreibt und somit standardmäßig als Autor festgelegt werden soll. Im Backend ist das meist der *Super User*. *Verwende Geerbt* übernimmt den tatsächlichen Autor.

Der Bereich *Valdierung* legt fest, wie Meldungen bei fehlerhafter Eingabe angezeigt werden sollen, sofern Sie entsprechende Angaben machen. Wenn Sie sich im Bereich *Formulare & Inhaltstypen* befinden, können Sie durch Anklicken von *Nummer 3* festlegen, dass es sich um ein *Pflichtfeld (Erforderlich)* handelt, und auch angeben, was für ein Wert in das Feld geschrieben werden soll (*Bestätigung*).

Als mögliche Vorgabe haben Sie *Url*, *eMail*, *Alphanumerischer Wert*, *Kategorie* und *Ajax-Prüfung*. Daneben geben Sie den jeweiligen Text ein. Möchten Sie nun einen neuen Beitrag anlegen und speichern, ohne die erforderlichen Felder zu beschreiben, erhalten Sie einen entsprechenden Hinweis:



**Bild 7.2:** Ein erforderliches Feld, seine Validierung und die Ausgabe der »Warnung«.

Sie können diese Ausgabe in den Optionen farblich gestalten. Im Hinweistext (*Warnung*) können Sie auch HTML-Angaben machen und so den Text gestalten, beispielsweise fett oder kursiv, oder einen Zeilenumbruch einfügen. Es stehen Ihnen insgesamt 255 Zeichen, einschließlich der HTML-Elemente, zur Verfügung.

Alles, was Sie hier zur Validierung einstellen, hat Auswirkung auf Ihre Validierungsangaben, jedoch nicht auf die von SEBLOD, wie beispielsweise der Hinweis auf eine erforderliche Eingabe, mit Ausnahme des *Validierungs-Präfix*, das wird auch vom SEBLOD-System übernommen.



**Bild 7.3:** Das *Validierungs-Präfix* wird geändert – die Farbangaben nicht.

Stellen Sie den *Validierungs-Scroll* auf *Ja*, gleitet SEBLOD bei einem Fehler sanft zu dem Feld und zeigt die Validierungsnachricht an. Andernfalls springt es direkt zu dem Feld.

Im unteren Bereich *Liste* können Sie in der *Seitenanzeige* einstellen, wie viele Einträge pro Seite im Frontend angezeigt werden sollen, wenn Sie dort die Listenauswahl bzw. Suche auswählen. Standardmäßig steht hier die Zahl 25, wobei diese Einstellung sich nicht auf Module, sondern ausschließlich auf den Contentbereich auswirkt. Das liegt aber an Joomla! und darauf kann SEBLOD keinen Einfluss nehmen.

*Inhalt vorbereiten* bezieht sich auf die Einstellungen in der *Seitenanzeige*. Während Sie noch auf der einen Seite auf die Liste schauen, bereitet SEBLOD bereits die nächste Seite zwecks schnellerer Anzeige vor. Wenn Sie eine große Site mit vielen Zugriffen haben, kann es sinnvoll sein, den *Inhalt vorbereiten* auf *Nein* zu stellen, um etwas Performance zu sparen.

## 7.2 Komponenten

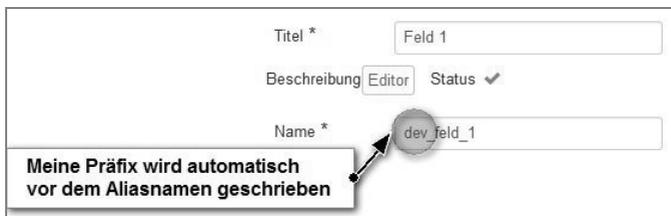
Das Register *Komponenten* wirkt sich ausschließlich auf das Backend und die Administration von SEBLOD aus.

### 7.2.1 App-Folder

Ist identisch mit den *App Optionen* und regelt, ob beim Speichern der App das Datum und/oder die Versionsnummer an den Dateinamen angehängt werden soll.

### 7.2.2 Konstruktion

Mit dem *Entwicklungs-Präfix* kann für jedes Feld (Alias) der App, des Inhaltstyps oder der Suchlisten automatisch ein Präfix vergeben werden. Das Präfix kann maximal drei Zeichen lang sein. SEBLOD fügt dann automatisch einen Unterstrich hinzu. Das ist eine übliche Schreibweise von Präfixen.



**Bild 7.4:** Präfix beim Alias eines Feldes.

Der Vorteil liegt in der Übersicht begründet. Sie können stets nach diesem Präfix suchen und sortieren. Ebenfalls kann es hilfreich sein, zusammenhängende Projekte, Inhaltstypen und Apps mit einem gemeinsamen Präfix zu kennzeichnen. Ein weiterer Vorteil ist, dass Sie damit bei mehreren ähnlichen Apps den Alias in den einzelnen Feldern leichter vergeben können, weil im Alias das Präfix vorangestellt wird.

### 7.2.3 Entwicklung

Wird der *Entwicklungsspeicher* auf *Ja* gesetzt, erhalten Sie im Speicherbereich der einzelnen Felder zusätzlich die Möglichkeit, eine Speicherform namens *Entwicklung* auszuwählen.



**Bild 7.5:** Ein neues Speicherformat.

Als Datenbanktabellenspalte erhalten Sie verschiedene *Bool-Bezeichnungen* und einmal einen Spaltennamen *Optionen*. Offenbar scheint jedoch dieser Entwicklungsbereich noch nicht so ganz ausgereift zu sein bei SEBLOD, zumindest nicht bis Manuskriptabgabe dieses Buches.

#### **Vorsicht, Absturz!**

Wenn Sie Felder, die im Speicherbereich *Entwicklung* liegen, in Ihre Formulare einbinden, kommt es zu Abstürzen. Ich hatte von umfangreichen SQL-Fehlermeldungen bis hin zu weißen Seiten alles dabei.

Es gibt dennoch eine sinnvolle Anwendung für die *Entwicklung*. Sie können bei den *Multisites optionale Felder* anlegen. Dafür können Sie die *Entwicklungsfelder* benutzen. Details lesen Sie bitte im Kapitel 20 »Multisites mit SEBLOD« nach.

### 7.2.4 Versionen

Sie können von Ihren Inhaltstypen Versionen erstellen. So können Sie zum einen überprüfen, was wann geändert wurde, und zum anderen jederzeit auf alte Versionen zurückgreifen. In den Optionen stellen Sie zum einen ein, ob überhaupt Versionen angelegt werden sollen und, wenn ja, ob nach jeder Speicherung oder nach einem bestimmten Zeitintervall.

*Nach jeder Speicherung* klingt natürlich verlockend. Ich selber speichere aber ganz oft ab, und gern auch zwischendurch. *Nach jeder Speicherung* kann daher sehr schnell zu einer wahren Versionsflut führen, die schnell unübersichtlich wird.

*Nach der Speicherung* erscheint mir sinnvoller, zumal Sie das Zeitintervall an Ihre Arbeitsweise und Ihre Bedürfnisse anpassen können. Dazu geben Sie eine beliebige

Zahl ein und wählen darunter aus, ob die Zahl sich auf das Verstreichen von *Minuten*, *Stunden* oder *Tagen* bezieht.

Mit *Version vor Wiederherstellung* wird eine weitere neue Version angelegt, bevor sie mit der älteren Version überschrieben wird. Das hat den Vorteil, dass Sie wieder zurückkönnen, wenn Ihnen die alte Version doch nicht gefällt.

Die Versionen können Sie bei den *Formularen & Inhaltstypen* einsehen, wenn Sie ganz rechts auf den kleinen, unscheinbaren Button klicken:



**Bild 7.6:** Klein und unscheinbar: der Versionsbutton.

Das setzt natürlich voraus, dass Sie mindestens zwei Versionen Ihres Inhaltstyps bereits erstellt haben. Klicken Sie dann auf diesen Button, sehen Sie Ihre Versionen dieses Inhaltstyps:

The image shows a table with columns for 'Titel', 'Version', and 'Details'. The table lists five versions of the 'Developer Test' content type. Callouts point to specific features: a downward arrow button for reverting, a blue text link for more details, a description field, and a circular button for reverting to an older version.

	Titel	Version	# Titel / Name	# Felder	ID
1	Developer Test Nochmal...	Rev. 8 20.06.2013 08:46 Super User	Developer Test dev_developer_test	2 - -	505
2	Developer Test Automatisch erstellt, bevor es zurück geht zu Rev. 1	Rev. 7 20.06.2013 08:29 Super User	Developer Test dev_developer_test	2 - 1	504
3	Developer Test Eigentlich fehlt hier was	Rev. 5 20.06.2013 08:27 Super User	Developer Test dev_developer_test	2 - -	503
4	Developer Test	Rev. 4 Super User	dev_developer_test	2 - -	502
5	Developer Test	Rev. 1 Super User	dev_developer_test	2 - -	501

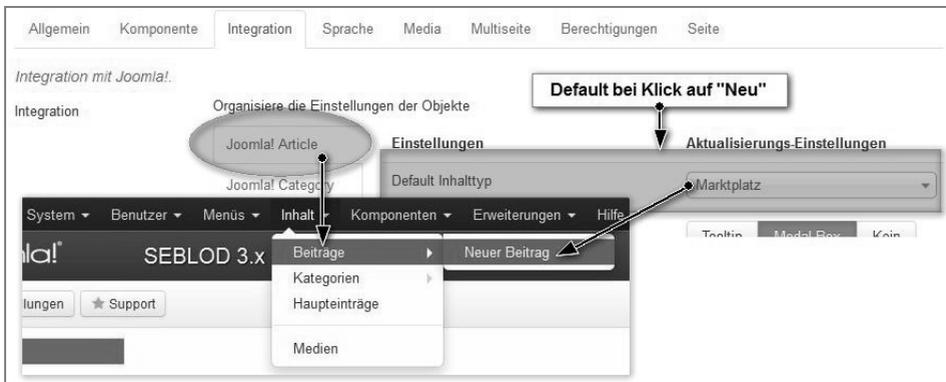
**Bild 7.7:** Alle Versionen des ausgewählten Inhaltstyps.

Ganz links der Button mit dem Häkchen gibt Ihnen die Möglichkeit, auf eine alte Version zurückzugehen. Wenn Sie das machen und in den Optionen eingestellt haben, dass eine *Version vor der Wiederherstellung* angelegt werden soll, sehen Sie bei einem erneuten Aufruf der Versionen einen entsprechenden Hinweis in der letzten Version: *Automatisch erstellt, bevor es zurückgeht zu Rev.n*

*n* steht für die Revisionsnummer, die Sie zurückgespielt haben. Neben dem Datum und wer die aktuelle Version angelegt hat, erhalten Sie auch einen Überblick darüber, wie viele Felder den einzelnen Formulartypen in der jeweiligen Revision zugeordnet waren und sind.

## 7.3 Integration

In der *Integration* lassen sich einige Einstellungen gezielt auf die einzelnen Inhaltstypen festlegen. Unter *Default Inhaltstyp* legen Sie fest, welcher Inhaltstyp ausgewählt werden soll, wenn Sie im Pull-down-Menü beim jeweiligen Hauptmenüverweis direkt *Neu* auswählen. Wenn Sie beispielsweise bei *Joomla! Article* als *Default Inhaltstyp* die *News* einstellen, wird das *Admin Formular* für die *News* geladen, wenn Sie direkt auf *Inhalt/Neuer Beitrag* klicken.



**Bild 7.8:** Den Standard für *Neu* festlegen.

Alle Einstellungen, die ich für den *Artikel* erkläre, gelten auch für die anderen Bereiche, soweit ich nicht auf Abweichungen hinweise: *Kategorien*, *Benutzer* und *Benutzergruppen*.

Mit *Add Button* ist die Auswahlbox unter *Neu* gemeint. Sie können sich zwischen einer Modalbox oder einer Liste (*Tooltip*) entscheiden.

Wenn Sie *Kein* auswählen, wird bei Klick auf *Neu* bzw. *Neuer Beitrag* das ausgewählt, was Sie unter *Default Inhaltstyp* eingestellt haben, aber es wird keine weitere Auswahl angezeigt.

*Add Button (Alternativ)* bezieht sich auf die Auswahl *Standard Joomla-Artikel*. Das können Sie wahlweise oben oder unten in der Liste oder in der Modalbox anzeigen, oder Sie schalten diese Auswahl ganz aus.

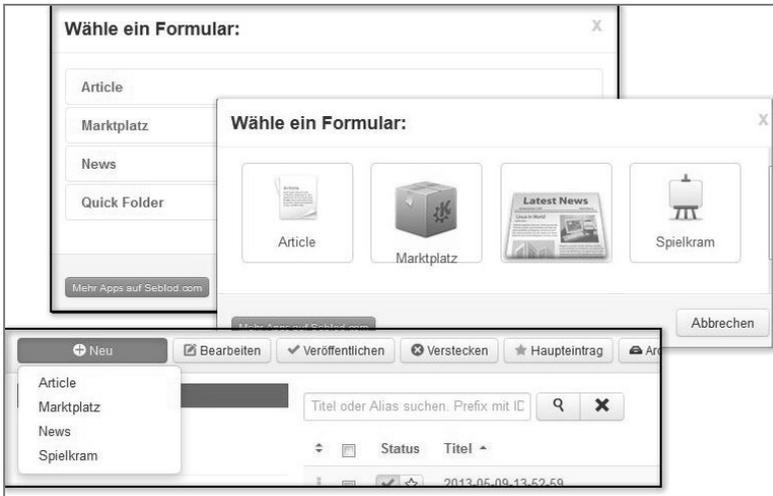


Bild 7.9: Modalbox: List, Modalbox: Icon, Tooltip.

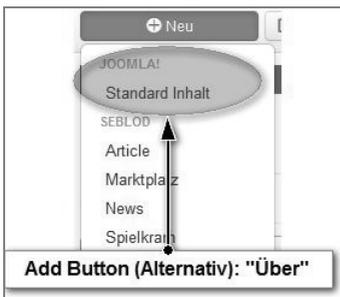


Bild 7.10: Anzeige des Joomla!-Standards.

*Add Umleitung* ist eine zusätzliche Sicherung zur Einstellung *Default Inhaltstyp*. Wenn Sie dort einen eigenen Inhaltstyp einstellen, haben Sie ja die Möglichkeit, diesen Inhaltstyp direkt beim Anklicken von *Neu* aufzurufen, ohne die detaillierte Auswahl zu erhalten. Wenn Sie nun aber *Add Umleitung* auf *Nein* stellen, erhalten Sie doch die Auswahl und stellen quasi den *Default Inhaltstyp* damit auf *Kein*. Setzen Sie hingegen *Add Umleitung* auf *Ja*, wird direkt Ihr Inhaltstyp aufgerufen. Das Einzige, was mir dazu einfällt, ist, dass Sie den direkten Aufruf temporär außer Kraft setzen können, weil Sie vielleicht einige andere Inhaltstypen erstellen möchten.

Stellen Sie den *Edit Link* auf *Automatisch*, wird jeder Beitrag in der Form geöffnet, in der er erstellt wurde. Wenn Sie hingegen *Edit Link* auf *Forciere SEBLOD* setzen, dann wird jeder Beitrag in der Form, die unter *Default Inhaltstyp* gewählt wurde, geöffnet. Auf diese Weise können Sie auch bestehende Beiträge im Nachhinein an andere Eingabeformulare – SEBLOD-Inhaltstypen – anpassen. Wenn Sie die Maus in der Beitragsübersicht direkt in die Zeile eines Beitrags bewegen, erhalten Sie neben dem

Beitragstitel einen kleinen Pfeil, mit dem Sie verschiedene Bearbeitungswerkzeuge für den Beitrag erhalten.



**Bild 7.11:** Werkzeugmenü in der Beitragsübersicht.

Wählen Sie *Edit Link (Alternativ)* in den *Optionen* aus, erhalten Sie in diesem Werkzeugmenü einen weiteren Eintrag, mit dem Sie den Beitrag in der Joomla!-Original-Beitragsansicht bearbeiten können. Mit *Multilanguage* können Sie Ihre Beiträge in mehreren Sprachen verfassen und untereinander verknüpfen. Wollen Sie lediglich Beiträge in verschiedenen Sprachen schreiben, können Sie diese Einstellung auf *Nein* stehen lassen. Es reicht, wenn Sie ein Feld vom Typ *Inhalt Sprachen* in Ihr Formular einfügen, Näheres entnehmen Sie bitte direkt dem Kapitel 8 »Alle Felder« (ab Seite 99). Wollen Sie hingegen auch die Beiträge untereinander verknüpfen, müssen Sie weitere Einstellungen an anderer Stelle vornehmen.

Zunächst müssen Sie das Plug-in *System – Sprachenfilter* in der Plug-in-Übersicht aktivieren sowie unter den *Basisoptionen* die *Verknüpften Einträge* auf *Ja* stellen. Fügen Sie nun in Ihr Formular sowohl den Feldtyp *Inhaltssprache* (Gruppe *Joomla Bibliothek (JForm)*) als auch den Feldtyp *Associations* (Gruppe *Joomla Library (JForm)*) ein.

#### Schloss offen lassen

Beide Feldtypen dürfen Sie in einem bestehenden Formular nicht mit geschlossenem Schloss einfügen. Achten Sie darauf, dass das Schloss geöffnet ist, wenn Sie das Feld in ein bestehendes Formular mittels des Pluszeichens einfügen, denn sonst werden die Spracheinstellungen zwar abgespeichert, aber in der Übersicht nicht angezeigt, und der Beitrag wird allen Sprachen zugeordnet, und es wird nicht verknüpft.

Wenn Sie nun einen gespeicherten Beitrag editieren oder einen neuen das erste Mal speichern, erhalten Sie Auswahlfelder zur Verknüpfung mit bestehenden Beiträgen einer anderen Sprache. Dabei werden alle installierten Sprachen angezeigt.



**Bild 7.12:** Beitragsverknüpfungen mit anderen Sprachen.

Wenn Sie einen Beitrag für die Verknüpfung auswählen möchten, werden Ihnen nur die Beiträge angezeigt, die der entsprechenden Sprache zugeordnet sind. Eine mögliche Anwendung ist gegeben, wenn Sie Beiträge per *Weiterlesen* anklicken und dann die Sprache wechseln. Denn es existiert für diesen Beitrag dann kein Menüverweis, und somit führt ein Sprachwechsel zur Startseite der jeweiligen Sprache, statt im Beitrag zu bleiben.

## 7.4 Sprache

Das ist ein sehr verwirrender Einstellungsbereich. Sie haben die Möglichkeit, Ihre eigenen Übersetzungen für SEBLOD zu erstellen, indem Sie die unteren Dateien downloaden. Sie erhalten eine leere Datei, lediglich mit den Sprachschlüsseln. Meines Erachtens ist das nicht mehr notwendig, seit es eine sehr gute Übersetzung gibt.

Die *Mehrsprachigkeit*, und damit *JText*, ist aber tatsächlich eine interessante Einstellung.

Joomla! selber und viele Erweiterungen haben mehrere Sprachen. Damit die Anzeige in der jeweils ausgewählten Sprache funktioniert, sind im Programmiercode Platzhalter für die jeweiligen Sprachelemente enthalten. Diese sogenannten Sprachschlüssel werden zur Laufzeit mit dem tatsächlichen Text ersetzt.

Wenn Sie eine App schreiben, die in mehreren Sprachen angezeigt werden soll, wählen Sie hier zunächst *Ja* aus. Wenn Sie nun ein neues Feld in einer App erstellen und anschließend den Inhaltstyp in den Beiträgen aufrufen, erhalten Sie statt der bisherigen Bezeichnung den Sprachschlüssel.



**Bild 7.13:** Erzeugter Sprachschlüssel statt Bezeichnung.

Im Beispiel habe ich zuvor ein neues Textfeld mit dem Titel »Eigener Text« erstellt, den ich auch als Bezeichnung gewählt habe. Der Sprachschlüssel setzt sich aus `COM_CCK_[IHR BEZEICHNUNGSTEXT]` zusammen.



Bild 7.14: Der Bezeichnungstext.

In der Abbildung sehen Sie den Bezeichnungstext in Groß- und Kleinbuchstaben. Der Sprachschlüssel benutzt aber ausschließlich Großbuchstaben und statt Leerzeichen den Unterstrich. Diesen Sprachschlüssel geben Sie am besten in die Datei `de-DE.com_cck.ini` ein, die Sie im `language`-Verzeichnis finden:

```
001 Joomla!root/language/de-DE/de-DE.com_cck.ini
```

Natürlich müssen Sie nun Sprachdateien in allen gewünschten Sprachen anlegen, die Sie im jeweiligen Unterverzeichnis der jeweiligen Sprache abspeichern. In die deutsche Sprachdatei schreiben Sie dann Nachfolgendes hinein:

```
001 COM_CCK_EIGENER_TEXT="Eigener Text"
```

Zur Laufzeit wird der Platzhalter mit Ihrem Text ersetzt.

## 7.5 Media

*Media* ist eine schicke Einstellung, wie ich finde. Zum einen können Sie für die einzelnen vorgegebenen Dateiformate die verschiedenen Suffixe festlegen. Ich denke, die Auswahl spricht für sich. Sie können die Auswahl ergänzen oder bestimmte Formate löschen. Auswirkung hat das überall dort, wo ein Autor Daten auf den Server laden kann.

Schick sind die *Freien Einstellungen*. Sie können zunächst eine Bezeichnung eingeben, also das, was bei den vorgegebenen Links steht (*Archiv-Erweiterung*, *Audio-Erweiterung* etc.), und im Feld darunter geben Sie die erlaubten Formattypen ein. So schaffen Sie sehr individuelle Upload-Möglichkeiten.

## 7.6 Multisite

Wenn Sie eine neue Seite unter *Konstruktion/Seiten* anlegen, wird in den *Optionen* automatisch auf *Ja* bei *Multi-Sites System* gewitcht. Alternativ können Sie hier auch

Ja einstellen, wenn Sie Multisites einrichten möchten. Wenn Sie eigene Felder (Details dazu im Kapitel 20 »Multisites mit SEBLOD« ab Seite 247), können Sie diese hier explizit Ihren Seiten zuweisen.

Die Berechtigungen beziehen sich entweder auf den Eigentümer der Site, der in aller Regel der Super-User (Super-Benutzer) ist, oder sind entsprechend vererbt.

## 7.7 Berechtigungen

Die Berechtigungen an sich sind identisch mit der Joomla!-ACL und betreffen an dieser Stelle ausschließlich SEBLOD selber. Damit kann also eingestellt werden, wer im Backend mit SEBLOD arbeiten darf und, wenn ja, womit und wie. Die Einstellungen hier haben nichts mit den Einstellungen bei den einzelnen Feldern und Apps zu tun. Sollten Sie unsicher sein, was die ACL in Joomla! betrifft, dann muss ich Sie leider an dieser Stelle auf allgemeine Joomla!-Literatur verweisen. Auf meiner Tutorialsite<sup>20</sup> finden Sie aber auch umfangreiche Tutorials zur ACL. An dieser Stelle würde das ein wenig den Rahmen dieses Buches sprengen.

## 7.8 Seite

Unter diesem Einstellungstab geht es in erster Linie um das Layout der Site im Zusammenhang mit SEBLOD.

### 7.8.1 Site (Frontend)

*Fehlersuche/Debug* bestimmt, wer denn überhaupt die Fehler bzw. die Debugausgabe sehen darf, wenn etwas ausgegeben wird. Das betrifft den Debugmodus, der in der Konfiguration von Joomla! eingeschaltet werden kann. Es gibt bei Joomla! ein Plug-in, mit dem Sie Details zum Debug einstellen können. Und dort können Sie auch einstellen, wer den Debugmodus überhaupt zu Gesicht bekommt. Das ist bei Live-Seiten sehr wichtig, da mit dem Debugmodus auch so einiges über die Datenbank und andere Dateien preisgegeben wird.

Also Daten, die man Menschen mit kriminellern Potenzial besser nicht zeigt. Und bei SEBLOD können Sie in den *Optionen* einstellen, ob generierte Fehler in SEBLOD angezeigt werden bzw. wer sie sehen darf. Wenn Sie vielleicht während der Entwicklung konkretere Fehlermeldungen erstellen, dann ist es sinnvoll, diese eventuell nur dem Super-Admin während der Testphase zu zeigen.

Da Sie letztlich neue Beiträge in SEBLOD erstellen, haben Ihre Autoren natürlich auch das kleine Joomla!-Icon zur Bearbeitung eines Beitrags. An dieser Stelle können Sie

---

<sup>20</sup> <http://www.time4joomla.de>

explizit angeben, dass die SEBLOD-Inhalte dieses Bearbeitungssymbol nicht erhalten sollen.

Joomla!3.x ist ein System, das explizit mobiltauglich gemacht wurde. Dazu gehört auch responsives Webdesign. Wenn Sie ein nicht responsives Template benutzen und das Browserfenster verkleinern, erhalten Sie meist eher früher als später einen horizontalen Scrollbalken. In responsiven Templates werden die Inhalte nach der Fenstergröße angeordnet. Was eigentlich nebeneinandersteht, wird untereinander abgebildet, die Navigation passt sich an, und Bilder werden verkleinert.

Damit passt sich ein responsives Template auch den unterschiedlichen mobilen Geräten an. Sie brauchen nur noch ein Template – egal was auf die Site zukommt. Eine Alternative sind Erweiterungen, die verschiedene Templates den verschiedenen Endgeräten anpassen. SEBLOD bietet Ihnen hier auf einfache Weise an, dass Sie die Inhalte automatisch anpassen und in das Template integrieren lassen. Voraussetzung ist allerdings, dass Sie auch ein responsives Template benutzen, da SEBLOD sich stets auf den eigentlichen Inhaltsbereich bezieht und natürlich nicht auf das ganze Template.

SEF steht für »Search Engine Friendly«, zu Deutsch Suchmaschinenoptimierung (SEO). Machen Sie in Joomla! eine Kategorieverlinkung und wählen dort dann einen einzelnen Beitrag aus, dann haben Sie bis zur Version 2.5x immer erst die Beitrags-ID und dahinter den Beitragsalias gesehen. Zum Beispiel in dieser Art:

```
001 .../17-mein-beitrag.html
```

SEBLOD bietet eine Menge Möglichkeiten, diese URL umzugestalten und den eigenen Bedürfnissen anzupassen. Allerdings hat sich die URL-Darstellung mit Joomla! 3.x verändert, die SEBLOD-Einstellungen folgen nur nicht der neuen Joomla!-Darstellung, und die Einstellungen in den *Optionen* zeigen auch keinerlei Wirkung. Zumindest war das so in meiner Testumgebung.

*SEF (Canonical Urls)* ist eine sehr nette Einstellungsmöglichkeit. Wenn Sie Seiten mit ähnlichen oder identischen Inhalten haben, dann wählt Google beispielsweise auf gut Glück die relevanteste Seite aus, wobei Google im schlimmsten Fall auch Webseiten mit gleichem Inhalt bestrafen kann, weil Google annimmt, Sie wollten sich einen Vorteil mit gleichen Texten verschaffen.

Oft genug haben Sie aber vielleicht nur auf verschiedenen Seiten unterschiedliche Auflistungen gleicher Produkte und oftmals gleiche Einleitungstexte etc. Sie können Google nun mitteilen, dass es sich um eine sogenannte kanonische Seite handelt, also um eine Seite, deren Inhalt auch an anderer Stelle steht, aber dass diese hier die relevanteste ist und Sie möchten, dass Google diese und nicht die anderen Seiten indiziert. Dazu schreiben Sie im Head-Bereich der Website:

```
001 <link rel="canonical" href=".../test.html">
```

Mit der Einstellung in den *Optionen* von SEBLOD wird dieser Eintrag zwischen den Head-Elementen Ihres Templates automatisch von SEBLOD eingetragen.

## 7.8.2 Scripts &amp; Styles



Bild 7.15: Style 1.



Bild 7.16: Style 2.



Bild 7.17: Style 3.



Bild 7.18: Style 4.



Bild 7.19: Style 5.

Die Einstellungen für die Modalbox beziehen sich auf die Anzeige der Bilder im Frontend, sofern Sie als Anzeigart die Modalbox ausgewählt haben. Der *Standard* entspricht dem *Style 1*. Die einzelnen Styles können Sie den kleinen Bildern hier auf der Buchseite entnehmen.

SEBLOD hat eigene CSS-Formatierungen, die Sie auch gezielt bei den Inhaltstypen verwenden können. An dieser Stelle können Sie explizit auswählen, ob die CSS-Definitionen von SEBLOD und Joomla! auch verwendet werden sollen (*Alle*) oder ausschließlich Ihre (*Benutzerdefiniert*) oder auch überhaupt *Keine*. Sie können eigene CSS-Formatierungen direkt hier in den *Optionen* eingeben, was besonders interessant mit der vorigen Einstellung *Keine* Definition sein kann.

Unter *JS* können Sie JavaScript eingeben, das sich auf das Aussehen im Frontend auswirkt.



## Alle Felder

In diesem Kapitel geht es darum, Ihnen die einzelnen Feldtypen vorzustellen und zu zeigen, was Sie wo eintragen können bzw. welche Auswirkungen das hat. Es geht mir in diesem Kapitel weniger darum, was Sie mit den einzelnen Feldtypen anfangen können – wengleich es bei einigen Feldern auch ein paar kleinere Beispiele gibt.

Bei nahezu allen Feldern sind bestimmte Eingabefelder gleich:

- Sie haben zunächst immer den *Titel* des Feldes. Der *Name* ist nur bis vor dem ersten Speichern beschreibbar und stellt den Aliasnamen dar, den SEBLOD und Joomla! intern benutzen und den Sie bei einigen Feldern, in denen Verweise zu anderen existierenden Feldern gemacht werden müssen, ebenfalls benutzen. Der *App-Folder* zeigt an, zu welcher App das Feld gehört. Mit dem *Editor* können Sie eine Beschreibung angeben. An einigen Stellen können Sie diese Beschreibung auch anzeigen lassen.
- In den *Standard Wert*, den Sie in den meisten Feldern finden, können Sie einen Wert gemäß dem Feldtyp eintragen, soweit nichts anderes in den nachfolgenden Kapiteln erwähnt wird.
- Die *Bezeichnung* ist das *Label* des Feldes, also der Name, der meistens vor dem Eingabefeld angezeigt wird.
- Den *Speicherbereich* entnehmen Sie bitte Kapite 4 («Die Speicherung und ihre Hürden«).

## 8.1 JavaScript und CSS

Unten rechts im Speicherbereich können Sie den kleinen orangefarbenen Pfeil anklicken und damit den unteren Bereich – *Mittel* – öffnen. Sie haben hier die Möglichkeit der zusätzlichen CSS- und HTML-Gestaltung für das Feld und können auch JavaScript eingeben.

In *CSS-Klassen* können Sie einen eigenen Klassennamen eintragen und über diesen dann das Feld nach Gusto formatieren. Oder Sie tragen einen der vorgegebenen CSS-Klassennamen ein, die Sie in Anhang A finden.

Unter *Eigene Attribute* haben Sie die Möglichkeit, HTML-Elementattribute einzutragen, also beispielsweise:

```
001 width="100"
```

Mit dem `style`-Attribut können Sie auch direkt CSS-Angaben an das Feld übergeben. Beispielsweise:

```
001 style="color:red;font-size:18pt;"
```

Da das Attribut `id` bereits von SEBLOD vorgelegt ist, können Sie an dieser Stelle keine neue `id` angeben. Der Eintrag `id="meineID"` wird also ignoriert.

Ebenso können Sie JavaScript eingeben. So können Sie eine Ausgabe eintragen, die bei Aufruf des Formulars angezeigt wird. Beispielsweise öffnet die JavaScript-Befehlszeile

```
001 alert ("Ein wichtiger Hinweis");
```

einen entsprechenden Hinweistext, der mit OK bestätigt werden muss. Sie können natürlich auch umfangreicheren JavaScript-Code eingeben.

Sie haben auch die Möglichkeit, mit JavaScript den Standardwert zu beeinflussen bzw. zu überschreiben. Mit folgendem JavaScript-Code können Sie in einem Feld mit dem Alias `"dummyfeld"`, aus dem SEBLOD die ID erstellt, den Standardwert überschreiben:

```
001 var text = document.getElementById("dummyfeld");
```

```
002 text.value = "Hallo SEBLOD-Welt!";
```

Auf diese Weise haben Sie auch die Möglichkeit, beispielsweise URLs zusammenzusetzen, die sich aus der Eingabe ergeben. Im folgenden Beispiel werden zwei Felder im Adminformular, wahlweise im Frontend-Formular, angelegt:

- Ein Textfeld (`beitragsid`) mit der standardmäßigen Speicherung der Artikel-ID.  
Dieses Feld kann auch gut als *Nur lesen* oder *Versteckt (hidden)* angelegt werden.
- Ein zweites Textfeld (`linkfeld`) für den JavaScript-Code, der eine URL zusammensetzt, die sich aus der Artikel-ID ergibt.

Die Speicherung kann gut auf *Benutzerdefiniert introtext* eingestellt werden, oder statt *introtext* gibt es den *link*. Wollen Sie in letztere Spalte abspeichern, können Sie auch als *Standard* speichern.

Wichtig sind die Aliase. Heißen sie anders bei Ihnen, müssen Sie die ID-Namen im Code entsprechend anpassen:

```
001 var link = document.getElementById("linkfeld");
002 var beitragsid = document.getElementById("beitragsid");
003 link.value = "index.php?option=com_content&view=article&id=" +
    beitragsid.value;
```

In `var link` sprechen Sie die ID an und weisen sie etwas weiter unten mit `beitragsid.value` dem Link zu. In `link.value` setzen Sie den Link entsprechend zusammen. Einziger Nachteil dieser Methode ist, dass Sie den Beitrag erst einmal speichern müssen, da sonst keine ID vorhanden ist. Wenn Sie so etwas einsetzen wollen, ist ein entsprechender Hinweistext, beispielsweise mit einem Freitextfeld, sinnvoll.

## 8.2 Auswahl

### 8.2.1 Dynamisch

Sie können mit diesem Feldtyp eine Verbindung zu einer Datenbanktabelle herstellen und somit den Inhalt dieser Auswahlbox aus einer Tabelle entnehmen.

In *Query* wählen Sie zunächst, ob Sie die Abfragen mit den angebotenen Feldern in diesem Feldtyp machen (*Konstruktion*) oder eine SQL-Anweisung eingeben möchten (*Frei*). Wer sich ein wenig mit SQL auskennt, findet bei der *Konstruktion* Folgendes vor:

```
SELECT [Optionswert] FROM [Tabelle] WHERE [Wo]
```

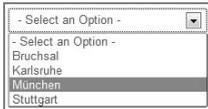
Das, was Sie in *Optionsname* eintragen, ist der Wert, der in der Auswahlbox ausgegeben wird.

Ein kleines Beispiel dazu: Ich habe in der Datenbank eine neue Tabelle mit dem Namen `#_orte` und neben der ID eine Spalte mit Postleitzahlen und mit den zugehörigen Städten angelegt.

id	plz	ort
1	70000	Stuttgart
2	76646	Bruchsal
3	76000	Karlsruhe
4	80000	München

**Bild 8.1:** Eine kleine Tabelle zum Üben.

In *Tabelle* trage ich den Namen der Tabelle ein, aus der ich meine Daten beziehen möchte, in meinem Beispiel also `#__orte`. Achten Sie dabei auf die beiden Unterstriche nach der Raute. SEBLOD ersetzt zur Laufzeit `#_` mit dem richtigen Präfix. Ich möchte die Städte in meiner Auswahlbox ausgeben, weshalb ich in *Optionsname* »ort« schreibe. Abspeichern möchte ich dann aber nur die Postleitzahl, deshalb schreibe ich deren Spaltennamen in *Optionswert*: »plz«. Das reicht bereits, um eine dynamische Auswahlliste angezeigt zu bekommen.



**Bild 8.2:** Eine dynamische Auswahlliste.

Wähle ich nun München aus und speichere den Beitrag ab, wird die Postleitzahl abgespeichert, weil ich diese im *Optionen Wert* angegeben habe.

Um die Verwirrung frühzeitig zu klären:

- Der *Optionen Name* dient der Anzeige in der Auswahlbox.
- Der *Optionen Wert* wird abgespeichert.

Sie können natürlich auch in beide den gleichen Namen eintragen. Tragen Sie beispielsweise sowohl in *Optionen Name* als auch in *Optionen Wert* »Ort« ein, wird die Stadt in der Auswahlliste nicht nur angezeigt, sondern auch gespeichert.

*Wenn* entspricht dem SQL-Befehl *Where*. Sie können in diesem Feld also eine Bedingung angeben. Weiter unten ist ein Beispiel angegeben, wie Sie die Inhalte eines Menüs auslesen können. Die dortige *Where-Bedingung*: »`menutype='mainmenu'`« wäre dann der Teil, der in das *Wenn*-Feld eingetragen wird.

*Sortierung nach* gibt Ihnen die Möglichkeit, in einer Datenbanktabellenspalte Ihrer Wahl das Auswahlfeld *Aufsteigend* oder *absteigend* zu sortieren. Wenn Sie lange Listen haben, können Sie mit *Limit* die Ausgabe auf eine bestimmte Anzahl beschränken.

Die *Optionen-Attribute* müssen erst aktiviert werden, indem Sie auf *Toggle* klicken. Dann erscheinen links unten im *Konstruktions-Bereich* einige Möglichkeiten für eigene Attribute.

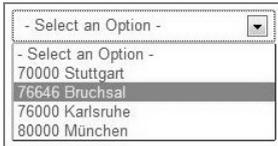
Würden Sie das direkt in das freie Feld für die SQL-Anweisung eingeben, lautete der SQL-Befehlssatz:

```
001 SELECT ort AS text,plz AS value FROM #__orte
```

Wenn Sie mehrere Spalten in der Auswahlliste anzeigen möchten, müssen Sie *Query* auf *Frei* stellen, weil Sie das nur mit SQL erreichen können. Um zwei oder mehr Spalten miteinander zu verknüpfen, geben Sie nachfolgenden SQL-Befehlssatz ein:

```
001 SELECT CONCAT(plz, " ", ort) AS text, plz AS value FROM #__orte
```

CONCAT () verknüpft die Spalten. Zwischen den Spalten habe ich noch ein Leerzeichen eingefügt. Nun haben Sie eine entsprechende Ausgabe:



**Bild 8.3:** Spalten verknüpfen.

Leider können Sie das aber nicht auch so auf den Speicherbereich, den `value`, anwenden. Gespeichert wird nur der Inhalt einer Spalte.

Ein anderes Beispiel, wie Sie die Menüverweise des Hauptmenüs auslesen können und den Verweislink abspeichern, finden Sie direkt in der Dokumentation von SEBLOD:

```
001 SELECT title AS text, link AS value FROM #__menu WHERE menutype='mainmenu'
```

Setzen Sie *Mehrfach* auf *Ja*, können Sie mehr als einen Eintrag aus der Liste auswählen. In *Reihe* geben Sie die Höhe der Auswahlbox vor. Der *Trenner* wird dann bei *Mehrfachauswahl* im Frontend zwischen den einzelnen Auswahlen angezeigt. Standardmäßig ist das ein Komma.

Die *Spracherkennung* hat nur Einfluss bei Mehrsprachigkeit auf der Website und bezieht sich darauf, wie ermittelt werden soll, woher der Benutzer kommt und ergo was für eine Sprache dann zur Verfügung gestellt wird. Joomla! hat ein Plug-in für die Steuerung der Sprache, in dem Sie einstellen können, wie sprachlich mit einem Besucher Ihrer Website umgegangen werden soll.

Wenn Sie eine Erweiterung zum Feststellen der *GeoIP* benutzen, können Sie das hier entsprechend einstellen, wobei ich Ihnen an dieser Stelle nicht sagen kann, ob SEBLOD mit anderen Erweiterungen einwandfrei zusammenarbeitet und sie von allein erkennt oder Sie in SEBLOD Entsprechendes installieren oder programmieren müssen.

Passenderweise stellen Sie bei den *Sprachencodes* ein, welche Sprachen infrage kommen und für welche Sprache das vorliegende Feld gedacht ist (*Standard*).

Mit *Füge Statische Optionen* zu können Sie zusätzliche Auswahlmöglichkeiten in die Liste bringen, wahlweise an den Anfang (*Über*) oder ans Ende (*Unter*). Beispielsweise könnten Sie damit die zusätzliche Auswahl »nicht aus Süddeutschland kommend« oder Ähnliches einfügen. Wenn Sie hinter Ihren *Optionen* noch einen Defaultwert mittels Gleichheitszeichen einfügen, wird bei Auswahl nicht der Text, sondern der Defaultwert gespeichert. Um im Beispiel zu bleiben, tragen Sie in den *Optionen*

```
001 Nicht aus Süddeutschland=1
```

ein. Es wird dann die `1` in der Datenbank gespeichert.

Im ersten Feld der *Eigenen Attributen* können Sie zum Beispiel Folgendes eintragen:

```
001 style="color:red;"
```

Die Auswahlliste erscheint dann in roter Schrift. Allerdings werden die eigenen Optionen nach wie vor in der voreingestellten Schriftfarbe angezeigt.

Dabei geben Sie in das erste Feld das Attribut und gegebenenfalls den Wert ein, und im zweiten Feld können Sie die Reihenfolge der Attribute bestimmen. Allerdings haben die Attribute einige Tücken. So war es mir nicht möglich, Attribute im Nachhinein zu löschen, auch hatte ich im Quelltext einige Merkwürdigkeiten im Code, wie diese hier:

```
001 <option produkt="" color="red" "" style="color:red;" ...
```

Sie können auch eigene Attribute definieren, wie beispielsweise Produktnummern. Allerdings können Sie entweder überall den gleichen Wert zuweisen, oder Sie müssen hier mittels Programmierung eingreifen. Wenn Sie einen Attributnamen mehrmals benutzen, wird nur der erste angezeigt. Geben Sie beispielsweise das `style`-Attribut mehrmals an – einmal in roter Farbe, an anderer Stelle mit großer Schrift –, dann wird nur die rote Farbe (in diesem Beispiel) angezeigt, also die erste Definition des Attributs.

## 8.2.2 Einfach

Eine einfache Auswahlliste können Sie mit diesem Feldtyp erstellen.

*Wähle Bezeichnung* schreibt den Text direkt in die Auswahlliste oben darüber. Sie können jederzeit neue *Optionen* hinzufügen, indem Sie das grüne Plus anklicken. Mit dem roten Minus löschen Sie *Optionen*, und mit dem blauen Punkt können Sie mit gedrückter Maustaste die Reihenfolge verändern.

Anstatt eine Auswahlliste einzugeben, können Sie sie auch gruppieren, indem Sie Überschriften erzeugen. Dazu geben Sie einfach am Ende der Überschrift `=optgroup` ein.

Mit der *Anordnung* können Sie die Liste sortieren, entweder von A bis Z oder umgekehrt. Lassen Sie es hingegen auf *Folgende Optionen* stehen, wird die Reihenfolge übernommen, die Sie anordnen.

Am Ende der Liste gibt es noch die *Eigenen Attribute* zum Auswählen. Doch leider passiert zumindest bei mir rein gar nichts, wenn ich es aktiviere. Vielleicht wird das in einer späteren Version behoben.



**Bild 8.4:** Eine einfache Liste mit Gruppierungen.

Den *Standard Wert* können Sie im Übrigen auf zwei verschiedene Arten festlegen. Entweder schreiben Sie einfach die Option in das Feld, oder Sie geben den *Optionen* Valuewerte. Dazu müssen Sie jeweils am Ende der Optionen einen Wert vergeben, beispielsweise *Äpfel=1, Birnen=2* etc., und man kann im *Standardfeld* diesen Valuewert schreiben.

### 8.2.3 Mehrfach

Mit diesem Feldtyp können Sie eine Liste anzeigen und mehrere Einträge daraus auswählen, indem Sie die `[Strg]`-Taste gedrückt halten und mit der linken Maustaste auswählen. Die *Reihen* bestimmen, wie groß diese Auswahlliste ist. Im Beispiel habe ich die Zahl 5 eingetragen.



**Bild 8.5:** Mehrfache Auswahlliste.

Der *Trenner* bestimmt dann das Aussehen im Frontend, wo die Auswahl in einer Reihe mit dem Trenner ausgegeben wird.

Die *Eigenen Attribute* habe ich im Kapitel »Checkbox« erklärt.

## 8.2.4 Zahlen

Mit diesem Feldtyp können Sie Zahlenreihen zur Auswahl bringen. Ein möglicher Einsatzzweck sind beispielsweise Geburtsjahre, Geburtstage und Geburtsmonate. Sie geben dazu *Startwert*, *Endwert* und *Zählschritt* an. Angenommen, Sie möchten die Zahlenreihe 2, 4, 6, 8 und 10 zur Auswahl bringen, dann geben Sie als Startwert die 2, als Endwert die 10 und als Zählschritt 2 ein.

Mittels des *Mathematisch*-Eingabefeldes können Sie bestimmen, wie die Schritte und somit die Reihenfolge der Zahlen dargestellt werden sollen. Nachfolgende Beispiele mit Schritt 2:

- Mit *Summe* (+) haben Sie stets eine aufsteigende Zahlenreihe: 2, 4, 6, 8, 10.
- *Unterschied* (-) gibt die Zahlenreihe von hinten nach vorn aus: 10, 8, 6, 4, 2 - Voraussetzung ist allerdings, dass der Startwert höher als der Endwert liegt.
- *Produkt* (\*) - Die einzelnen *Schritte* potenzieren sich nach jeden Schritt um die Schrittweite. Die jeweilige Zahl, die nach dem nächsten Schritt entsteht, wird mit der Schrittzahl multipliziert: 2, 4 ( $2 \times 2$ ), 8 ( $2 \times 4$ ), 16 ( $2 \times 8$ ) und so weiter bis zur maximal möglichen Zahl, die innerhalb der vorgegebenen Höchstzahl liegt.
- Der *Quotient* (/) macht das Ganze umgekehrt. Er teilt jeweils um die Zahl, die in der *Schrittweite* liegt. Bei Startwert 256: 256, 128, 64, 32, 16, 8, 4, 2.

Die beiden optionalen Angaben für die erste und die letzte Zahl sind mit etwas Vorsicht zu genießen. Geben Sie beispielsweise als Zahlenweite 2 bis 255 und Schritt 2 ein, dann wird bei 2 begonnen und bei 128 aufgehört, weil  $2 \times 128 = 256$  ergeben würde. Wenn Sie aber möchten, dass die Zahl 255 noch angezeigt wird, geben Sie diese Zahl in *Letzte* (optional) ein.

Möchten Sie zudem, dass die 0 auch mit angezeigt wird, tragen Sie sie in *Erste* (optional) ein. Tragen Sie allerdings eine Zahl ein, die in der berechneten Zahlenreihe liegt. Weil Sie nicht bei 2, sondern vielleicht erst mit der 10 beginnen wollen, erhalten Sie die Zahlenreihe: 10, 2, 4, 8, 16 etc. Die *Erste* und *Letzte* wird also lediglich vor oder nach der Liste geschrieben.

Mit *erzwingen Stellen* geben Sie die führenden Nullen an: 2, 02, 002, 0002, 00002, 000002. Die beiden optionalen Zahlen erhalten automatisch keine vorangestellten Nullen. Diese müssen Sie bei Bedarf selbst direkt ins Zahlenfeld schreiben.

## 8.3 Button

### 8.3.1 Senden

Der Button ist eine sehr faszinierende Angelegenheit in SEBLOD, da er eine Vielzahl von Einstellungsmöglichkeiten mitbringt und sich so vielfältig benutzen und einsetzen

lässt. Neben einigen Standardfeldern haben Sie sowohl die Möglichkeit, den *Task* vorzugeben als auch unter *Text/Link* dem Button ganz neue Möglichkeiten zuzuordnen.

In *Task* ist zunächst standardmäßig *Speichern & Schließen/Geernt* ausgewählt. Zwar sollte man erwarten, dass hier die Joomla!-übliche Vorgehensweise stattfindet, jedoch wurde bei mir nicht das Eingabeformular geschlossen. Stattdessen verhielt sich diese Einstellung wie *Speichern & Neu*. *Speichern & Neu* arbeitet dann aber korrekt, *Speichern* speichert den Inhalt, bleibt im Formular und löscht bisher Eingetragenes nicht. *Abbrechen* bricht die Eingabe ab, ohne etwas zu speichern.

Bei der Einstellung *Save & Redirect* hatte ich leider eine Fehlermeldung im Frontend. Dennoch wurde der Beitrag ordnungsgemäß gespeichert, und ich wurde zur Startseite gebracht. Bei der Einstellung *Save & Skip* blieb ich nach dem Speichern im Formular, das geleert wurde. Dies scheint also identisch mit *Speichern & Neu* zu sein. *Speichern & Anzeigen* speichert das Formular und zeigt anschließend die Eingabe als fertigen Beitrag an.

Sie haben die Möglichkeit, beim Button direkt einzustellen, dass der Inhalt des Formulars *Exportiert* werden soll. Dafür benötigen Sie allerdings das kostenpflichtige Add-on von SEBLOD. Wenn Sie es haben, müssen Sie zuerst in dem Add-on eine Session speichern. Diese können Sie dann direkt beim Button auswählen und so direkt aufrufen. Bei meinen Tests wurde mir aber leider angezeigt, dass der Zugriff für mich nicht gestattet sei, obwohl ich als Super-Admin angemeldet war.

Ein SEBLOD-Toolbox-Add-on *Ausführen* zu können scheint mir sehr interessant zu sein. Leider war bis Manuskriptabgabe diese Buttoneinstellung noch in Arbeit.

Sie können den Button mit einem *Etiketten-Symbol* versehen, also einem Icon. *Davor* oder *Anhängen* setzt das ausgewählte Icon vor oder nach der Buttonbeschriftung, und *Statt Text* zeigt nur das Icon ohne Buttonbeschriftung. In der Auswahlbox *Wähle* werden Ihnen diverse mögliche Icons zur Auswahl gestellt.

#### Iconanzeige

Die Icons selbst sind per Font eingebunden und liegen nicht als Grafik vor.

Die Icons werden nur angezeigt, wenn Sie als *Typ: Button* ausgewählt haben.

In *Typ* wählen Sie zwischen der älteren Definition des Buttons (`input`) und der heute üblichen Form des Buttons (`button`). Der Unterschied drückt sich per HTML aus:

```
001 <input type="submit" value="Klick mich">
```

```
002 <button type="button">Klick mich</button>
```

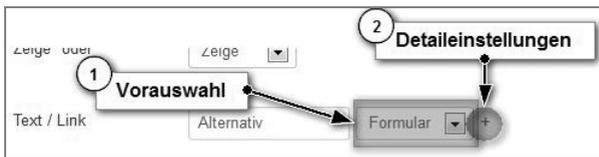
*Zeige Alternative* bietet einige Möglichkeiten. Wählen Sie *Nutze Standard* aus, dann wird neben dem Button ein *Abbrechen*-Link angezeigt, je nach Einstellung in *Zeige »oder«* mit dem Wort »oder« zwischen dem Button und dem *Abbrechen*-Link oder eben nicht.

**Hinweis**

Im Backend werden Sie nach Anklicken des *Abbrechen*-Links stets in die Übersicht von SEBLOD gebracht, im Frontend zur Startseite.

Wählen Sie bei der *Alternative* »Verbergen« aus, wird nur der Button angezeigt.

Interessant ist, wenn Sie bei *Zeige Alternative Benutzerdefiniert* auswählen, denn dann haben Sie eine Menge Möglichkeiten. Sie erhalten weiter unten die Eingabefelder zum *Text/Link*. Im ersten Eingabefeld geben Sie den Text an, der anstelle von *Abbrechen* stehen soll. Danach können Sie verschiedene Aktionen auswählen und durch Anklicken des kleinen Pluszeichens weitere detaillierte Einstellungen gemäß Ihrer Wahl einstellen.



**Bild 8.6:** Detaileinstellungen bei benutzerdefinierter Alternative.

Tragen Sie als Beispiel den Text »Kategorie erstellen« in das erste Feld (in Bild 8.6 steht dort *Alternativ*) ein und wählen Sie *Formular* aus. Klicken Sie auf das Pluszeichen. Im *Inhaltstyp Formular* wählen Sie *Category* aus. Alles andere ignorieren wir zunächst. Wenn Sie nun einen neuen Beitrag mit Ihrem Inhaltstyp erstellen und statt auf den Button auf *Kategorie erstellen* klicken, erhalten Sie das Formular zur Kategorieerstellung angezeigt. So können Sie jeden Inhaltstyp einfügen und alternativ aufrufen.

Schauen wir nun, was es bei den Inhaltstypen im Detail noch einzustellen gibt:

Wenn Sie den Inhaltstyp auf *vererbt* stehen lassen, können Sie unter *Edition* auswählen, welche Seite in dem Formular aufgerufen werden soll. Das ist besonders dann sinnvoll, wenn Sie in einem mehrseitigen Formular nicht zur nächsten Seite, sondern zu einer anderen Seite springen möchten. *Vererbt* bedeutet, dass Sie im gleichen Inhaltstyp bleiben.

Wählen Sie ein *Menüelement* aus, erscheint die neue Form unter dem ausgewählten Menüelement im Frontend, was Sie im Browser (Tab- und Fensterbeschriftung) und in den Breadcrumbs erkennen können.

In das *Klassenfeld* können Sie einen CSS-Klassennamen eintragen. Dieser wird in den Alternativlink eingefügt, und Sie können somit diesen Link individuell formatieren.

Desgleichen können Sie *Rel* im Alternativlink nutzen.

*Rel* = Relation = Bezug. Es gibt eine Reihe vorgegebener Bezüge<sup>21</sup> bei Verweisen.

<sup>21</sup> <http://de.selfhtml.org/html/verweise/typisierte.htm>

Sie können hier aber auch ein *nofollow*<sup>22</sup> eintragen, sodass eine Suchmaschine diesem Link nicht folgen würde.

Das *Ziel* bezieht sich darauf, wie der Verweis angezeigt werden soll:

- *\_blank* öffnet einen neuen Browsertab (Browserfenster).
- *\_self* öffnet den Verweis im aktuellen Fenster.
- *\_parent* bewirkt das Gleiche wie *\_self*.
- *\_top* bewirkt das Gleiche wie *\_self*.

*\_parent* und *\_top* wurden in Framesets benutzt, die heute normalerweise keine Anwendung mehr finden oder zumindest sehr selten eingesetzt werden.

Der *Titel* ist ein Verweisattribut und wird demzufolge dort auch eingefügt, sofern Sie *Benutzerdefiniert* auswählen und einen *Titel* eintragen. Der *Titel* wird als Tooltipp angezeigt, wenn Sie mit der Maus eine Sekunde über dem Verweis verweilen.

Der sich daraus ergebende Verweis sieht folgendermaßen aus:

```
001 <a class="meine_klasse" title="Mein Titel" target="_self" rel="
    "nofollow" href="#">Kategorie erstellen</a>
```

*Tmpl* bietet Ihnen die Möglichkeit, ausschließlich das Formular (*Komponente*) oder die ganze Eingabemaske (*Standard*) auszuwählen. *Nur das Formular* zeigt die obere Werkzeugleiste mit den Speicherbuttons nicht an. Das hat bei der Kategorieerstellung vermutlich nicht sehr viel Sinn, kann aber bei eigenen Formularen, die anders aufgebaut sind und eigene Buttons zum Speichern anbieten, sehr sinnvoll sein.

Die *angepassten Variablen* sind für Programmierer interessant, da sie hier Variablen an die URL mit anhängen können. Beispielsweise ergibt der Eintrag

```
001 id=1
```

am Ende der URL diesen Eintrag:

```
001 component/cck/form/article?Itemid=106&id=1
```

Der Bereich *Kein Zugang* zeigte bei meiner Version leider keine Wirkung. Ich vermute, dass es sich um einen Bug handelt, und kann aber nur raten, um was es sich handeln könnte. Ich vermute, dass in verlinkten Formularen, die nicht die Rechte haben, die notwendig sind, damit der jeweilige Benutzer sie editieren kann, der Verweis gar nicht erst angezeigt wird.

Diese Detailangaben variieren bei den verschiedenen Auswahlmöglichkeiten leicht, bedeuten aber stets das Gleiche. Die Auswahlmöglichkeiten unter *Link*:

- *Delete - Admin Form*: Damit können Sie vorhandene Inhalte in den einzelnen Feldern löschen, bleiben aber innerhalb des Eingabebereichs.

<sup>22</sup> <http://de.wikipedia.org/wiki/Nofollow>

- *Delete – Seitenformular*: Sie löschen alles, was Sie gerade eingetragen haben, der Beitrag wird also komplett gelöscht und geschlossen, und Sie werden zur Startseite weitergeleitet.
- *Inhalt*: Leider erhalte ich bei dieser Einstellung nur Fehlermeldungen und kann nichts Gescheites ein-/angeben. Eine Nachfrage beim SEBLOD-Team brachte leider auch keine Klärung.
- *Formular*: Sie können ein vorhandenes SEBLOD-Formular auswählen.
- *Liste & Suche*: Sie können einen vorhandenen Listen- und Suchtyp auswählen.

Bei der Auswahl *Delete* erhalten Sie ein Feld *Confirm*. Wählen Sie *Ja* aus, müssen Sie erst bestätigen, dass Sie den Inhalt löschen wollen. Andernfalls wird er ohne Nachfrage gelöscht.

## 8.4 Captcha

### 8.4.1 Mathematisch

Ein eigener *Captcha* in SEBLOD ist zwar sehr verlockend, doch leider muss ich zugeben, ihn nicht zum Laufen bekommen zu haben. Das zugehörige Plug-in ist natürlich veröffentlicht. Vielleicht funktioniert der *Captcha* in einer späteren Version von SEBLOD.

## 8.5 Formular

In dieser Gruppe finden Sie Formulartypen, die Sie von den HTML-Formularelementen her kennen.

### 8.5.1 E-Mail

Sie finden alles zu diesem Feld im Kapitel 16.13 »Feldtyp Email«.

### 8.5.2 Checkbox

Der *Standard Wert* hat hier eine andere Bedeutung. Falls Sie Inhalte bei den *Optionen* eingetragen haben, so versehen Sie sie jeweils mit einem Wert (z. B. Ja=1) Im Feld *Standard Wert* tragen Sie nun die Zahl des Optionswertes ein, der bereits vorselektiert sein soll.

Bild 8.7: Standard Wert setzen.

Alternativ können Sie den Eintrag in den *Optionen* natürlich auch direkt in den *Standard Wert* schreiben. Der entscheidende Unterschied ist, dass bei der direkten Eingabe des Optionenfeldes lediglich ein Kontrollkästchen aktiviert werden kann. Schreiben Sie hingegen eine der vergebenen Zahlen mit einem Komma getrennt in den *Standard Wert*, können Sie so auch mehrere Kontrollkästchen aktivieren.

Mit der *Ausrichtung* geben Sie an, ob die Kontrollkästchen nebeneinander (*Horizontal*) oder untereinander (*Vertikal*) angezeigt werden sollen. Wählen Sie *Vertikal* aus, erhalten Sie außerdem die Möglichkeit, anzugeben, in wie vielen *Spalten* die *Kontrollkästchen* ausgegeben werden sollen.

Der *Trenner* hat erst Auswirkungen im Frontend, wenn Sie das Ergebnis der Kontrollkästchen anzeigen. Hier ein Beispiel mit einer horizontalen Ausrichtung und dem vorgegebenen Komma, mit dem die zuvor angekreuzten Antworten ausgegeben werden:

Bild 8.8:  
Vom Eingabeformular zur Ausgabe.

Wenn Sie einen Haken bei *Eigene Attribute* setzen, erhalten Sie unter jeder Option sechs Eingabefelder plus Dasselbe bei den *Eigenen Attributen*. Sie können zunächst unter *Eigene Attribute* ein Attribut definieren und es bei den *Optionen* zuweisen.

Beispielsweise können Sie so Tastaturkürzel vergeben. Dazu schreiben Sie unten *accesskey* in eines der Felder und weisen unter den *Optionen* dann das Tastaturkürzel zu. In meinem Beispiel habe ich *Ja* das Tastenkürzel [A], *Nein* das [S] und *Vielleicht* das [D] zugewiesen. Wenn Sie nun einen neuen Beitrag eingeben, können Sie mit den Browsertastenkürzeln plus der zugewiesenen Taste direkt die Haken setzen bzw. löschen.

**Tastenkombinationen sind für jeden Browser unterschiedlich**

Firefox: Großschreibtaste + Alt + Accesskey

MS-Internet Explorer: Alt + Accesskey und danach Enter drücken

Google Chrome: Alt + Accesskey

Safari: Strg + Accesskey

Optionen

Ja=1

a 08

Nein=2

s 15

Vielleicht=3

d 4711

Eigene Attribute

accesskey produktnr

**Bild 8.9:** Eigene Attribute setzen.

Sie können weitere *Attribute* angeben. In meinem Beispiel habe ich eine *Produktnummer* vergeben. Vielleicht soll ja ein Skript entsprechende Abfragen machen. Sie können auch unten das Attribut `style` angeben und dann direkte CSS-Formatierungen oben vergeben. Allerdings war es bei mir so, dass nur jeweils ein CSS-Eintrag übernommen wurde, andernfalls gar nichts. Und leider habe ich auch nicht den gewünschten Effekt damit erreicht. Aber vielleicht funktioniert das in einer späteren Version oder bei einem anderen Feld. Da es sich um Attribute handelt, werden sie natürlich auch direkt in den zugehörigen HTML-Elementen angegeben:

```
001 <input id="dev_kontrollstchen2" class="checkbox" type="checkbox"
    produktnummer="4711" accesskey="d" size="1" value="3" name="dev_
    kontrollstchen[]">
```

### 8.5.3 Passwort

Geben Sie hier etwas ein, wird der Inhalt mit Punkten angezeigt, also ganz so, wie es sich für ein Passwortfeld gehört. Geben Sie einen *Standard Wert* vor, wird er ebenfalls in Punkten angezeigt.

### 8.5.4 Radiobutton

Bei den *Radiobuttons* gilt das Gleiche, wie in Kapitel 8.5.2 »Checkbox« bei den Kontrollkästchen beschrieben.

### 8.5.5 Text

Mit *Mindestlänge* legen Sie fest, wie viele Zeichen mindestens eingetragen werden müssen. Die *Maximallänge* gibt an, wie viele Zeichen maximal eingetragen werden können. Sie können dann auch nur bis zu dieser *Maximallänge* Zeichen eingeben. Die *Größe* bestimmt die Breite/Länge des Feldes. Wenn Sie mit dem Standard-Template arbeiten, passiert allerdings nichts, wenn Sie den Wert ändern, was daran liegt, dass das Template eine feste Breite<sup>23</sup> für Formulare vorgibt.

### 8.5.6 Versteckt

Sie können auch Felder mit angeben, die bei der Eingabe nicht sichtbar sind, aber im Frontend mit ausgegeben werden können. Ein möglicher Einsatz könnte eine Landing Page sein, auf der Sie ein Formular haben. Dann könnten Sie ein bereits vorhandenes Formular nutzen und mit einem verborgenen Feld den Titel der Landing Page weiterreichen, indem Sie im *Standard Wert* einen entsprechenden Hinweis eintragen.

## 8.6 HTML

### 8.6.1 div

Das HTML-Element `div` steht für einen Bereich. Sie können mit `div` Bereiche im Frontend zusammenfassen und per CSS formatieren.

Wenn Sie im Formular *Inhalt* oder *Einleitung* die *Nummer 3* anklicken, können Sie bei `div` ein *Markup*, eine *CSS-Klasse*, vergeben. Dann können Sie anschließend per CSS entsprechend formatieren. Geben Sie beispielsweise im *Markup* »mein\_ende« ein, erhalten Sie im Quellcode des Frontends diese Ausgabe:

```
001 <div class="mein_ende">...</div>
```

<sup>23</sup> Beim Standard-Template *isis* finden Sie in der *template.css* in Zeile 773 die Angabe für die *input-Breite* (`width:210px;`). Löschen Sie diese, gilt die Einstellung im Feld.

## 8.6.2 Icon

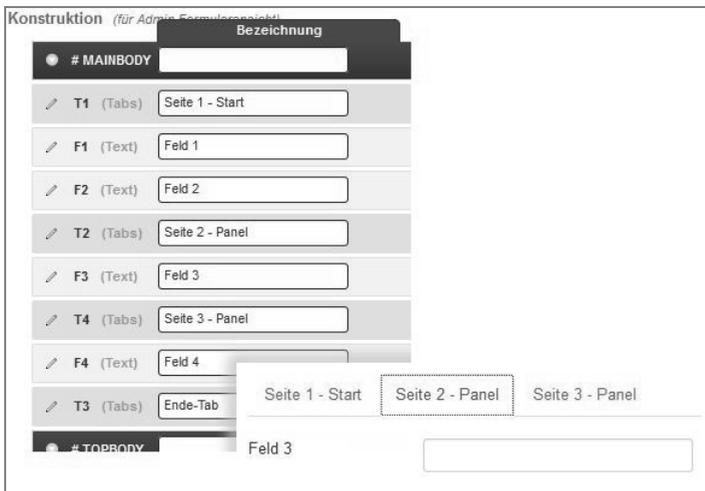
Wählen Sie den Feldtyp *Icon* aus, haben Sie die Möglichkeit, ein Icon auszuwählen, mit dem Sie Ihr Frontend gestalten können.

Die Icons selber finden Sie im Joomla!-Root-Verzeichnis unter:

● `/media/system/images/`

## 8.6.3 Tabs

Natürlich können Sie ein Formular auch als Tab-Ansicht abbilden. Wichtig ist hier die Auswahl *Auftreten*. Der Tab-Block beginnt stets mit dem *Start* und endet mit dem *Ende*. Dazwischen kommen beliebig viele *Panels*.



**Bild 8.10:**  
Tab-Ansicht.

Der *Start-Tab* leitet nicht nur den Tab-Block ein, sondern ist auch gleich der erste Tab. Darunter kommen so viele Felder, wie Sie möchten. Dem nächsten Tab weisen Sie dann ein *Panel* zu und legen auch hier wieder so viele Felder an, wie Sie möchten. Wenn Sie weitere Tabs wünschen, dann geben Sie ab jetzt jedem Tab ein *Panel* im *Aufkommen*. Wollen Sie den Block beenden, setzen Sie am Ende ein Tab mit dem *Aufkommen* »Ende«. Sie können vor oder nach dem Tab-Block natürlich beliebig viele Felder legen.

### Tab-Felder und übrige Felder

Die Tab-Felder sind übrigens etwas dunkler als die übrigen Felder, sodass Sie einen guten Überblick bei der Gestaltung haben.

In *Bezeichnung* tragen Sie die Tab-Beschriftung ein. Zum *Standard Wert* kann ich Ihnen leider nichts sagen. Sie können die Felder in einem Tab auch mit einer *Gruppen-*

bezeichnung belegen. Das hat aber erst dann Einfluss, wenn Sie mehrere Tab-Blöcke definieren. Ein paar Beispiele zu den Tabs:



Bild 8.11: Beispiele mit Tabs.

*Tab 1* und *Tab A* haben jeweils *Start* im *Auftreten* ausgewählt. Die beiden Felder *Tab END* wurde entsprechend das *Ende* zugewiesen. Die Tabs dazwischen erhalten jeweils ein *Panel*. Wenn Sie das so speichern und zur Anzeige bringen, stehen alle Tabs in einer Reihe, und bei den Feldern entsteht ein scheinbares Durcheinander.

*Feld A* und *Feld B* wechseln die Tabs entsprechend durch, *Feld C* und *Feld D* tun das zwar auch, aber die Felder der zweiten Tab-Reihe stehen direkt unter den Feldern der ersten Tab-Reihe und wechseln erst, wenn die zugehörigen Tabs angeklickt werden. Solange Sie nicht auf *Tab B* klicken, steht in der zweiten Reihe immer *Feld C*.



Bild 8.12: Verwirrendes Tab-Verhalten.

Falls Sie ein solches Szenario bauen möchten, müssen Sie einen klaren Kopf bewahren, da schnell der Überblick verloren gehen kann.

Wenn Sie das erste *Tab END* löschen – also zwischen *Tab 2* und *Tab A* –, aber *Tab A* weiterhin als *Start* definieren, dann bekommen Sie eine Variante vom obigen Effekt. In *Tab 1* und *Tab 2* stehen jetzt nur noch die zugewiesenen Felder. In *Tab A* und *Tab B* bleibt aber das letzte Feld des ersten Blocks – also *Feld B* – stehen:

Bild 8.13: Verwirrung, Teil 2.

Fügen wir das erste *Tab END* wieder ein und vergeben wir nun unseren zwei Tab-Blöcken Gruppenbezeichnungen. Dazu editieren Sie bitte die einzelnen Tabs und weisen dem oberen Tab *Gruppe\_1* zu. Sie können natürlich auch fantasievollere Namen nehmen. Wichtig ist aber, dass Sie keine Leerzeichen schreiben. Nehmen Sie stattdessen Unterstriche oder Bindestriche.

#### Ohne Umlaute und Sonderzeichen

Umlaute haben bei mir funktioniert, generell sollten Sie aber solche Dinge ohne Umlaute, Sonderzeichen und Leerzeichen schreiben.

Es reicht bereits, den ersten Block mit einer *Gruppenbezeichnung* zu belegen, um unsere Tabs in zwei Reihen zu sehen:

Bild 8.14: Tabs-Blöcke untereinander.

Wenn Sie nun wieder das erste *Tab END* entfernen und speichern, haben Sie den Effekt, dass der erste Tab allein angezeigt wird mit seinem zugewiesenen Feld, und beim Anklicken von *Tab 2* wird der zweite Tab-Block sichtbar:

Bild 8.15: Der zweite Block im zweiten Tab.

Wenn Sie mehr als zwei Tab-Blöcke erstellen möchten, arbeiten Sie mit mehreren unterschiedlichen *Gruppenbezeichnungen*, für jeden Block einen eigenen.

## 8.7 Inhalt

### 8.7.1 Freier Text

Wie der Name bereits vermuten lässt, können Sie beliebigen Text anlegen und in Ihr Formular bringen. Damit können Sie beispielsweise Hilfstexte zum Ausfüllen anlegen. Klicken Sie beim *Standard Wert* einfach auf den *Editor*, der dann in einer Modalbox aufgeht.

Wenn Sie Bilder in den Text eingeben, werden sie nur im *Seitenformular* angezeigt. Im Adminformular wird lediglich der Text ohne Bild ausgegeben.

### 8.7.2 Link

Sie können Felder zum Eintragen von Verweisen benutzen. Leider haben bei mir alle *Standards* in diesem Feldtyp nicht funktioniert. Nichtsdestotrotz können Sie einiges hier einstellen. Unter *Zeige Link* tragen Sie den Bezeichnungstext des Linkfeldes ein.

*Zeige Text* erlaubt es, einen Verweistext anzulegen. So wird nicht die URL später im Frontend angezeigt, sondern stattdessen der Verweistext.

Das *Ziel* bezieht sich darauf, wie der Verweis angezeigt werden soll:

- *\_blank* öffnet einen neuen Browsertab (Browserfenster).
- *\_self* öffnet den Verweis im aktuellen Fenster.
- *\_parent* bewirkt das Gleiche wie *\_self*.
- *\_top* bewirkt das Gleiche wie *\_self*.

*\_parent* und *\_top* wurden in Framesets benutzt, die heute normalerweise keine Anwendung mehr finden oder zumindest sehr selten eingesetzt werden.

Sie können das *Ziel* ausblenden und damit rein theoretisch Ihr voreingestelltes *Ziel* automatisch setzen - wenn es denn funktionieren würde. So wird standardmäßig stets *\_blank* ausgeführt.

*Zeige Klasse* gibt die Möglichkeit, eine CSS-Klasse anzugeben.

Mit *Zeige Vorschau* wird Ihnen nach dem ersten Speichern der fertige Link unter dem Eingabeformular angezeigt.

Ein Praxisbeispiel finden Sie in Kapitel 3.1.2 »Direkt zu den Feldern« (Feldtyp Link).

## 8.8 Joomla-Bibliothek (JForm)

In dieser Gruppe finden Sie spezifische Joomla!-Felder. *JForm* kennzeichnet dabei die Joomla!-Klasse, was vielleicht für die Programmierer unter Ihnen interessant ist. Die Feldtypen in dieser Gruppe zeigen bereits vorhandene Daten an, die dann weiterbenutzt werden können, wie beispielsweise die Kategorien.

The screenshot shows a Joomla! user management interface. The 'Benutzer' field is set to 'Super User'. Under 'Benutzergruppen', the 'Super Benutzer' group is selected with a checked checkbox. Other groups listed include 'Öffentlich', 'Gast', 'Manager', 'Administrator', 'Registriert', 'Autor', 'Editor', and 'Publisher'. Below this, there are several dropdown menus: 'Editoren' (set to '- Standard verwenden -'), 'Hilfeseiten' (set to 'English (GB) - Joomla! help wiki'), 'Inhaltssprache' (set to 'Alle'), 'Medien' (with an eye icon and 'Auswählen' button), 'Template Stil' (set to '- Standard verwend'), and 'Zeitzone' (set to '- Standard verwenden -').

**Bild 8.16:** Diese Felder sind in der Handhabung nahezu identisch.

Die meisten Felder sind in der Handhabung identisch und bieten keine weiteren Einstellungsmöglichkeiten mit Ausnahme von *Bezeichnungsfeld* und *Standard Wert*.

### 8.8.1 Assoziationen

In diesem Feld gibt es nichts Weiteres einzustellen. Sie erhalten ein Feld, mit dem Sie einen Beitrag auswählen können, der mit dem aktuell Editierten in einer anderen Sprache verknüpft sein soll. So können Sie Beiträge, die in mehreren Sprachen vorliegen, miteinander in Beziehung setzen. Weitere Details entnehmen Sie bitte Kapitel 7.3 »Integration«.

The screenshot shows the 'Article Associations' field. It displays 'German (DE-CH-AT)' and a 'Select an Article' button with a 'Select' button next to it.

**Bild 8.17:** Beiträge verschiedener Sprachen miteinander verknüpfen.

## 8.8.2 Benutzer

Sie erhalten ein Auswahlfeld, in dem Sie nach dem Anklicken des kleinen Buttons rechts neben dem Feld einen Benutzer auswählen können.

## 8.8.3 Benutzergruppen

Die vorhandenen Benutzergruppen werden angezeigt und können per Checkbutton ausgewählt und zugewiesen werden.

## 8.8.4 Editoren

Die installierten *Editoren* werden in einer Auswahlbox angezeigt.

## 8.8.5 Elternguppe

Eine etwas verwirrende Feldtypbezeichnung. Es handelt sich um die Benutzergruppen, die mit diesem Feldtyp als Auswahlbox ausgegeben werden, allerdings nur, wenn Sie einen Wert im *Standard Wert* vorgeben. Die *ID* finden Sie bei den *Benutzergruppen*. Geben Sie keinen Wert vor und lassen das Feld leer, wird der ganze Feldtyp nicht angezeigt.

Wenn Sie den Inhalt nicht *Benutzerdefiniert* abspeichern möchten, ist die *Standardspeicherung* in der *Benutzergruppe* unter *parent\_id*.

## 8.8.6 Hilfeseiten

Sie können in einer Auswahlbox eine Hilfeseite auswählen. Allerdings gibt es nach wie vor nur eine Hilfeseite: das englischsprachige Joomla!-Wiki. Somit ist dieses Feld eigentlich überflüssig. Aber vielleicht ändert sich das ja irgendwann. Es gab zumindest einmal Bestrebungen des deutschen Übersetzungsteams<sup>24</sup>, einen deutschsprachigen Helpserver einzusetzen.

## 8.8.7 Inhaltssprache

Sie erhalten ein Auswahlfeld für die installierten Sprachen. Weitere Details finden Sie in Kapitel 7.3 »Integration«.

---

<sup>24</sup> <http://www.jgerman.de>

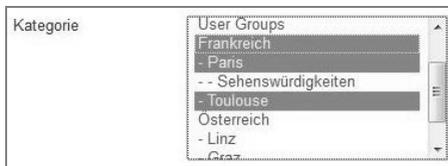
### 8.8.8 JForm – Kategorie

Mit diesem Feld wird Ihnen eine Auswahl der vorhandenen Kategorien angezeigt. Wenn Sie einen *Standard Wert* vorgeben möchten, müssen Sie die Kategorie-ID eingeben, die Sie in der Kategorieübersicht finden.

Unter *Erweiterung* können Sie angeben, auf welche Komponente sich die Kategorien beziehen. Geben Sie nichts ein, bezieht sich die Kategorieauswahl standardmäßig auf die Beiträge. Sie können so beispielsweise die Kategorien eines Forums oder des Eventmanagers auflisten. Eingeben müssen Sie die Komponente mit dem Namen, wie er im Komponentenverzeichnis angelegt ist. Herausfinden können Sie das unter anderem mit einem FTP-Programm. Beziehen Sie sich beispielsweise auf die Beiträge, müssen Sie *com\_content* eingeben.

Mit *Wähle Bezeichnung* können Sie den Text, der direkt in der Auswahlbox angezeigt wird, verändern.

Wählen Sie bei *Mehrfach Ja* aus, erhalten Sie ein neues Feld, in dem Sie die Anzahl der Reihen angeben können. Sie können nun mehrere Kategorien auswählen, indem Sie die **[Strg]**-Taste gedrückt halten, während Sie mit der Maus auf die Kategorien klicken.



**Bild 8.18:** Mehrfachauswahl bei den Kategorien.

Das ist eine sehr nette Möglichkeit, denn von Haus aus ist es zumindest in der Joomla!-Version 3.2.2 nicht möglich, einen Beitrag mehreren Kategorien zuzuordnen.

Wenn Sie *Mehrfach* auf *Ja* gesetzt, mehr als eine *Reihe* und anschließend *Mehrfach* zurücksetzen auf *Nein* eingestellt haben, ist die Anzeige der Kategorieauswahl fehlerhaft. Sie müssen zunächst die *Reihe* wieder auf *1* stellen, bevor Sie *Mehrfach* auf *Nein* setzen.

Endlich eine Mehrfachkategorieauswahl in Joomla! – allerdings hat die Sache zumindest bis zur Version 3.2.2 noch nicht funktioniert. Ich erhalte lediglich eine SQL-Fehlermeldung, und solange ich nicht explizit die Mehrfachauswahl bei den Kategorien abschalte, kann ich den Beitrag nicht speichern. Das SEBLOD-Team hat auch schon zugegeben, dass dort noch ein wenig nachgearbeitet werden muss. Vielleicht funktioniert es ja in einer späteren Version von SEBLOD.

### 8.8.9 JForm – Zugangslevel

Dieser Feldtyp gibt die vorhandenen *Zugriffsebenen* als Auswahlbox vor. Unter *Standard Wert* müssen Sie eine *Zugriffsebenen-ID* vorgeben, die Sie bei den Zugriffsebenen finden.

Die *Bezeichnung* finden Sie am Anfang der Auswahlbox wieder. Wenn Sie den Text standardmäßig anzeigen wollen, müssen Sie den *Standard Wert* frei lassen, sonst wird dieser Text erst angezeigt, wenn die Auswahlbox »aufgeklappt« wird.

### 8.8.10 Komponenten Layout

Damit haben Sie die Möglichkeit, unterschiedliche Layouts für die Beitragskomponente auszuwählen.



Bild 8.19: Auswahl alternativer Layouts.

Sie können in Joomla! jederzeit ein Override einfügen, in dem Sie das Aussehen der Ausgabe individuell anpassen können. Wenn Sie ein Override für die Beiträge programmiert haben, können Sie es hier auswählen.

Unter *Erweiterung* können Sie einstellen, auf welche Komponente sich die Auswahl bezieht bzw. welches Override<sup>25</sup> angezeigt wird und auf welche Komponente es sich auswirkt.

*Ansicht* kommt dann ins Spiel, wenn Sie zu einer Komponente verschiedene mögliche Ansichten haben. Das ist beispielsweise bei der Beitragsansicht so, wo Sie unter anderem zwischen den Kategorieansichten (*Blog* und *Liste*) und der Einzelansicht (*Einzelner Beitrag*, *Archiv*) wählen können.



Bild 8.20:  
Verschiedene  
Ansichten einer  
Komponente.

### 8.8.11 Medien

Sie erhalten ein Auswahlfeld, mit dem Sie auf den Medienordner zugreifen und dort ein Bild auswählen können.

<sup>25</sup> Auf Joomla.org gibt es weitere Informationen zu Overrides, allerdings in Englisch. Die Dokumentation gilt aber auch für Joomla! 3x: [http://docs.joomla.org/Layout\\_Overrides\\_in\\_Joomla\\_2.5](http://docs.joomla.org/Layout_Overrides_in_Joomla_2.5)  
Mittlerweile gibt es auch eine deutschsprachige Seite, die sich mit Overrides beschäftigt:  
<http://www.j-over.de>

### 8.8.12 Menüeintrag

Mit diesem Feld können Sie einen vorhandenen Menüverweis auswählen.

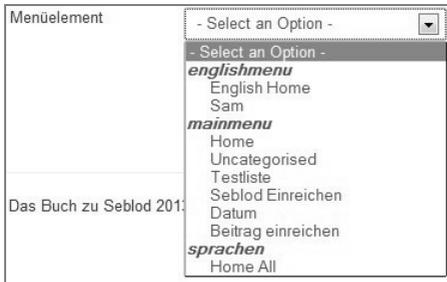


Bild 8.21: Auswahl eines Menüverweises.

Als *Standard Wert* müssen Sie eine Menü-ID eintragen. Sie finden sie im Überblick des Menüs.

*Wähle Bezeichnung* ändert den Text direkt in der Auswahlbox.

### 8.8.13 Regeln

Mit diesem Feldtyp erhalten Sie die komplette Rechteverwaltung. Geben Sie als *Ort Standard* an, wird die Rechteverwaltung direkt angezeigt. Alternativ können Sie die Rechteverwaltung in einem modalen Fenster ausgeben. Im Formular erhalten Sie dann einen Button mit der Aufschrift *Berechtigungen*.

*Erweiterung* und *Abschnitt* verhalten sich genau so, wie beim Feldtyp *Kategorie* beschrieben, wobei *Abschnitt* mit *Ansicht* gleichzusetzen ist.

### 8.8.14 Schlagwörter

Hinter den Schlagwörtern versteckt sich das Tag, das seit Joomla! 3 als neues Feature dazugekommen ist. Sie können diese Schlagwörter den einzelnen Beiträgen hinzufügen. Nach diesen Tags können Sie später im Frontend die Inhalte filtern. Um Tags eingeben oder vorhandene auswählen zu können, wählen Sie im Speicherformat *Standard* aus und bei den Tabellenspalten *tags*.

Sie können zwischen verschiedenen Modi wählen:

- *Nutze Allgemein*: Verhält sich exakt wie *Ajax*.
- *Ajax*: Kann leider nicht mit Umlauten umgehen. Zumindest wurden meine vielen »Bären« im Beispiel nicht angezeigt. Gab ich aber die Buchstabenfolge »lin« ein, wurden ordnungsgemäß die »Bärlinade« und der »Berliner Bär« angezeigt. *Ajax* merkt sich diese Liste, und wenn Sie eines davon auswählen und dann wieder in das Feld klicken bzw. den Cursor mit den Pfeiltasten Ihrer Tastatur nach unten

bewegen, wird die bestehende Restliste wieder angezeigt. Wähle ich also die »Bärlinade« aus, wird beim erneuten Anklicken des Feldes der »Berliner Bär« zur Auswahl angezeigt. Damit ist quasi eine Mehrfachauswahl möglich.

- *Nested*: Sie erhalten eine komplette Liste der möglichen Tags angezeigt, wenn Sie mit der Maus in das Feld klicken. Geben Sie mindestens drei Buchstaben ein, werden Ihnen alle möglichen Tags mit diesen drei Buchstaben angezeigt, die in heller gelber Farbe leicht hervorgehoben werden. Wenn Sie ein Tag aus der Liste ausgewählt haben und erneut in das Feld klicken, wird Ihnen wieder die komplette Liste angezeigt.

*Allow Submission* – Wenn Sie hier *Ja* auswählen, können Sie neue Tags hinzufügen. Bei *Nein* können Sie nur aus den vorhandenen Tags auswählen.

#### Platz beim Aufklappen

Wenn Sie das Tagfeld ganz unten in Ihrem Formular setzen, klappt das Feld nicht vollständig auf. Ich habe einige Felder unter dem Tagfeld eingefügt, sodass das Feld genügend Platz zum Aufklappen nach unten hat.

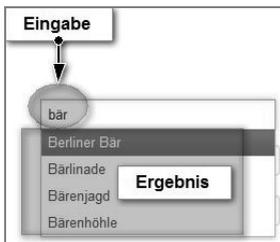


Bild 8.22: Gefilterte Bären bei den Tags.

### 8.8.15 Template: Stil

Die installierten Templates werden aufgelistet und können in einer Auswahlbox ausgewählt werden.

### 8.8.16 Zeitzone

Jedem seine *Zeitzone*. Mit dieser Auswahlbox kann sie ausgewählt werden.

## 8.9 Joomla!

Ein paar fertige Joomla!-Bauteile, die Sie verwenden können.

### 8.9.1 Artikel (Ähnliche)

Mit diesem Feld können Sie einen Beitrag auswählen, der zu dem aktuellen gehört und weitere passende Inhalte vermittelt, sowie einen Hinweis zu den AGB oder zum Datenschutz oder anderen Beiträgen geben.

Sie wählen die *Kategorien* aus (bei Mehrfachauswahl halten Sie die `[Strg]`-Taste fest), aus denen eine Liste von Beiträgen erstellt wird, aus der Sie dann einen passenden Beitrag auswählen können. *Wähle Bezeichnung* ist der Text, der in der Auswahlbox steht. In *Anzeige (Übergeordnet)* wählen Sie nun aus, wie der zweite Beitrag im Frontend ausgegeben wird:

- *Intro* gibt den Text wieder, der bis zum *Weiterlesen*-Button reicht, oder, wenn kein *Weiterlesen* vorhanden ist, den ganzen Beitrag.
- Bei *Link* erhalten Sie wieder ein kleines Pluszeichen. Im Frontend wird Ihnen nun ein Link zu dem Beitrag angezeigt.
- *Titel* gibt einfach nur den Titel des Beitrags aus.

Wenn Sie *Link* ausgewählt haben und auf das Pluszeichen klicken, haben Sie zunächst die Möglichkeit, die *SEF Urls* anzupassen.

#### Hinweis

Bei mir war es notwendig, hier */parent/id-alias* auszuwählen, da sonst der Link zu einer 404-Statusmeldung führte.

Mit *Menüelement* können Sie einen Menüverweis auswählen, der, auch wenn er mit dem eigentlichen Beitrag, der bei Klick auf den Link angezeigt wird, nichts zu tun hat, angezeigt wird.

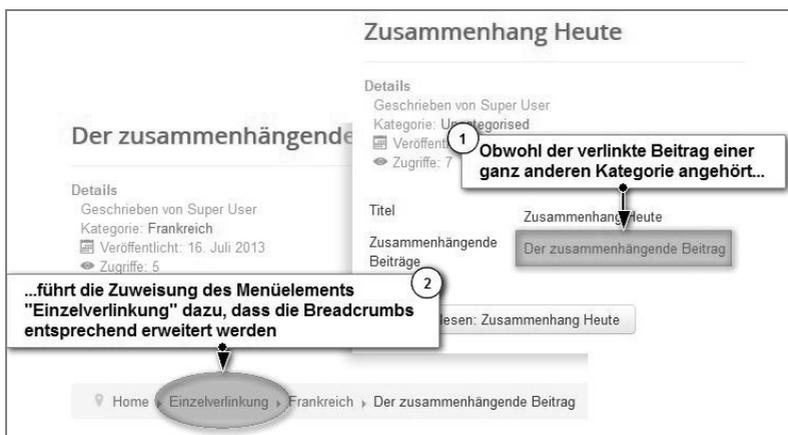


Bild 8.23: Nach der Zuweisung eines Menüelements.

## 8.9.2 Module

Mithilfe dieses Feldes können Sie ein Modul in Ihren Beitrag einfügen. Zunächst müssen Sie sich entscheiden, ob Sie ein Modul über seinen *Namen (&Titel)* ansprechen möchten oder über seine *Position*.

Wählen Sie *Name (&Titel)* aus, tragen Sie in das Feld *Position oder Name, Titel* den Modulnamen und, wenn Sie mehrere Module vom gleichen Typ haben, auch den Titelnamen des Moduls ein, das Sie abbilden möchten. Bei *Name* müssen Sie allerdings den Namen eintragen, den das Modul auf dem Webserver trägt. Den können Sie beispielsweise mit einem FTP-Programm im Verzeichnis *module/* sehen.

Angenommen, Sie möchten das Log-in-Modul einfügen, tragen Sie den Namen *mod\_login* ein. Wenn Sie mehrere Log-in-Module definiert haben, tragen Sie den *Titel* des Moduls ein, das Sie in diesem Formular abbilden möchten. Haben Sie nur ein Modul dieser Art veröffentlicht, können Sie den *Titel* auch leer lassen.

Alternativ können Sie eine *Position* angeben. Wählen Sie dazu im *Mode Position* aus und tragen Sie beispielsweise in *Position* »apfelkuchen« ein. Wenn Sie nun eines oder mehrere Module auf der Position *apfelkuchen* veröffentlichen, werden sie im Beitrag angezeigt.

Im *Stil* können Sie einige Vorgaben einstellen, die sich auf die Anzeige des Moduls im Quellcode auswirken und mittels CSS unterschiedlich angesprochen werden können. Die einzelnen *Stile* im Überblick:

- *Keine* - Entfernt das eingebettete DIV-Element mit der Klassenangabe zur `modletable` und auch das `h3`-Element in den Modulen. Das Modul wird also ohne Titel (Überschrift) ausgegeben. Das gesamte Modul kann nicht per CSS-Formatierung angesprochen werden. Lediglich CSS-Angaben, die direkt im eigentlichen Modul definiert sind, können auch weiterhin benutzt werden.

### 001 --Inhalt des Moduls--

Der Code sieht etwas »albern« aus, aber ich möchte damit deutlich machen, dass das Konstrukt `style="none"` wirklich keinen Code erzeugt, in dem das eigentliche Modul liegt.

- *Übersicht* - Ist identisch mit der Positionsvoransicht. Der Code dafür sieht folgendermaßen aus:

```
001 <div class="mod-preview">
002 <div class="mod-preview-info">POSITIONSNAME[outline]</div>
003 <div class="mod-preview-wrapper">
004 --Inhalt des Moduls--
    </div>
  </div>
```

- *Tabelle* - Das Modul selber wird in eine Tabelle eingebettet:

```
001 <table class="moduletable" cellspacing="0" cellpadding="0">
002 <tbody>
003 <tr>
004 <th>Mein Modul</th>
005 </tr>
006 <tr>
007 <td>
008 --Inhalt des Moduls--
009 </td>
010 </tr>
011 </tbody>
012 </table>
```

- *Abgerundet* - Wenn man Module mit abgerundeten Ecken erzeugen möchte, kann man mithilfe dieser Einstellung die Grundlagen dafür legen. Das Modul selber liegt in mehreren DIV-Elementen. Daraus kann man abgerundete Ecken machen. Allerdings wird solcherlei heutzutage mit CSS3 sehr einfach gemacht, und es braucht dafür nicht mehr diese verschachtelten DIVs.

Die Modulklassse ändert sich bei *rounded* (*abgerundet*) zu *module*:

```
001 <div class="module">
002 <div>
003 <div>
004 <div>
005 <h3>Modultitel</h3>
006 --Inhalt des Moduls--
007 </div>
008 </div>
009 </div>
010 </div>
```

- *XHTML* - Das ist quasi der Normalzustand. Das Modul ist in ein DIV-Element eingebettet. Mit dieser Einstellung werden auch Modultitel ausgegeben. Lässt man den Modus hingegen leer, wird der Titel nicht angezeigt.

```
001 <div class="moduletable">
002 <h3>Mein Modul</h3>
003 --Inhalt des Moduls--
004 </div>
```

## 8.10 Kollektion/Sammlung

Mit *Sammlung* sind Feldtypen gemeint, die ihrerseits auf bereits vorhandene Felder oder Formulare verweisen.

Die *X-Feldtypen* bieten zudem die Möglichkeit, vorhandene Felder/Gruppen in den Formularen zu vervielfältigen und somit gleiche Inhalte nach Bedarf mehrmals in einer Eingabemaske abzubilden.

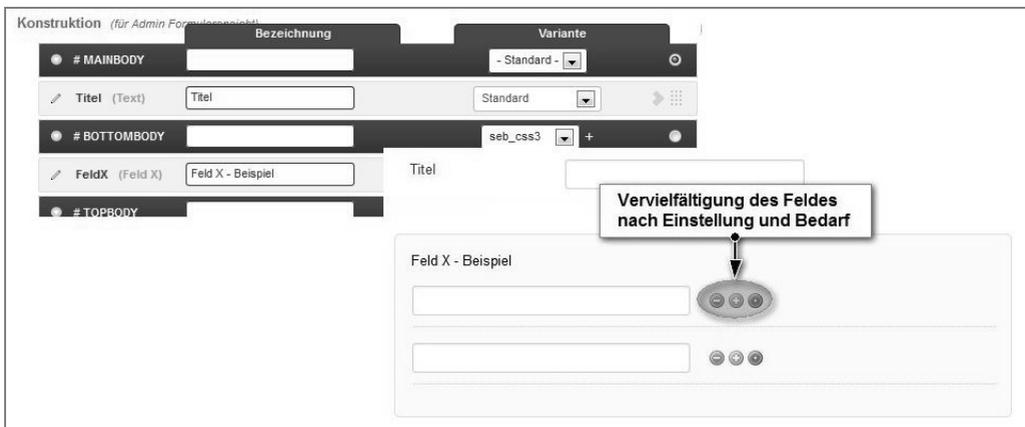
### 8.10.1 Gruppe

Haben Sie eine »Unter-App« und dazu ein Formular angelegt, können Sie mit diesem *Gruppe*-Feld auf dieses Formular verweisen und es so in die »Hauptform« einbinden. Beispiele für diesen Vorgang finden Sie in den *Article*-Inhaltstypen von SEBLOD. Sie können aber auch jedes andere existierende Formular einbinden. Sie müssen also nicht zuerst eine »Unter-App« erstellen.

Im erforderlichen Feld *Inhaltstyp (Formular)* wählen Sie das Formular aus, das als Gruppe eingebunden werden soll.

### 8.10.2 Feld X

Mit *Feld X* nehmen Sie Bezug auf ein bereits angelegtes Feld, indem Sie dessen Aliasnamen in *Feld* eintragen. Sie können danach dieses Feld beliebig oft in Ihrem Formular vervielfältigen.



**Bild 8.24:** Mit *Feld X* einzelne Felder beliebig oft ins Formular einfügen.

In *Standard* stellen Sie ein, wie oft das einzelne Feld bereits angezeigt wird, wenn Sie das Formular aufrufen.

**Hinweis**

Sie können hier auch die Zahl 0 eintragen. Dann wird das Feld nicht angezeigt, es gibt aber leider auch keine Möglichkeit, das Feld auszuwählen und zu vervielfältigen.

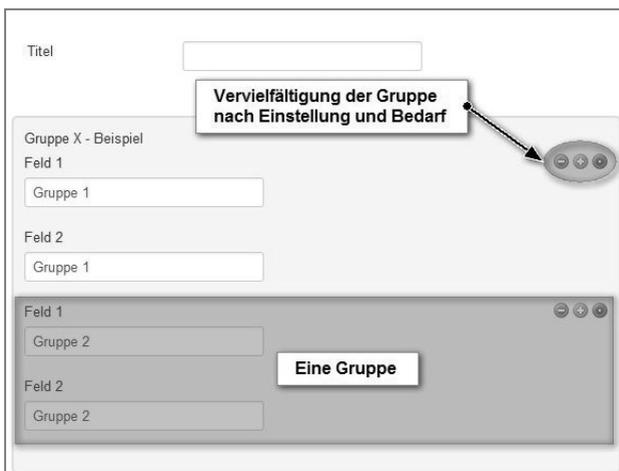
In *Maximum* geben Sie vor, wie oft das Feld maximal vervielfältigt werden kann, und in *Minimum*, wie oft es mindestens vorhanden sein muss. Stellen Sie unter *Minimum* eine höhere Zahl ein als unter *Standard*, wird der Beitrag auch gespeichert, wenn Sie weniger Felder eingebunden haben, als in *Minimum* vermerkt. Haben Sie jedoch mehr Felder als *Minimum* eingebunden, können Sie anschließend nur so viele Felder löschen, bis Sie das *Minimum* erreicht haben.

Mit *Hinzufügen*, *Löschen* und *Verschieben* stellen Sie ein, welche Buttons pro Gruppe angezeigt werden und ergo welche Aktionen der Benutzer in den Gruppen ausführen kann.

- *Hinzufügen* ist das grüne Pluszeichen, mit dem Sie ein neues Feld hinzufügen können.
- *Löschen* ist das rote Minuszeichen, mit dem Sie ein Feld löschen können.
- *Verschieben* ist der blaue Punkt. Halten Sie die linke Maustaste gedrückt auf dem blauen Punkt und bewegen Sie sie dann, können Sie die Reihenfolge der Felder ändern.

### 8.10.3 Gruppe X

Dieser Feldtyp funktioniert zunächst genauso wie *Gruppe* weiter oben. Der entscheidende Unterschied ist, dass Sie mit *Gruppe X* diese Gruppe in Ihrem Formular beliebig vervielfältigen können. Und damit verhält sich dieser Feldtyp nahezu identisch mit *Feld X* weiter oben. Überall, wo dort das Wort *Feld* steht, müssen Sie es mit *Gruppe* austauschen.



**Bild 8.25:** Mit *Gruppe X* eine Gruppe von Feldern beliebig oft in das Formular einbinden.

Unter *Ausrichtung* stellen Sie die Anzeige der Felder in der Gruppe ein:

- *Horizontal* zeigt die Felder nebeneinander an.
- *Vertikal* zeigt die Felder untereinander an.
- *Tabelle* verhält sich etwas anders, als bisher beschrieben. Sie erhalten keine Buttons, mit denen Sie neue Gruppen anlegen, löschen oder verschieben können. Stattdessen wird die eingestellte Standardanzahl in Tabellenform ausgegeben.



**Bild 8.26:** *Ausrichtung:*  
*Tabelle* verhält sich etwas anders.

## 8.11 Picker

Fertige Feldtypen, die es ermöglichen, auf sehr einfache Art Farbe oder ein Datum auszuwählen.

### 8.11.1 Farbauswahl

Etwas Farbe kann nie schaden. Mit diesem Feld können Sie auf einfache Art eine Farbe auswählen und in hexadezimaler Schreibweise speichern. Im Frontend wird dann auch dieser Farbwert ausgegeben.

### 8.11.2 Kalender

Im *Format* tragen Sie ein, wie das Datum angezeigt werden soll. Dafür stehen Ihnen einige Möglichkeiten zur Verfügung, wie Sie der nachfolgenden Tabelle entnehmen können.

Formatzeichen	Beschreibung	Beispiel für Rückgabewerte
<i>Tag</i>	---	---
<i>d</i>	Tag des Monats, zweistellig, mit führenden Nullen	01 bis 31

Formatzeichen	Beschreibung	Beispiel für Rückgabewerte
<i>D</i>	Wochentag, gekürzt auf drei Buchstaben	<i>Mon bis Sun</i>
<i>j</i>	Tag des Monats ohne führende Nullen	<i>1 bis 31</i>
<i>l</i> (kleines L)	ausgeschriebener Wochentag	<i>Sunday bis Saturday</i>
<i>N</i>	numerische Repräsentation des Wochentages gemäß ISO-8601 (in PHP 5.1.0 hinzugefügt)	<i>1</i> (für Montag) bis <i>7</i> (für Sonntag)
<i>S</i>	Anhang der englischen Aufzählung für einen Wochentag, zwei Zeichen	<i>st, nd, rd</i> oder <i>th</i> , zur Verwendung mit <i>j</i> empfohlen
<i>w</i>	numerischer Tag einer Woche	<i>0</i> (für Sonntag) bis <i>6</i> (für Samstag)
<i>z</i>	der Tag des Jahres (von 0 beginnend)	<i>0 bis 365</i>
<i>Woche</i>	---	---
<i>W</i>	ISO-8601, Wochennummer des Jahres, die Woche beginnt am Montag (hinzugefügt in PHP 4.1.0)	Beispiel: <i>42</i> (die 42. Woche im Jahr)
<i>Monat</i>	---	---
<i>F</i>	Monat als ganzes Wort, wie January oder March	<i>January bis December</i>
<i>m</i>	Monat als Zahl mit führenden Nullen	<i>01 bis 12</i>
<i>M</i>	Monatsname mit drei Buchstaben	<i>Jan bis Dec</i>
<i>n</i>	Monatszahl ohne führende Nullen	<i>1 bis 12</i>
<i>t</i>	Anzahl der Tage des angegebenen Monats	<i>28 bis 31</i>
<i>Jahr</i>	---	---
<i>L</i>	Schaltjahr oder nicht	<i>1</i> für ein Schaltjahr, ansonsten <i>0</i>

Formatzeichen	Beschreibung	Beispiel für Rückgabewerte
<i>o</i>	Jahreszahl gemäß ISO-8601; dies ergibt den gleichen Wert wie <i>Y</i> , außer wenn die ISO-Kalenderwoche ( <i>W</i> ) zum vorhergehenden oder nächsten Jahr gehört, wobei dann jenes Jahr verwendet wird (in PHP 5.1.0 hinzugefügt)	Beispiele: <i>1999</i> oder <i>2003</i>
<i>Y</i>	vierstellige Jahreszahl	Beispiele: <i>1999</i> oder <i>2003</i>
<i>y</i>	Jahreszahl, zweistellig	Beispiele: <i>99</i> oder <i>03</i>
<i>Uhrzeit</i>	---	---
<i>a</i>	kleingeschrieben: ante meridiem (Vormittag) und post meridiem (Nachmittag)	<i>am</i> oder <i>pm</i>
<i>A</i>	großgeschrieben: Ante Meridiem (Vormittag) und Post Meridiem (Nachmittag)	<i>AM</i> oder <i>PM</i>
<i>B</i>	Swatch-Internetzeit	<i>000</i> bis <i>999</i>
<i>g</i>	Stunde im 12-Stunden-Format ohne führende Nullen	<i>1</i> bis <i>12</i>
<i>G</i>	Stunde im 24-Stunden-Format ohne führende Nullen	<i>0</i> bis <i>23</i>
<i>h</i>	Stunde im 12-Stunden-Format mit führenden Nullen	<i>01</i> bis <i>12</i>
<i>H</i>	Stunde im 24-Stunden-Format mit führenden Nullen	<i>00</i> bis <i>23</i>
<i>i</i>	Minuten mit führenden Nullen	<i>00</i> bis <i>59</i>
<i>s</i>	Sekunden mit führenden Nullen	<i>00</i> bis <i>59</i>
<i>u</i>	Mikrosekunden (hinzugefügt in PHP 5.2.2)	Beispiel: <i>654321</i>
<i>Zeitzone</i>	---	---

Formatzeichen	Beschreibung	Beispiel für Rückgabewerte
<i>E</i>	Zeitzonebezeichner (hinzugefügt in PHP 5.1.0)	Beispiele: <i>UTC</i> , <i>GMT</i> , <i>Atlantic/Azores</i>
<i>I</i> (großes i)	fällt ein Datum in die Sommerzeit	<i>I</i> bei Sommerzeit, ansonsten <i>0</i>
<i>O</i>	Zeitunterschied zur Greenwich Mean Time (GMT) in Stunden	Beispiel: <i>+0200</i>
<i>P</i>	Zeitunterschied zur Greenwich Mean Time (GMT) in Stunden mit Doppelpunkt zwischen Stunden und Minuten (hinzugefügt in PHP 5.1.3)	Beispiel: <i>+02:00</i>
<i>T</i>	Abkürzung der Zeitzone	Beispiele: <i>EST</i> , <i>MDT</i> ...
<i>Z</i>	Offset der Zeitzone in Sekunden; der Offset für Zeitzone westlich von UTC ist immer negativ und für Zeitzone östlich von UTC immer positiv	<i>-43200</i> bis <i>50400</i>
<i>Vollständige(s) Datum/Uhrzeit</i>	---	---
<i>c</i>	ISO 8601, Datum (hinzugefügt in PHP 5)	2004-02-12T15:19:21+00:00
<i>r</i>	Gemäß <i>RFC 2822</i> formatiertes Datum	Beispiel: <i>Thu, 21 Dec 2000 16:01:07 +0200</i>
<i>U</i>	Sekunden seit Beginn der UNIX-Epoche (January 1 1970 00:00:00 GMT)	siehe auch <i>time()</i>

Quelle: <http://php.net/manual/de/function.date.php>

Um also beispielsweise den »5.12.2013 - 16:10« abzubilden, tragen Sie dieses Format ein:

001 N.m.Y - H:i

*Speicherformat* schreibt unterschiedliche Werte in die Datenbank. Bei *Datetime* wird die Auswahl folgendermaßen gespeichert:

```
001 2013-07-19 10:20:00
```

Bei der Einstellung *Zeitstempel* wird eine Integerzahl gespeichert, aus der sich Datum und Uhrzeit berechnen lassen. Das gleiche Datum wie oben sieht dann so aus:

```
001 1374229200
```

*Datetime* speichert den Wert so ab, wie Sie ihn schreiben, und zeigt ihn auch so wieder an. Beim Format *Zeitstempel* (im Programmierjargon *Timestamp* genannt) berechnen die Datenbank und der Webserver das Datumsformat aus dem Integerwert. Das könnte manchmal auch ein anderes Ergebnis liefern, so ist beispielsweise bei meiner lokalen Installation das »heutige Datum« in die Vergangenheit verrutscht.

Die *Zeitspanne* gibt Ihnen die Möglichkeit, die Auswahlmöglichkeiten im Kalender einzuschränken.

- *Alle*: Sie können jedes beliebige Datum auswählen.
- *Vergangenheit* bedeutet, dass Sie nur ein Datum aus der Vergangenheit, also ab gestern, auswählen können.
- *Vergangenheit & Heute* bedeutet, dass Sie jedes Datum aus der Vergangenheit plus das heutige Datum auswählen können.
- *Heute & Zukunft*: Jedes Datum in der Zukunft ab heute ist auswählbar.
- *Zukunft*: Jedes Datum ab morgen können Sie auswählen.

Mit *Motiv* können Sie das Aussehen des Kalenders ein wenig verändern: *Gold*, *Stahl* und im Stil von *Win2000*.

Sie können sich im Kalender die *Wochennummern* anzeigen lassen und bei der Uhrzeit einstellen, ob Sie die 24-Stunden-Ansicht oder die 12-Stunden-Ansicht wünschen und wo unter dem Kalender das angezeigt werden soll: *Rechts* oder *Links*.



**Bild 8.27:** Die Uhrzeit ganz unten in Ausrichtung *Rechts*.

## 8.12 Textarea

Anders als bei einem Textfeld haben Sie hier die Möglichkeit, Text über mehrere Zeilen einzugeben.

### 8.12.1 Textbereich

Der Feldtyp *Textbereich* stellt Ihnen einen Bereich zur freien Eingabe von unformatiertem Text zur Verfügung. Die Größe dieses Textbereichs geben Sie in Form von *Spalten* und *Reihen* an.



Bild 8.28: Textarea.

Mit *Mindestlänge* legen Sie fest, wie viele Zeichen mindestens eingetragen werden müssen. Die *Maximallänge* gibt an, wie viele Zeichen maximal eingetragen werden können. Sie können dann auch nur bis zu dieser *Maximallänge* Zeichen eingeben.

Im Abschnitt *Prozesse* können Sie einige wenige Formatangaben machen. Unter *Neue Zeilen* geben Sie an, wie sich das Textfeld nach Drücken der `[Enter]`-Taste verhalten soll.

- *Zeilenumbrüche (BR)* fügt am Ende der Zeile das HTML-Element `<br>` ein.
- *Zeilenumbrüche (BR) in einem Absatz (P)* fügt nach jedem Zeilenumbruch auch das HTML-Element `<br>` ein und umschließt den ganzen Block mit einem `<p>...</p>`.
- *Absätze (P)* fasst den ganzen Absatz mit einem `p`-Element ein (`<p>...</p>`).

Sehen können Sie das alles nur in der Datenbank oder dann in der Ausgabe im Frontend.

- Mit *Lösche Leerzeilen* werden Zeilen, in denen nichts steht, entfernt.

### 8.12.2 WYSIWYG-Editor

Mit dem WYSIWYG<sup>26</sup>-Editor können Sie freien Text über einen Editor eingeben. Dabei können Sie nicht nur einen installierten WYSIWYG-Editor einbinden, sondern auch beispielsweise den Code-Editor, der sich zum Schreiben von Programmierungen anbietet.

Lassen Sie den *Editor auf Standard verwenden* stehen, wird der genommen, den Sie in der Joomla!-Konfiguration voreingestellt haben.

<sup>26</sup> WYSIWYG: What You See Is What You Get – Was du siehst, ist, was du bekommst.

Interessant ist der *Standard Wert*. Denn hier wird Ihnen der Standardeditor angezeigt, und Sie haben alle Freiheiten, den Standardtext zu gestalten.

Beim *Ort* geben Sie an, ob der Editor direkt angezeigt wird oder in einem modalen Fenster. Im letzteren Fall erhalten Sie einen Button mit der Aufschrift *Editor*.

Die *Breite und Höhe* des Editorfeldes können Sie nach eigenem Gusto einstellen. 400 x 280 zeigt Ihnen den Editor in 400 px Breite und 280 px Höhe an. 50 % x 280 gibt Ihnen den Editor in 50 % Breite und 280 px Höhe aus. Sie können also die Maßangaben variieren und auch mischen.

## 8.13 Upload

Darüber können Elemente hochgeladen werden. Grundsätzlich ist das ein Bild oder eine Datei. Da Sie aber in der Vergabe der Mediensuffixe sehr flexibel sind, können Sie auch Mediadateien hochladen.

### Auf Umlaute verzichten

Es gibt offenbar Schwierigkeiten in SEBLOD mit Dateinamen, die Umlaute enthalten. Die Dateien selber werden wohl hochgeladen, aber die Verlinkungen zur Datei gehen später im Frontend verloren. Der Grund ist, dass SEBLOD Umlaute »falsch« umwandelt (aus dem »ü« wird beispielsweise »Ã¼«). Verzichten Sie daher auf Umlaute. Auch sollten Sie generell Dateinamen kleinschreiben und ohne Leerzeichen. Statt Leerzeichen können Sie gut mit einem Unterstrich arbeiten, beispielsweise *baeren\_im\_winter.pdf*.

### 8.13.1 Datei

Sie können eine Datei hochladen bzw. ein Upload-Feld zur Verfügung stellen. Im *Ordner* tun Sie kund, wohin die Datei gespeichert werden soll. Dabei ist es wichtig, darauf zu achten, dass am Ende immer ein Slash steht. Vorgegeben ist das *images*-Verzeichnis, Sie können aber auch beispielsweise in das *media*-Verzeichnis oder ein eigens dafür angelegtes Verzeichnis speichern.

Das *Speicherformat* ist von der Wortwahl etwas verwirrend, weil hier kein Format ausgewählt, sondern lediglich das Aussehen im Frontend beeinflusst wird. Wählen Sie *vollständiger Pfad* aus, wird im Frontend der gesamte Pfad mit dem Dateinamen angezeigt. Wählen Sie hingegen *Dateiname* aus, wird nur der Dateiname angezeigt. Beides ist im Übrigen ein Verweis, den Sie anklicken, um die Datei dann downloaden zu können. Setzen Sie darunter *1 Ordner/Benutzer* auf *Ja*, kann der *vollständige Pfad* nicht mehr ausgewählt werden. Es wird dann nur der Dateiname ausgegeben. Das ist ein kleines Sicherheitsfeature. Die Benutzer-ID ist nicht für die Öffentlichkeit bestimmt.

1 *Ordner/Inhalt* legt für jede Datei ein neues Verzeichnis an. Dafür wird die Beitrags-ID genommen, die entsteht, wenn ein neuer Beitrag angelegt wird. Angenommen, Ihr Beitrag hat die ID 58, dann wird ein Verzeichnis im vorgegebenen Ordner – standardmäßig *images* – erstellt: *images/58/dateiname.pdf*. Das zweite Feld, das die Zahl 0755 vorgegeben hat, sind die Verzeichnisrechte für diesen neuen Ordner, auch *chmod*<sup>27</sup> genannt.

1 *Ordner/Benutzer* bedeutet, dass Sie für jeden Benutzer, der eine Datei hochlädt, ein Unterverzeichnis erstellen, wobei die Benutzer-ID zugrunde gelegt wird. Angenommen, Sie haben einen Benutzer mit der ID 976, dann wird das Verzeichnis *images/976/dateiname.zip* erstellt (.zip und .pdf sind natürlich nur Beispiele).

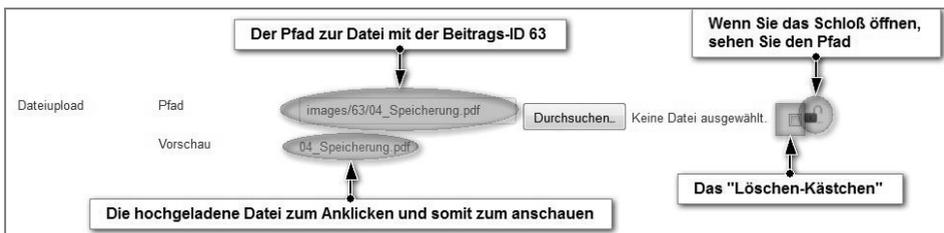
Wählen Sie beides aus, also einen Ordner sowohl für den Benutzer als auch für den Inhalt, erhalten Sie auch eine Kombination aus diesen beiden IDs, zunächst die Benutzer-ID und anschließend die Beitrags-ID: *images/976/58/Dateiname.pdf*.

Bei den *erlaubten Erweiterungen* nehmen Sie Bezug auf die Einstellungen in den Optionen bei den einzelnen Auswahlmöglichkeiten. Sie können aber auch eine benutzerdefinierte Liste erstellen und sind somit völlig unabhängig von den Voreinstellungen.

Um zu verhindern, dass jemand eine zu große Datei hochlädt, können Sie eine *Maximalgröße* vorgeben, je nach Bedarf in Byte, Kilobyte oder Megabyte.

Einmal hochgeladen, kann die hochgeladene Datei im Formular selber angezeigt werden, wahlweise als Dateiname, der als Verweis anklickbar ist, als Icon oder nur als Dateiname ohne Verweis.

Mit dem *Zeige Löschen*-Kästchen können Sie sich eine Checkbox anzeigen lassen. Setzen Sie dort einen Haken und speichern den Beitrag, wird die Datei gelöscht.



**Bild 8.29:** Ein Datei-Upload-Feld.

Wenn Sie *Zeige spezifischen Pfad* auf *Zeige* gesetzt haben, können Sie im Eingabeformular ein kleines gelbes Schloss anklicken und öffnen. Dann wird Ihnen der Pfad zur Datei angezeigt. Den *Path* können Sie eindeutschen oder mit einem anderen Wort

<sup>27</sup> *chmod* = change mode. Die drei letzten Zahlen sind für den *Besitzer*, die *Gruppe* und für *andere* zuständig. 755 bedeutet, dass der Besitzer alle Rechte an dem Ordner hat, die Gruppe und andere nur lesen und ausführen, jedoch nicht schreiben dürfen. Weitere Informationen gibt es unter anderem hier: <http://de.selfhtml.org/helferlein/chmod.htm>

belegen. Die *Größe* bezieht sich auch hier auf das Eingabefeld. Der *erweiterte Modus* und die Auswahl *Ja (JSON-formatierter Wert)* erlauben es Ihnen, einen eigenen Anzeigenamen für Ihre Datei zu vergeben. In meinem kleinen Beispiel habe ich zunächst bei *Zeige Titel* den Wert »Zeige« ausgewählt und in das freie Feld *Name für die Datei:* hineingeschrieben.

Im Adminformular habe ich als Dateianzeigename *Aus dem Buch* gewählt. In der Vorschau ist auch genau dieser Satz zu sehen, und beim Anklicken erscheint die zuvor hochgeladene Datei. Im Frontend wird auch dieser Name ausgegeben und kann so heruntergeladen werden.

Pfad	<input type="text" value="images/972/65/04_Speicherung.pdf"/>	<input type="button" value="Durchsuchen..."/>
Name für die Datei:	<input type="text" value="Aus dem Buch"/>	
Vorschau	Aus dem Buch	

**Bild 8.30:**

Eigener Anzeigename im Frontend für die Datei.

## 8.13.2 Image

Eine vollständige Beschreibung mit Praxiseinsatz finden Sie in Kapitel 3.1.2 »Direkt zu den Feldern« (Feldtyp Bild).

## 8.14 Video

Momentan steht lediglich ein Feldtyp für ein YouTube-Video zur Verfügung. Wer keine Angst vor dem Programmieren hat, wird aber sicherlich recht einfach das zugehörige Plug-in auch für andere Videosysteme anpassen können.

### 8.14.1 YouTube

Sie können Videos von YouTube einfügen. Hierfür müssen Sie beim *Standard Wert* den Videocode angeben. Der Videocode ist der letzte Teil der YouTube-URL.

001 <http://youtu.be/YeIPCskMM2M>

Wenn Sie die ganze URL eintragen, wird kein Video angezeigt.



**Bild 8.31:** Videos von YouTube einbinden.

Interessant ist, dass das Video vom *Standard Wert* bereits im Adminformular angezeigt wird, sofern Sie *Zeige Vorschau* auf *Zeige* einstellen.

In *Breite/Höhe* geben Sie an, in welcher Größe das Video dargestellt werden soll, und in *Markup* wählen Sie die Art, wie das Video in Ihrer Site eingebettet wird. *Iframe* bettet die Datei, das Video, in Ihrer Website ein. Mit *embed* benötigen Sie weniger Code (ein Grund, diese Technik zu bevorzugen), brauchen aber ein Plug-in zur Anzeige. Moderne Browser haben in der Regel keine Probleme mit der Darstellung eines Videos, das mit *embed* eingebunden ist. Sollten Sie für das Intranet entwickeln und vielleicht Probleme mit älteren Browsern haben, probieren Sie die Einbettung mit *Iframe*.

## 8.15 # Core

### 8.15.1 42

Bei diesem Feld handelt es sich um ein Überbleibsel aus Joomla! 1.5. Damals befand sich der Administrator stets unter der Benutzer-ID 42. Aber natürlich ist das Feld auch in heutiger Zeit zu benutzen. Es handelt sich um ein Feld für Programmierer und auch dafür, ein wenig zu experimentieren – sofern Sie der Programmierung mit PHP ein wenig mächtig sind.

Hier ein Beispiel direkt aus dem SEBLOD-Forum:

Fügen Sie in dem Bereich *Formular erstellen (PrepareForm)* diese Zeile Code ein:

```
001 $form = '<input type="text" id="toto" name="toto" value="plop" />';
```

Wenn Sie nun im Backend einen neuen Beitrag schreiben, erhalten Sie ein Eingabefeld mit dem *Standard Wert* *plop*.

The image shows a screenshot of a Content Construction Kit (CCK) form. On the left, there is a label 'Core 42'. To its right is a text input field containing the text 'plop'. The entire form is enclosed in a thin black border.

**Bild 8.32:** Das CCK mit dem *plop*.

Sie können sich so also mittels Code einen Formulartyp zusammenbauen. Allerdings haben Sie hier das Problem, dass nichts gespeichert wird. Sie müssen nun entsprechende Speicherroutinen programmieren, weshalb diese Methode zwar generell machbar ist, aber letztlich die vielen Vorteile, die SEBLOD bietet, an dieser Stelle nicht zur Verfügung stellt. Haben Sie jedoch entsprechenden Code zur Verfügung oder scheuen die Arbeit nicht, dann haben Sie natürlich die Möglichkeit, Ihr selbst programmiertes Formular, den Inhaltstyp und sogar den Onlineshop mit den ganzen Möglichkeiten, die SEBLOD bietet, zu verbinden.

Beachten Sie die unterschiedlichen Eingabebereiche für den Inhalt, das Formular und den Speicherbereich – *PrepareContent*, *PrepareForm* und *PrepareStore*.



# 9

## Formulare & Inhaltstypen

### 9.1 Konfiguration: Admin- und Seitenformular



Bild 9.1: Konfiguration im Inhaltstyp.

#### 9.1.1 Konfiguration

Sie können Ihren Autoren nach dem Absenden eines Beitrags eine Nachricht anzeigen lassen. Im *Nachrichtenstil* geben Sie vor, in welcher Farbe und mit welcher Überschrift die Nachricht ausgegeben wird. Bei mir war zwischen *Mitteilung* und *Nachricht* kein farblicher Unterschied - beide waren in einem Brauntönen gehalten. Lediglich die Überschrift hat sich zwischen *Mitteilung* und *Nachricht* geändert.

Ein *Fehler* wird auch als Überschrift ausgegeben, im bekannten Joomla!-Rot für Fehleranzeigen. Den Text, der ausgegeben wird, schreiben Sie in das Feld *Nachricht (Danke)*. Als Nachrichtenstil *Fehler* auszuwählen hat nur Einfluss auf das Aussehen, jedoch nichts mit einem eventuell auftretenden Fehler zu tun. Dieser Text wird

einfach nach dem Speichern des Beitrags angezeigt. *Kein* zeigt keinen Text an und entspricht dem Standardverhalten.

Unter *Weiterleiten* legen Sie fest, wohin der Autor nach dem Speichern eines Beitrags geleitet wird. Diese Einstellung gibt es nur bei der Auswahl *Seitenformular*, was aber logisch ist, da Sie im Backend nicht zu einer anderen Seite geleitet werden können.

- *Inhalt*: Sie erhalten bei dieser Auswahl eine Auswahlbox mit Ihren Menüs.
- *Aktuell*: Sie bleiben auf der Seite und können einen neuen Beitrag schreiben. Haben Sie einen Beitrag, dessen Eingabe über mehrere Seiten reicht, springen Sie zur ersten Eingabeseite.
- *Aktuell (ganz)*: Damit bleiben Sie auf der Seite und können gleich einen neuen Beitrag schreiben. Die Formularfelder werden allerdings nicht geleert. Wenn Sie den Inhalt über mehrere Seiten eingeben, dann bleiben Sie auf der letzten Seite, wo gespeichert wurde.
- *Formular*: Verhält sich identisch mit *Aktuell*.
- *Formular (Ausgabe)*: Auch hier bleiben Sie auf der Seite, aber das Formular wird nicht geleert. Die Inhalte in den Formularfeldern bleiben also stehen. Bei einem Formular über mehrere Seiten springen Sie zum Anfang.
- *Url*: Sie müssen eine komplette URL angeben. Wenn Sie den Benutzer zur Startseite geleiten möchten, müssen Sie die URL in dieser Art eintragen:

```
001 http://localhost/j31/index.php
```

## 9.1.2 Datensicherheit

An dieser Stelle muss ich leider passen. Ich habe nicht herausfinden können, wo der tiefere Sinn dieses Feldes liegt.

## 9.1.3 Allgemein

Diese Auswahl erhalten Sie in der *Konfiguration* nur in der *Seitenformular-Ansicht*.

Wenn Sie einen neuen Menüverweis vom Typ SEBLOD anlegen und Ihr Formular dort auswählen, finden Sie diese Einstellungen im Menü unter dem Tab *Grundlegende Optionen (CCK)* wieder. Sie können im Formular direkt entsprechende Angaben machen und diese Angaben auch im Menüverweis überschreiben. Somit haben Sie mehrere Möglichkeiten der Gestaltung, neben der globalen Eingabe eine individuelle bei den Menüs.

### 9.1.4 Kein Zugang

Ähnlich wie weiter oben unter »Konfiguration« können Sie hier auch eine Nachricht vorgeben, diesmal allerdings eine Nachricht, wenn der Benutzer keinen Zugriff auf den Inhalt hat. Zu Testzwecken habe ich einen Menüverweis zum Einreichen eines Beitrags erstellt und diesen für die Öffentlichkeit sichtbar gemacht. Da Beiträge nur Autoren einreichen dürfen, gibt es beim Aufrufen dieses Links als Besucher der Seite eine Fehlermeldung, die besagt, dass er nicht berechtigt ist, diesen Inhalt einzusehen.

Sie können hier für solche Fälle und für diesen Inhaltstyp einen eigenen Text kreieren. Der *Nachrichtenstil* ist identisch mit der Beschreibung weiter oben. Unter *Aktion* gibt es die Möglichkeit, eine *Weiterleitung* für solche Fälle zu erstellen. Standardmäßig hat SEBLOD hier einen Link vorgegeben, der das Log-in unter dem Hinweistext anzeigt.

```
001 index.php?option=com_users&view=login
```

### 9.1.5 Variante

Mit dem Zähler stellen Sie ein, über wie viele Seiten Ihr Eingabeformular reicht.

### 9.1.6 Validation

Mit *Validation* sind die Hinweistexte gemeint, dass es sich um ein Pflichtfeld handelt oder die Überprüfung ergeben hat, dass der Inhalt in Ordnung ist oder eben nicht. Wo sie angezeigt werden, können Sie in der *Position* vorgeben, es bezieht sich stets auf das Feld, in dem der Validationstext ausgegeben wird. Wie ich an anderer Stelle bereits schrieb, hat sich mir das *Blättern* in diesem Zusammenhang noch nicht erschlossen. Wenn Sie es herausfinden, freue ich mich, von Ihnen zu hören. Die *Farbe* bezieht sich auf die Schriftfarbe, und die *Hintergrundfarbe* bezieht sich – auf die Hintergrundfarbe der Validationsfelder.

#### **Adminformular und Seitenformular**

Die Konfigurationen für das Adminformular und für das Seitenformular sehen zwar fast gleich aus, sind aber gesondert zu betrachten. Sie schreiben stets eine Konfiguration nur für eines der beiden Formulare. Die Konfiguration für das Adminformular wird also nicht automatisch für das Seitenformular übernommen und umgekehrt.

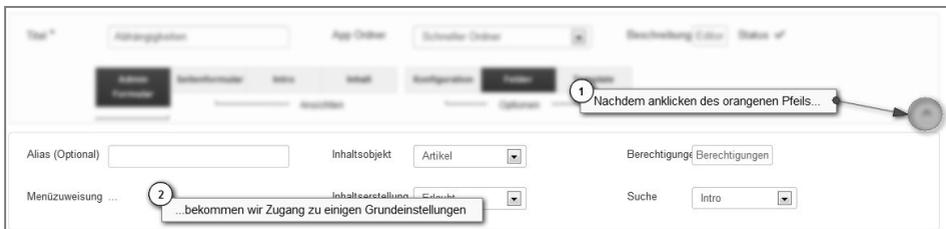
## 9.2 Konfiguration: Intro und Inhalt

Es gibt eine extra Konfiguration für das Intro und den Inhalt. Die Konfiguration bezieht sich auf das Aussehen der URL und den Titel. Leider haben diverse Versuche bei mir rein gar nichts bewirkt. Bei der SEF-URL könnte das daran liegen, dass Joomla! ab der Version 3.0 offenbar das Verhalten hier etwas verändert hat, sodass SEBLOD mit der

veränderten Struktur noch nicht klarkommt. Es kann aber auch gut sein, dass in einer späteren Version von SEBLOD nachgebessert wird.

### 9.3 Grundeinstellungen

Wenn Sie ganz oben rechts auf den orangefarbenen Pfeil klicken, klappt ein Bereich heraus, der einige Grundeinstellungen für den Inhaltstyp beinhaltet.



**Bild 9.2:** Grundeinstellungen im Inhaltstyp.

Neben dem *Alias*, den SEBLOD/Joomla! intern benutzt, können Sie auch den Inhaltstyp über das *Quick Menu* direkt einem Menü zuordnen oder einen neuen Menüverweis anlegen. Haben Sie mehrere Menüs angelegt, werden sie hier auch aufgelistet. Sie können einem Ihrer Menüs diesen Inhaltstyp hinzufügen. Es wird dann in Ihrem ausgewählten Menü ein neuer Menüverweis mit dem Namen des Inhaltstyps erstellt. Haben Sie keine eigenen Menüs angelegt, haben Sie standardmäßig mindestens das *Main Menu* zur Auswahl.

Alternativ können Sie auch einen bereits vorhandenen Menüverweis auswählen. Joomla! legt dann einen neuen Untermenüverweis unter dem ausgewählten Menüverweis mit dem Namen des Inhaltstyps an.

Unter *Inhaltsobjekt* geben Sie an, um was für einen Inhaltstyp es sich genau handelt. Insbesondere wenn Sie eine leere Datei erstellen, sollten Sie hier angeben, um was es sich eigentlich handelt. Unter der Überschrift *Joomla* finden Sie hier die bereits bekannten Inhaltsobjekte wieder: *Artikel*, *Benutzer*, *Benutzergruppe* und *Kategorie*. Außerdem können Sie *Frei* und *Keine* auswählen.

*Frei* führt dazu, dass Ihr Inhaltstyp nicht angezeigt wird, wenn Sie einen neuen Beitrag etc. anlegen möchten. Ein mögliches Anwendungsszenario ist, dass Sie einen Inhaltstyp/ein Formular für eine Gruppe erstellen möchten. Dann sollte man das Formular auch nur zusammen mit dem ganzen Inhaltstyp sehen können. *Kein* ordnet kein *Inhaltsobjekt* zu. Anders als *Frei* können Sie mit diesem Inhaltstyp dann auch nicht weiterarbeiten.

Mit der *Inhaltserstellung* erlauben oder verweigern Sie grundsätzlich. Je nachdem, was Sie an Möglichkeiten vorgeben (*Admin-* und oder *Seitenformular*), kann editiert, neu

angelegt oder bearbeitet werden oder eben nicht, wenn Sie auf *verweigert* stellen. Sie können hier aber auch differenzieren, indem Sie die Erlaubnis nur für das Backend (*Nur Administrator*) oder nur für das Frontend (*nur Seite*) geben.

Die *Berechtigungen* regeln, wer welche Editier- und Erstellungsrechte auf den Inhaltstyp hat, wobei ein Teil der Berechtigungen unisono mit den Joomla!-Rechten einhergeht.

- *Erstellen* erlaubt generell das Anlegen neuer Inhalte.
- *Löschen* erlaubt es, alle erstellten Inhalte in den Papierkorb zu verschieben. Das endgültige Löschen kann nur über das Backend erfolgen.
- *Löschen eigener Inhalte* erlaubt nur das Löschen des selbst erstellten Inhalts, nicht jedoch fremder Inhalte.
- *Bearbeiten* regelt, dass grundsätzlich jeder Inhalt editiert werden kann.
- *Eigene Inhalte bearbeiten* erlaubt ausschließlich das Editieren eigener Inhalte.

Interessant sind die drei Rechte, die es sonst nicht in Joomla! gibt und die Anzahl der möglichen Beiträge nach verschiedenen Vorgaben regelt.

- *Erstelle (Max. Untergeordnete)*: Sie können durch das Eintragen einer Zahl vorgeben, wie viele Inhalte innerhalb eines Kategoriestrangs maximal verfasst werden dürfen, und zwar unabhängig vom Autor. Ansonsten gilt hier das Gleiche, wie im nächsten Absatz beschrieben.
- *Erstelle (Max. Untergeordnete/Autor)*: Geben Sie hier eine Zahl an, legen Sie fest, wie viele Inhalte in einem Kategoriestrang insgesamt von einem Benutzer verfasst werden dürfen. Mit einem »Kategoriestrang« meine ich sowohl die Hauptkategorie als auch zugehörige Unterkategorien.

Angenommen, Sie tragen hier die Zahl 2 ein und haben nachfolgende Kategorien:

- Schwimmen
  - Tauchen
- Golfspielen
  - Minigolf
  - Rasengolf

Dann könnten Sie zwei Beiträge innerhalb des Kategoriestrangs Schwimmen und Tauchen schreiben und zwei Beiträge im Kategoriestrang Golfspielen und den zugehörigen Unterkategorien Minigolf und Rasengolf.

- *Erstelle (Max./Autor)*: Tragen Sie hier eine Zahl ein, kann der Benutzer, der der jeweiligen Benutzergruppe zugeordnet ist, nur diese Anzahl von Beiträgen insgesamt verfassen. Tragen Sie beispielsweise die Zahl 5 hier ein, dann kann der Autor nur maximal fünf Beiträge schreiben. Ab dem sechsten Beitrag erhält er den Hinweis »Sie dürfen nicht mehr Elemente dieser Art hinzufügen.« - Unglücklicherweise aber erst, wenn er versucht zu speichern.

Wenn Sie Änderungen in der Rechteverwaltung vornehmen, müssen Sie erst den Inhaltstyp speichern, bevor Sie die Änderungen sehen.

Bei der *Suchindexierung* geben Sie an, ob die Suche das *Intro* oder den *Inhalt* indexieren soll. So können Sie auch gezielt Ihre Formulare entsprechend anlegen.

## 9.4 Konstruktionsbereich

Im *Konstruktionsbereich* befinden sich die Header für die einzelnen Positionen. Hier können Sie Einstellungen vornehmen, die sich stets auf die ganze Position beziehen und somit Einfluss auf die Formatierung der einzelnen Felder nehmen.

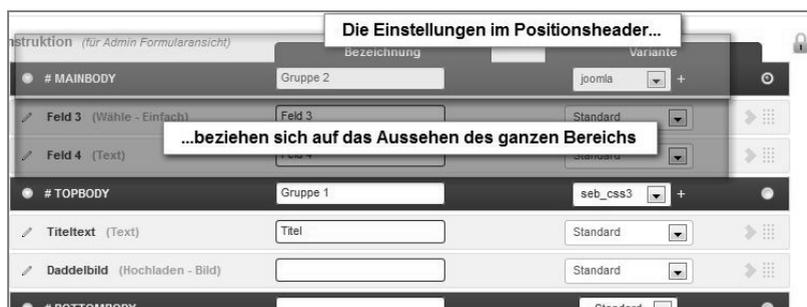


Bild 9.3: Einstellungen im Positionsheader wirken sich auf alle zugeordneten Felder aus.

Innerhalb der Positionsangaben können Sie eine *Bezeichnung* angeben und so die einzelnen Positionen optisch gruppieren.

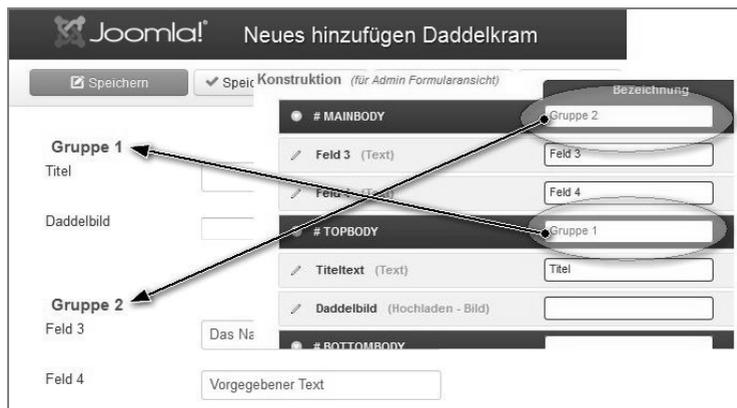


Bild 9.4: Einfache Gruppierungen.

Auf der rechten Seite unter *Variante* haben Sie die Möglichkeit, die Ausgabe der einzelnen Positionen etwas detaillierter zu gestalten.

- *Standard* gibt Ihnen die einzelnen Felder untereinander aus.
- Wählen Sie die Einstellung *Leer*, wird diese Position nicht ausgegeben. Somit können Sie also auch einen ganzen Bereich verstecken.
- Die Einstellung *Kein* sorgt dafür, dass die Felder nicht sauber untereinander, sondern unkontrolliert nebeneinander ausgegeben werden. *Kein* ignoriert CSS-Einstellungen.

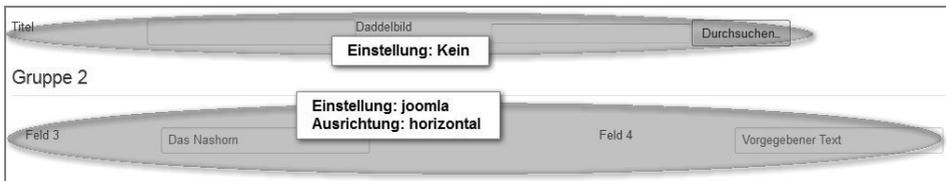


Bild 9.5: *Kein* ignoriert CSS-Angaben.

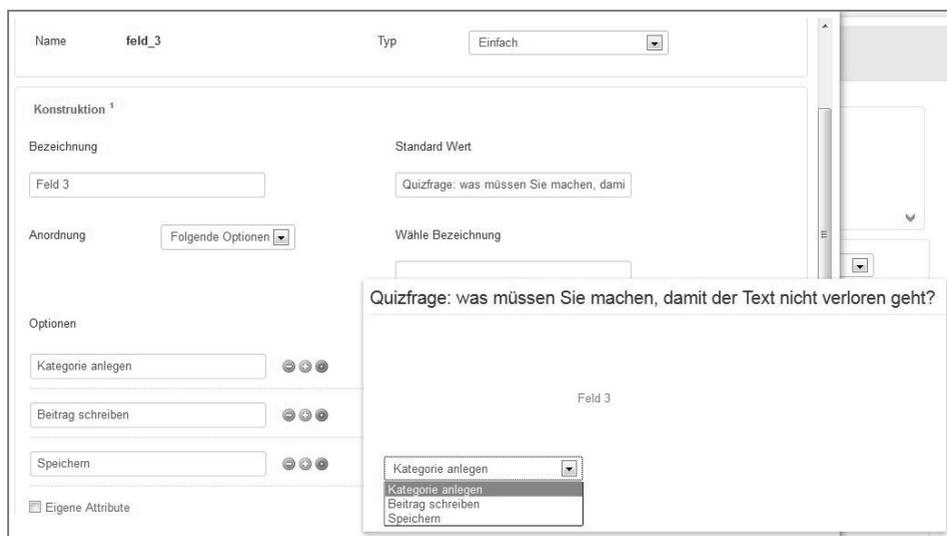
- Mit *joomla* können Sie differenziertere Einstellungen vornehmen und den Bereich optisch ein wenig gestalten. Dazu klicken Sie das kleine Pluszeichen an, dass neben *joomla* erscheint.
- Unter *Positionslegende* geben Sie den Alias des *Feldnamens* an, dessen Inhalt als Gruppenname benutzt werden soll. In der Abbildung setze ich einen Bezug auf *Feld 3*, wo ich den *Standard Wert Das Nashorn* eingegeben habe, was nun als Gruppenbezeichnung verwendet wird.



Bild 9.6: *joomla*-Einstellungen und ihre Auswirkungen.

Sie können dafür natürlich jedes Feld benutzen, das einen *Standard Wert* enthält. Hier ein einfaches Beispiel, um ein Quiz mit einem einfachen Auswahlfeld zu erzeugen:

Der Bereich *Felder* bezieht sich auf die reinen Felder ohne ihre Bezeichnung. Zum einen können Sie auswählen, ob die Felder untereinander (*Vertikal*) oder nebeneinander (*Horizontal*) ausgegeben werden. Die *Breite* bezieht sich auf den Platz der einzelnen Felder. Die Angabe kann wahlweise in Pixeln oder in Prozent angegeben werden. Prozent bezieht sich stets auf den sichtbaren Bereich. Da Sie nicht wissen, in welcher Größe der Browser des Benutzers eingestellt ist, kann eine Pixelangabe für sauberere Ergebnisse sorgen, ist aber nicht dynamisch.



**Bild 9.7:** Hätten Sie's gewusst?

In *Zeige Bezeichnung* geben Sie entweder explizit an, ob in diesem Positionsbereich die Feldbezeichnungen angezeigt werden oder nicht oder ob die Standardeinstellungen, die Sie bei den Template-Optionen eingestellt haben, gelten. Wenn Sie die Bezeichnungen anzeigen wollen, können Sie bestimmen, ob die *Bezeichnungsposition über* oder *links* neben den Feldern sein soll.

Der Bereich *Feldbezeichnung* kümmert sich dann auch um die Bezeichnungen, sofern Sie selbige auf *anzeigen* eingestellt haben. *Ausrichtung Links* oder *Rechts* bezieht sich auf die Bezeichnung selbst.

Im weiter oben gezeigten Bild 9.6 mit der Unterschrift »*joomla*-Einstellungen und ihre Auswirkungen.« sehen Sie eine solche *Rechts-Ausrichtung*.

Diese Ausrichtung ist natürlich auch von der *Breite* bestimmt, die Sie auch wieder in Prozent oder Pixeln angeben können. Bei einer horizontalen Ausrichtung ist dieser Wert interessanter, da er auch den Abstand zum Feld mitsteuert. Der *Rahmenabstand*

ist zunächst etwas verwirrend, weil ja eigentlich keine Rahmen vorhanden sind. Tatsächlich handelt es sich um den Abstand zwischen Bezeichnung und Feld.

## 9.5 CSS-Formatierung

SEBLOD liefert hier eine eigene CSS-Formatierung mit, mit der Sie andere Möglichkeiten der Gestaltung der einzelnen Bereiche haben als mit der Auswahl *joomla*.

Im *Positionsstil* können Sie zunächst eine *Hintergrundfarbe* und eine *Rahmenfarbe* für den ganzen Bereich festlegen. Als *Rahmenstärke* können Sie einen beliebigen Wert zwischen 0 und 10 auswählen. Der *Rahmenradius* wirkt sich auf die Ecken aus, wobei 0px eine harte Kante zeichnet und alles andere zu einer abgerundeten Ecke führt. Mit dem *Rahmenabstand* stellen Sie den inneren Abstand der Felder zum äußeren Rahmen ein. Hier gelten die üblichen CSS-Regeln, um alle vier anzusprechen. Da wir vier Seiten haben (oben, rechts, unten und links), können Sie auch vier Zahlen eintragen.

Ich stelle mir dabei immer eine Uhr vor. Es beginnt immer mittags um 12:00 Uhr. Im Uhrzeigersinn kommt als Nächstes 3:00 Uhr, dann 6:00 Uhr und zuletzt 9:00 Uhr - oben, rechts, unten, links.

Damit können Sie für alle vier Seiten individuelle Abstände vorgeben.

```
001 10px 50px 100px 20px
```

Zwischen die einzelnen Zahlen kommt ein Leerzeichen (also kein Komma). Das ist leicht, denken Sie jetzt bestimmt. Dann können wir eine Runde weitergehen: Wenn Sie links und rechts gleiche Werte haben, können Sie natürlich den vierten Wert auch weglassen.

```
001 10px 50px 100px
```

... bedeutet, dass Sie nach oben einen Abstand von 10 px haben, nach rechts von 50 px und nach unten einen Abstand von 100 px. Da Sie für links keinen Wert definiert haben, wird der Wert für rechts einfach übernommen, in unserem Beispiel also 50 px. Sie ahnen bereits, dass es mit zwei Zahlen auch funktioniert:

```
001 10px 50px
```

Die erste Zahl bezieht sich nun auf oben und unten und die zweite Zahl auf links und rechts, also immer gegenüberliegend. Zu guter Letzt können wir auch eine Zahl in das Feld schreiben:

```
001 10px
```

Diese gilt für alle vier Abstände. Wir haben damit auf allen vier Seiten einen Abstand von 10 px zum äußeren Rahmen. In der *Positionslegende* stellen Sie ein, wie diese *ausgerichtet* ist. Im Vergleich zu *joomla* haben Sie hier nun auch die Möglichkeit, den Text mittig auszurichten, und beim *Feldnamen* geben Sie wieder den Alias des Feldes ein, von dem der *Standard Wert* übernommen werden soll.

Die *Feldbezeichnung* ist identisch mit der von *joomla*. Unter *Feldformular* geben Sie die *Rahmenfarbe* eines angeklickten Feldes aus. Wenn Sie also mit der Maus in das Feld klicken, können Sie hier die Farbe, die angezeigt werden soll, einstellen.

Die *Breite* wird erst so richtig ersichtlich, wenn Sie die Felder horizontal ausgeben. Haben Sie eine zu geringe Breite, kann es zu Überlappungen kommen. Ein Beispiel zeigt das folgende Bild.

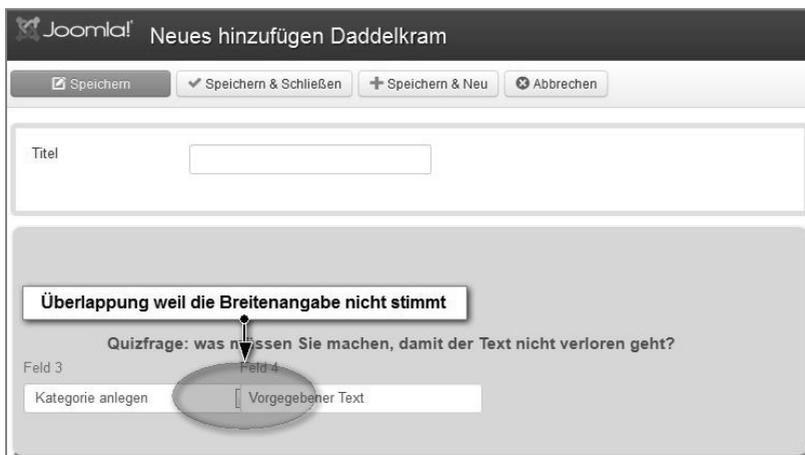


Bild 9.8: Breitenprobleme im `seb_css3`.

## 9.6 Menüverweis

Die Einstellungen hier sind weitestgehend identisch mit den Einstellungen für den Menüverweis bei *Listen & Suchtypen*. Lediglich die *URL Query (Vars)* gibt es bei den *Formularen & Inhaltstypen* nicht.

# 10

## Listen & Suchtypen

Das Aussehen der *Listen & Suchtypen* ist identisch mit dem der *Formulare & Inhaltstypen*. Dennoch haben die Konfiguration und teilweise auch die Nummern andere Bedeutungen. Auch finden sich in den *Ansichten* andere Bezeichnungen.

Sie können in diesem Bereich sowohl eine neue *Suche* erstellen, die über den ausgewählten Inhaltstyp geht, als auch eine *Liste*, vergleichbar mit der Joomla!-Kategorieliste, da Sie den Suchbereich auch ausblenden können.

### 10.1 Eine neue Suche

Klicken Sie auf *Listen & Suchtypen* und dann auf *Neu*. Sie erhalten dann eine Übersicht, in der Sie zunächst den Inhaltstyp für die Suche auswählen. Sie können aus den Standardtypen (*Artikel*, *Kategorie*, *User* und *User Group*) wählen oder aus eigenen Inhaltstypen.

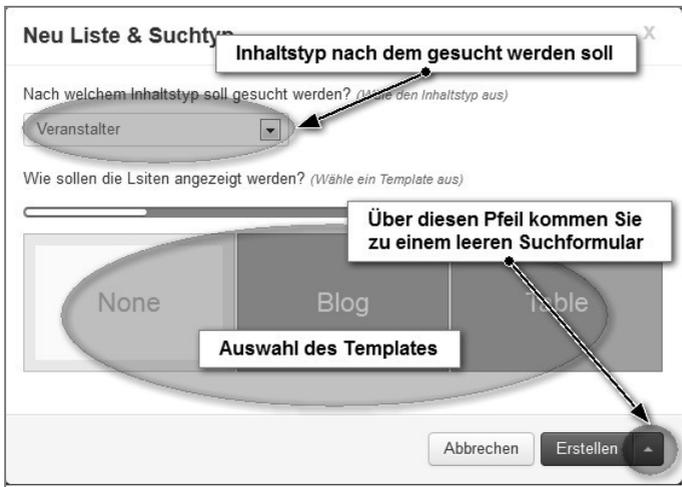


Bild 10.1: Eine neue Suchmaske anlegen.

Anschließend wählen Sie das *Template* aus. Die *Templates* *Blog* und *Table* entsprechen in der Ansicht den Joomla!-Menütypen *Kategorieblog* und *Kategorieliste* für Beiträge.

Wenn Sie den kleinen Aufwärtspfeil neben *Erstellen* anklicken, können Sie *Zurück zur Übersicht* auswählen. Das ist sicherlich eine falsche Übersetzung, denn Sie erhalten in Wirklichkeit die gleiche Eingabemaske wie bei der Auswahl *Template: Kein*.

Je nachdem, welches *Template* Sie ausgewählt haben, unterscheidet sich die Sucheingabemaske nach Anklicken von *Erstellen* ein wenig.

### 10.1.1 Template: Blog

Im *Blog* werden die Beiträge mit *Führenden*, *Einleitungen*, *Spalten* und *Links* ausgegeben. Einstellen können Sie das bei der Beitragsübersicht in den *Optionen*.

Dabei bedeutet die Anzahl *Führender* die Anzahl der Beiträge, die über alle Spalten reichen. Die *Einleitungen* geben die Anzahl der Beiträge in den Spalten wieder, die *Spalten*, in wie vielen Spalten die *Einleitungen* dargestellt werden, und die *Links* die Zahl der Verweise auf nachfolgende Beiträge, die Maximalzahl, die sich aus *Einführungen* und *Einleitungen* ergeben. Haben Sie mehr Beiträge, als Sie per Einstellung anzeigen können, erhalten Sie am Ende die Möglichkeit, »umzublättern«.

Damit das Suchergebnis im *Blog* auch angezeigt wird, müssen Sie nach dem Anklicken von *Erstellen* im Register *Element* die Felder angeben, die ausgegeben werden sollen.

Klicken Sie erneut auf *Template*, können Sie einige Einstellungen vornehmen.

Haben Sie ein eigenes Template entwickelt, können Sie es unter *Template* per Drop-down-Button auswählen. Zusätzlich haben Sie die Möglichkeit, eine CSS-Klasse anzugeben, mit der Sie das Frontend individuell formatieren können. Alle Einstellungen an dieser Stelle wirken allerdings nur auf das Suchformular. Die Ausgabe der Suchergebnisse haben Sie hingegen in *Inhalt* oder *Intro* der *Formulare & Inhaltstypen* festgelegt.

Im Bereich *Felder* geben Sie zunächst für diese Such- oder Listenmaske an, ob die Felderbezeichnungen angezeigt werden. Daneben geben Sie an, ob die Beschreibungen der Felder ausgegeben werden sollen und, wenn ja, wo genau sie stehen sollen. Die Beschreibungen finden Sie direkt bei den Feldern.

The screenshot shows the Joomla! search form configuration interface. At the top, there are buttons for 'Speichern', 'Speichern & Schließen', 'Speichern & Neu', and 'Schließen'. Below these, there are several fields: 'Titel \*' with a text input containing 'Article Title', 'App Ordner' with a dropdown menu showing '- Article', and 'Name' with a text input containing 'art\_title'. There are also 'On' and 'Off' toggle buttons. A callout box with an arrow points to the 'Beschreibung' field, which is currently empty and highlighted. The callout box contains the text 'Geben Sie hier eine Beschreibung für das Feld ein'. Below the main form, there is a section for 'Konstruktion' with a table header showing 'Bezeichnung', 'Typ', and 'Standard Wert'.

**Bild 10.2:** Beschreibungen der Felder eingeben.

Die einzelnen Anzeigemöglichkeiten bedeuten Folgendes:

- *Unter dem Feld:* Die Beschreibung wird direkt unter dem Feld angezeigt und nutzt die gesamte Breite.
- *Unter Formular/Wert:* Die Beschreibung wird unter dem Eingabefeld oder, wenn es ein reines Ausgabefeld ist, unter dem Wert ausgegeben.
- *Unter Bezeichnung:* Die Beschreibung wird nur unter der Bezeichnung für das Feld ausgegeben und nutzt nur die Breite der Beschreibung.

Der Bereich *Positionen* bietet zunächst die Auswahl der zu benutzenden CSS-Datei an und einen Rand für die Sucheingabe. Details zu den CSS-Dateien finden Sie in Kapitel 8.1 »JavaScript und CSS«.

Die Einstellungen der Positionen sind in Kapitel 3.1.4 »Eine Form muss her« (Ausgabe im Frontend: Inhalt) zu finden.

## 10.1.2 Template: Table

*Table* entspricht dem Joomla!-Menütyp *Kategorieliste*.

Sie müssen im Register *Liste* angeben, welche Felder in welcher Spalte ausgegeben werden sollen.

Im Register *Template* können Sie ein paar Einstellungen für die Formatierung der Ausgabe vornehmen. Der Bereich *Darstellung* ist identisch mit der Beschreibung im vorigen Kapitel.

Im Bereich *Style* finden Sie die CSS-Klassen für die Tabelle. In *Tabelle* stehen die CSS-Klassen, die auf die gesamte Tabelle wirken. Voreingestellt ist *category zebra table*. In *Tr (Gerade)* und *Tr (Ungerade)* finden Sie die CSS-Klassen für die einzelnen Zeilen.



Bild 10.3: Die CSS-Klassen im Frontend, mit dem Firebug sichtbar gemacht.

In *Table Layout* stellen Sie das generelle Verhalten der Tabelle zu ihren Inhalten ein. Dabei handelt es sich um CSS-Angaben:

- **table-layout:auto** - Der Inhalt bestimmt die Spaltenbreite. Bei umfangreichen Tabellen kann es zu Verzögerungen der Seitenanzeige kommen, da die Tabelle erst komplett geladen werden muss, damit der Browser die Spaltenbreite erkennen kann.
- **table-layout:fixed** - Die Spalten sind alle gleich groß, und der Inhalt wird, sofern er länger als die Breite der Zelle ist, umbrochen oder auch abgeschnitten, wenn nicht genügend Platz vorhanden sein sollte. Die Ladezeiten sind jedoch schneller, da bereits in der ersten Zelle die Breite der Spalten feststeht.

### 10.1.3 Template: Kein

Es wird der jeweilige Beitrag bzw. das Suchergebnis so ausgegeben, wie Sie es als *Content* im *Form- und Inhaltstyp* definiert haben. Sie brauchen bei dieser Auswahl keinerlei weitere Angaben zur Ausgabe zu machen.

Sie können aber dennoch eine Ausgabe definieren. Dazu klicken Sie auf den Tab/das Register *Template* und wählen dort wahlweise *Blog – seb\_blog* oder *Table – seb\_table*.

Wenn Sie *Blog* auswählen, können Sie das Register *Elemente* auswählen und es Ihrer gewünschten Ausgabe anpassen.

#### **Zurück zum Urzustand**

Wenn Sie sich im Register *Elemente* befinden, können Sie das Template nicht ändern. Das können Sie nur im Register *Liste*, auch dann, wenn Sie überhaupt keine Liste ausgeben wollen. Haben Sie bereits ein Template ausgewählt, können Sie auch durch Auswahl von *Disable List Template* alle Templates löschen und kommen so wieder in den Urzustand zurück.

Sie fügen in das *Suchformular* ein Feld oder mehrere Felder Ihres Formulars ein, mit denen gesucht werden kann.

## 10.2 Konfiguration

Die *Konfiguration* ist bei allen Templates gleich und wurde teilweise bereits in den obigen Kapiteln erklärt.

### 10.2.1 Bereich: Konfiguration

Der obere Bereich umfasst die *Konfiguration* für das ganze Suchformular (*Suchtyp*).

Wenn Sie die Suche das erste Mal per Menüverweis aufrufen, werden bereits einige Beiträge vorab angezeigt. In *Limit* können Sie angeben, wie viele Beiträge bei Erstaufruf und bei Suchtreffern maximal angezeigt werden sollen. Geben Sie eine 0 ein, werden immer alle infrage kommenden Beiträge angezeigt.

Der *Cache* ist ein Zwischenspeicher, auf den der Browser zurückgreift, um Inhalte nicht jedes Mal neu laden zu müssen. Damit kann eine Webseite schneller gemacht werden, es hat aber den Nachteil, dass Änderungen meist erst später angezeigt werden. Der *Cache* läuft nach einer bestimmten Zeit, die Sie in der Joomla!-Konfiguration einstellen können, ab. Die Seiten werden dann neu geladen. Vor der Ablaufzeit werden die Seiten in der Regel nicht neu geladen. Haben Sie eine Website, auf der sich ständig etwas ändert, ist es sinnvoll, den *Cache* auszuschalten, was standardmäßig vorgegeben ist.

Sie können aber sowohl bei der Suche als auch bei der Anzeige der Darstellung den *Cache* einschalten. Die *Globale Einstellung* übernimmt die Einstellungen aus der Joomla!-Konfiguration und erlaubt bei der Suche zusätzlich die Angabe, ob der *Cache* nur für den suchenden User oder generell für alle genutzt werden soll. Die Einstellung *Selbst* kann ich mir nur damit erklären, dass während der Suche vorübergehend der *Cache* benutzt und direkt nach der Suche wieder abgeschaltet wird. Allerdings kann ich dies leider nicht mit Gewissheit sagen.

*Optionale Variante* – Leider muss ich auch bei dieser Einstellung passen. Der Gedanke, dass das *Limit* beim Erstaufwurf funktioniert und die *Optionale Variante* bei den Suchergebnissen, hat sich leider nicht bestätigt.

Die *Seitenzahl* gibt an, wie viele Beiträge pro Seite angezeigt werden sollen. Stellen Sie eine *1* ein und haben fünf Treffer, dann werden fünf Seiten mit jeweils einem Beitrag pro Seite ausgegeben.

#### Am Limit

Sie müssen hier ein wenig aufpassen, dass Sie das Limit nicht zu niedrig einstellen, und eventuell mit den Zahlen beim Limit und der Seitenzahl ein wenig jonglieren.

Stellen Sie die *Fehlersuche* auf *jedermann*, werden auch bei nicht angemeldeten Besuchern Ihrer Website Debuginformationen ausgegeben. Also auch wenn alles einwandfrei läuft, werden der SQL-Aufruf und die Geschwindigkeit des Seitenaufbaus ausgegeben. Diese Einstellung empfiehlt sich daher nur bei lokaler Benutzung oder während der Entwicklung.

Besser ist allerdings die Einschränkung auf den *SuperBenutzer* – dann sehen nur Sie diese Informationen, was aus Sicherheitsgründen stets der bessere Weg ist. Bei *Nutze Allgemein* – können Sie in den *Optionen* im Tab *Seite* die *Fehlersuche* global vorgeben.

*SEF* steht für *Search Engine Friendly*, zu Deutsch *Suchmaschinenoptimierung* (*SEO*). Je nachdem, was Sie in der Joomla!-Konfiguration eingestellt haben, werden Ihre URLs ausgegeben. Die Einflussnahme ist nicht sehr vielfältig, und manchmal möchten Sie vielleicht ein bestimmtes Aussehen Ihrer URL haben. Sie können in SEBLOD im Feld *SEF Urls* dieses Aussehen entsprechend gestalten. Allerdings müssen Sie anschließend ausprobieren, ob noch alles einwandfrei funktioniert. Bei mir haben viele Einstellungen zu Fehlermeldungen geführt.

*SEF Inhaltstyp* – Hier kann ich leider auch nur eine Vermutung anstellen, die sich nicht testen ließ. Wenn Sie nach eigenen *Formularen & Inhaltstypen* suchen, können Sie hier angeben, wie es danach weitergeht – also nach der Suche. Meistens werden Sie vermutlich mit Artikeln weiterarbeiten (*article*) oder vielleicht mit Usern (*user*) ...

*Inhalt erstellen* bezieht sich auf die Einstellungen in der *Seitenanzahl*. Während Sie noch auf der einen Seite auf die Beiträge schauen, bereitet SEBLOD bereits die nächste Seite zwecks schnellerer Anzeige vor. Wenn Sie eine große Site mit vielen

Zugriffen haben, kann es sinnvoll sein, *Inhalt erstellen* auf *Nein* zu stellen, um etwas Performance zu sparen. Andererseits kann damit der Seitenaufruf schneller gemacht werden, aber es muss intern zwischengespeichert werden.

## 10.2.2 Bereich: Allgemein

Hier stellen Sie konkret das Aussehen Ihrer Suchergebnisse ein. Die meisten Einstellungen, die Sie hier finden, können Sie auch direkt im Menüverweis zur Suche einstellen.

Die *Automatische Weiterleitung* greift erst bei langen Suchmasken. *Automatische Weiterleitung* regelt, was nach der Suche geschehen soll: Soll auf den Inhalt geleitet werden oder in das Suchformular?

Die *Standard Reihenfolge* bezieht sich auf die Ergebnisse und kann Ihren Bedürfnissen angepasst werden.



**Bild 10.4:** Einstellungen für das Aussehen der Suchergebnisse.

Stellen Sie *Zeige Titel* auf *Zeige*, wird der Titel des *Suchformulars* bzw. der *Liste* ausgegeben. Sie können dann zusätzlich die Größe der Überschrift von *h1* bis *h3* einstellen. Damit können Sie für eine saubere Gliederung Ihrer Webseite sorgen.

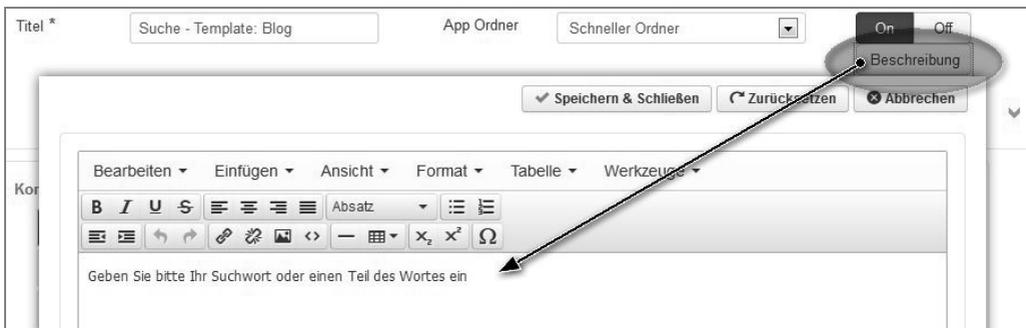
### Überschriften: h1, h2 und h3

Die Webdesigner dieser Welt streiten darüber, aber dennoch ist es sinnvoll, eine h1-Überschrift pro Seite nur ein einziges Mal zu verwenden, da sie von Suchmaschinen als besonders wichtig eingestuft wird – weshalb Sie diese Überschrift bewusst setzen sollten. Viele Templates »vergeuden« diese Überschrift für das Logo oder den Slogan der Webseite. Sie können bei richtiger Setzung der Struktur Ihrer Webseite ein Plus bei der Suchmaschinenoptimierung setzen, da die Suchmaschinen Ihre Überschriften entsprechend gewichtigen und den Inhalt entsprechend indizieren. Die h2-Überschrift steht in der Gliederung an zweiter Stelle und ist demzufolge ebenfalls sehr wichtig. Eine h3-Überschrift wird standardmäßig von Joomla! für die Modultitel benutzt, kann aber natürlich von Ihnen angepasst werden. h3 wird oft für Unterkapitel benutzt.

Zusätzlich zur Größe der Überschrift können Sie eine CSS-Klasse angeben, um Ihre Überschrift individuell zu gestalten. Wählen Sie beispielsweise eine h2-Überschrift aus und geben als Klassennamen `h2klasse` ein, steht im Quellcode zur Laufzeit diese Zeile:

```
001 <h2 class="h2klasse">Mein Suchformular</h2>
```

*Zeige Beschreibung* bezieht sich auf den Beschreibungstext, den Sie bei der Form angeben können.



**Bild 10.5:** Beschreibungstext eingeben.

Wahlweise können Sie diesen über oder unter dem Suchfeld ausgeben oder ganz verbergen.

Ebenso verhält es sich mit dem *Suchformular*: wahlweise *über* oder *unter* der Ergebnisliste. Wenn Sie es *verbergen*, erhalten Sie lediglich eine *Liste* Ihrer Beiträge. Da Sie aber auch Filter setzen können, haben Sie die Möglichkeit, eine individuell zusammengestellte Liste auszugeben.

Die *Artikelnummer* steht im Zusammenhang mit dem Feld *Elemente Beschriftung*. Sie haben hier die Möglichkeit, die Anzahl der Suchtreffer anzuzeigen und mit einer Beschriftung zu versehen.

Wie in der Abbildung 10.4 gezeigt, habe ich Beiträge gefunden und als Beschriftung eingetragen.

Dazu können Sie noch eine CSS-Klasse angeben, damit Sie die Beschriftung individuell formatieren können. Die Formatierung bezieht auch die Artikelnummer ein. Geben Sie als Klasse `ergebnis` ein, finden Sie zur Laufzeit im Quellcode diese Zeilen:

```
001 <div class="ergebnis"><span>3</span>Beiträge gefunden</div>
```

Ein kleiner Tipp für die nicht so Bewanderten in CSS:

```
001 .ergebnis { color:green; } /* gibt Ihnen die Zeile in grüner Schrift
aus. */
002 .ergebnis span { font-weight:bold; } /* Die Zahl (im Beispiel die "3")
wird noch zusätzlich fett ausgegeben. */
```

Sie sehen also, dass Sie die einzelnen Bereiche Ihres Suchformulars und der Ergebnisse sehr individuell gestalten können.

Mit *zeige Seitenzahl* haben Sie die Möglichkeit, unten einen Überblick darüber zu geben, auf welcher Seite von wie vielen sich der User gerade befindet.

Die *Seitennummerierung* erlaubt das Blättern durch mehrere Seiten und kann wahlweise *über*, *unter*, sowohl über als auch unter – *beide* – angezeigt werden oder ganz verborgen werden. Mit *verbergen* sollten Sie jedoch vorsichtig umgehen. Haben Sie mehr Treffer, als auf eine Seite passen, dann werden diese niemals angezeigt, wenn Sie die *Seitennummerierung* ausschalten.

Für die *Seitennummerierung* können Sie zusätzlich eine CSS-Klasse definieren. Standardmäßig ist hier *pagination* eingetragen. Diese CSS-Klasse wird von Joomla! und somit von den allermeisten Templates benutzt. Aber natürlich können Sie das ändern und eine eigene Formatierung definieren.

### 10.2.3 Bereich: Kein Ergebnis

Wenn Sie *kein Ergebnis* für die Suche haben, dann haben Sie die Möglichkeit, hier anzugeben, wie reagiert werden soll.

Der *Nachrichtenstil* wird jeweils, meist oben im Template, ausgegeben. Es hängt letztlich vom verwendeten Template ab. Wird gar nichts ausgegeben, fehlt im Template vermutlich diese Zeile:

```
001 <jdoc:include type="message" />
```

Diese Zeile ist dafür verantwortlich, dass entsprechende Systemmeldungen ausgegeben werden. Das sind *Nachrichten*, *Mitteilungen* und *Fehlermeldungen*. Alle werden mit unterschiedlichen Farben hinterlegt. Standardmäßig sind das Braun für *Hinweise*, Rot für *Fehler* und Blau für *Nachrichten*.

Sie können einen eigenen *Nachrichtentext* vorgeben. Wählen Sie als *Aktion Datei einfügen* aus, wird diese Nachricht noch einmal zusätzlich im Ergebnisbereich ausgegeben. Damit können Sie auch in Templates, in denen die oben genannte Zeile fehlt, Ihre Nachricht ausgeben. Wählen Sie unter *Aktion* den Eintrag *Template darstellen* oder *Kein* aus, wird die farbliche Joomla!-Ausgabe wie oben beschrieben ausgegeben. Die Option *Zeige Beschreibung* hatte bei mir leider keine Funktion.

## 10.3 Fields

Der *Konstruktionsbereich* ist weitgehend identisch mit den *Formularen & Inhaltstypen*. Die Nummern entsprechen *Liste* und *Element* dem *Intro* in *Listen & Suchtypen* und *Inhalt* den *Formularen & Inhaltstypen*.

Details finden Sie in Kapitel 10.5 und Kapitel 11. Es gibt lediglich eine Ausnahme bei *Nummer 3* im *Suchformular*.

## 10.4 Das Feld CCK

Dieses Feld ist von SEBLOD vorgegeben und wird benötigt, um zu erkennen, in welchen Inhaltstypen gesucht werden kann. Dieses Feld prüft, welcher Inhaltstyp auf *veröffentlicht* steht, und erlaubt so die Suche innerhalb dieses Typs. Wenn Sie dieses Feld löschen, funktioniert die Suche entweder gar nicht mehr oder nur unzureichend. Das Feld selber ist auch auf *Verborgenen* gesetzt. Es wird also nicht in Ihrem Formular angezeigt, jedoch von SEBLOD intern benutzt. SEBLOD speichert das Ergebnis seiner Überprüfung mit diesem Feld auch in eine eigene Datenbanktabelle.

## 10.5 Nummer 3 im Suchformular

Befinden Sie sich im *Suchformular*, nehmen Sie mit *Nummer 3* direkten Einfluss auf die Art der Suche pro Feld und können hier sehr detaillierte und individuelle Sucheinstellungen vornehmen.

- *Kein* – Das Feld wird zwar als Suchfeld angezeigt, und Sie können dort auch beliebige Suchbegriffe eintragen, aber es passiert nichts. Die Suche wird als solche nicht ausgeführt. Das Feld ist also so eine Art »Placebo«. Es ist zwar da, hat aber keinerlei Sinn. Eventuell könnte das sinnvoll sein, wenn Sie das Feld vorbelegen möchten ...

### 10.5.1 Basis

- *Beginnt mit* – Als Suchergebnis werden nur Resultate ausgegeben, die mit den Buchstaben beginnen, die Sie eingeben.
- *Endet mit* – Das Gegenteil von *Beginnt mit*. Das Suchergebnis endet mit den Zeichen, die Sie in das Suchfeld eingeben.
- *Genau* – Es muss identisch mit der Zeichenkette sein, die Sie eingeben.
- *Zulässig (Standard)* – Es wird alles ausgegeben, worin die Zeichenkette, die Sie eingeben, vorkommt. Beispielsweise ergibt der Eintrag »bei« das Suchergebnis »Beitrag Nummer 2« und »Ein toller Beitrag«.

### 10.5.2 Datum und Uhrzeit

- *Älter als (<)* – Älter als das eingegebene Datum/die Uhrzeit.
- *Älter gleich (< or =)* – Älter als das eingegebene Datum/die Uhrzeit oder gleich.
- *Jünger gleich (> or =)* – Jünger oder gleich.
- *Jünger als (>)* – Jünger als das eingegebene Datum/die Uhrzeit.

### 10.5.3 Ausschluss

- *Beginnt nicht mit* – Der gesuchte Text darf nicht mit den eingegebenen Buchstaben beginnen.
- *Nicht leer* – Es ist hier unwichtig, was Sie in das Suchfeld eingeben. Entscheidend ist lediglich, dass das Feld, nach dem gesucht wird, nicht leer ist. Dabei bezieht sich *nicht leer* auf das Ergebnis. Beiträge mit dem zu suchenden Feld, in denen nichts steht, werden nicht im Suchergebnis angezeigt.
- *Nicht endend mit* – Der gesuchte Text darf nicht mit den eingegebenen Buchstaben enden.
- *Nicht identisch* – Zeigt alles an, was nicht mit dem eingegebenen Text übereinstimmt, wobei *identisch (equal)* in der Regel im Zusammenhang mit Zahlen steht. Für Zeichenketten nehmen Sie besser *Like*.
- *Nicht Null* – Es muss etwas im Suchfeld stehen. Leere Felder werden nicht angezeigt, sehr wohl aber Felder mit leerer Zeichenkette. Siehe dazu auch den unteren Tippkasten.
- *Nicht In* – Diese Abprüfung funktionierte bei mir leider nicht.
- *Nicht gleich* – Im Englischen steht hier *Not Like*. Mit *Like* prüfen Sie Zeichenketten. Es wird alles angezeigt, was nicht Ihrer Eingabe entspricht.

### 10.5.4 Null

- *Ist Null* - Wird nur angezeigt, wenn absolut nichts im Feld steht (siehe auch unten den Tippkasten).
- *Ist nicht Null* - Wird nur angezeigt, wenn etwas im Feld steht.

### 10.5.5 Numerisch

- *Größer (>)* - Nur wenn der Wert größer ist als der eingegebene Wert.
- *Größer gleich (> or =)* - Nur wenn der Wert im Feld größer oder gleich ist.
- *Kleiner gleich (< or =)* - Der Wert muss kleiner sein als der angegebene Wert oder gleich.
- *Kleiner als (<)* - Der Wert muss kleiner sein.

### 10.5.6 Wörter

- *Jede Wörter* - Eines der eingegebenen Wörter muss passen.
- *Eines der Wörter genau* - Dies ergab bei mir leider immer die Mitteilung, dass es kein Ergebnis gäbe. Hier handelt es sich offenbar um einen Bug.
- *Jedes Wort* - Auch diese Einstellung arbeitete bei mir fehlerhaft. Der Suchbegriff verschwand, und es wurde keine Ergebnisliste ausgegeben.
- *Exakte Verschachtelung (nested exact)* - Führt leider zu einer 1054-Statusmeldung, also einem Fehler. Theoretisch müsste sich damit eine Verschachtelung prüfen lassen, wenngleich mir auf Anhieb kein Szenario dazu in Joomla! einfallen würde, zumindest nicht bei der Suche im Frontend.

#### **Bedeutung der Null**

*Null* ist nicht gleichbedeutend mit leer oder der Zahl 0!

In der Datenbank steht zunächst in einer Datenzeile *Null*, was gleichbedeutend ist mit »rein gar nichts«, nicht einmal einer leeren Zeichenkette, denn die wäre ja durchaus ein legitimer Wert. Aber *Null* bedeutet, dass noch nicht einmal das - eine leere Zeichenkette - in der Datenzeile enthalten ist.

Klicken Sie einmal in das Feld und verlassen es wieder, ohne etwas einzutragen, wird aus *Null* eine leere Zeichenkette: »«.

Somit steht also jetzt etwas in dem Feld.

Eine leere Zeichenkette bedeutet, dass das Feld bearbeitet wurde, dass jemand sich dabei etwas gedacht hatte, dieses Feld leer zu lassen.

Angenommen, Sie haben eine Krankenakte. Leer bedeutet dann vielleicht, dass der Patient gesund ist, *Null* hingegen, dass Sie das noch nicht mit Gewissheit sagen können.

## 10.6 Ordering

Sie können hier angeben, nach welchen Kriterien die Reihenfolge der Suchergebnisse ausgegeben wird. Das Feld, das weiter oben steht, ist in der Hierarchie wichtiger als das nachfolgende. In jedem Feld geben Sie dann noch an, ob in *aufsteigender* oder *absteigender* Reihenfolge gelistet werden soll.

Angenommen, Sie haben Namen und Städte in dieser Reihenfolge jeweils *aufsteigend* ausgewählt. Dann wird zuerst nach Namen von A bis Z sortiert, bei Namensgleichheit nach den Städten von A bis Z.

## 10.7 Liste

Die *Liste* kommt nur bei der Template-Auswahl *Table* zum Vorschein. Bei der Auswahl *Template: Blog* erhalten Sie die Mitteilung, dass es keine Position für diese Ansicht gibt. Sie geben hier die Felder an, die in der Trefferliste der Suche und/oder der Liste angezeigt werden sollen. Im Gegensatz zu allen anderen Ausgaberegistern haben Sie hier die Spaltenpositionen (*# Column-A*). Zur Erinnerung: Die Darstellung ist vergleichbar mit der Joomla!-Kategorieliste.

Die Nummern sind identisch mit *Intro* und *Inhalt* der *Formulare & Inhaltstypen*.

## 10.8 Element

Das Register *Element* erhalten Sie nur bei der Template-Auswahl *Blog*. Ansonsten ist diese Registerkarte nicht auswählbar. Sie ziehen hier alle Felder hinein, die im Blog angezeigt werden sollen. Die Ausgabe ist ansonsten identisch mit dem *Inhalt* der *Formulare & Inhaltstypen*.

## 10.9 Menüverweis

Die Suche bzw. Liste wird über das Menü und den *Menüeintragstyp SEBLOD/Liste & Suche* und anschließender Auswahl Ihrer erstellten *Suche* oder *Liste* (*Suchtyp (Liste)*) angezeigt. Sie haben im Menü weitere Einstellungsmöglichkeiten, die Sie auch im *Suchtyp* einstellen können und die direkt am jeweiligen Feld erklärt werden. Die Einstellungen können Sie aber auch im Menü überschreiben. Das könnte dann sinnvoll sein, wenn Sie eine *Liste* oder einen *Suchtyp* mehrmals anzeigen (mehrere Menüverweise) und mit individuellen Unterschieden ausstatten möchten.

Direkt unter dem *Suchtyp* haben Sie im Menüverweis die Möglichkeit, einzustellen, ob beim ersten Aufruf der *Suche* bereits eine Liste ausgegeben werden soll oder nicht. Dazu stellen Sie Entsprechendes bei *Suche? (beim ersten Laden)* ein.

Menütitel \* Suche Template: Blog

Details Grundlegende Optionen ( CCK ) Overrides (Live / Variation) Erw

Menüeintragstyp \* Liste & Suche

Suchtyp (Liste) \* Suche - Template: Blog

Suche? (beim ersten Laden)  Nein  Ja

Link

Zielfenster

Template-Stil

**Bild 10.6:** Menüverweis für die Suche.

Bei den *Grundlegenden Optionen* haben Sie ganz unten ein zusätzliches Feld, bei dem Sie an die URL weitere Variablen hängen können. Beispielsweise können Sie eine vorgegebene Kategorie übergeben, indem Sie bei *URL Query (Vars)* in das Feld *art\_catid=13* eintragen. 13 ist bei mir eine bestimmte Kategorie, und ich übergebe diesen Wert bei Erstaufruf der *Liste* und des *Suchtyps*.

Unter *Overrides (Live/Variation)* haben Sie die Möglichkeit, Suchbegriffe vorzugeben und zu entscheiden, ob und wie das Feld angezeigt wird. Damit überschreiben Sie die Einstellungen im *Suchtyp*.

*Raw Rendering* – so verstehe ich es jedenfalls – sollte Auswirkungen auf das Design haben und Individualeinstellungen diesbezüglich ignorieren, wie eigene Markups. Allerdings entstanden bei meinen Tests zwar sehr leichte Veränderungen im Design, aber nicht gravierend genug, um Ihnen dazu Einzelheiten mitzuteilen. Vermutlich übersehe ich an dieser Stelle jedoch etwas. Sollten Sie es herausfinden, freue ich mich über eine E-Mail von Ihnen.

Alle anderen Menüeinstellungen kennen Sie von Joomla!.



## Die Nummern in den Formularen & Inhaltstypen

Die Zahlenreihe in der Übersicht *Inhaltstypen und Formulare* haben, je nach ausgewähltem Formulartyp, unterschiedliche Funktionen. Wenn Sie Änderungen vornehmen, müssen Sie auch stets den ganzen Inhaltstyp speichern, da Sie sonst die Änderungen im Back- und/oder Frontend nicht sehen. Die einzelnen Nummern gelten auch für die *Listen & Suchtypen* – mit Ausnahme der *Nummer 3* (siehe Kapitel 10.5) beim *Suchformular* bzw. *Admin Formular*. *Nummer 6* ist bei den *Listen & Suchtypen* zusammengefasst mit *Nummer 3* und *Nummer 6* bei den *Formularen & Inhaltstypen*.

### **Tipp**

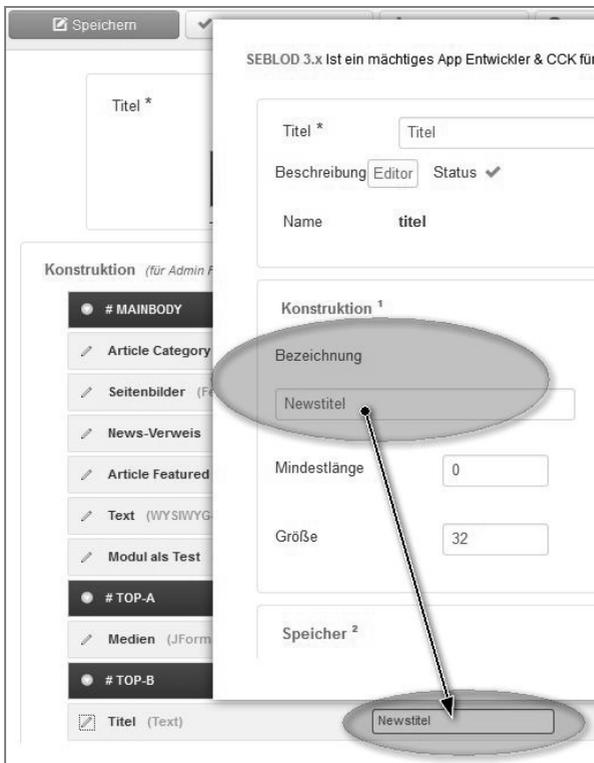
Wenn Sie eine Nummer ab *Nummer 2* auswählen und dann ein neues Feld anlegen, ist das Feld in der Übersicht immer mit der Ansicht von *Nummer 1* vorbelegt. Also nicht wundern, wenn die Anzeige etwas durcheinandergerät. Einfach eine andere Nummer auswählen, dann passt wieder alles.

## 11.1 Nummer 1

Diese Nummer ist bei Aufruf eines Inhaltstyps standardmäßig ausgewählt.

### 11.1.1 Einzelne Felder

In der Spalte *Bezeichnung* sehen Sie zum einen den Text, den Sie im Bezeichnungsfeld jeweils eingetragen haben.



**Bild 11.1:** Bezeichnungen im Überblick.

Wenn Sie keinen *Bezeichnungsnamen* eingetragen haben, wird auch nichts ausgegeben. Tragen Sie in dieser Übersicht in ein *Bezeichnungsfeld* das Wort *clear* ein, wird die Bezeichnung für dieses Feld nicht ausgegeben. Siehe dazu auch Seite 47 Abschnitt »Ausgabe im Frontend: Inhalt«.

### 11.1.2 Admin- und Seitenformular

In diesen Formularen sehen Sie in der rechten Spalte unter *Variante* verschiedene mögliche Einstellungen zu den einzelnen Feldern.



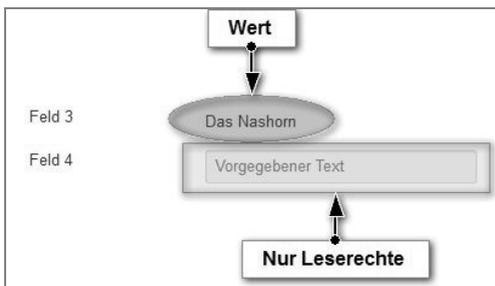
**Bild 11.2:** Wie soll das Feld angezeigt werden?

*Standard* ist die Voreinstellung und gibt das Feld so aus, wie es normalerweise funktioniert, beispielsweise ein Textfeld, in das Sie Text eingeben können.

Wählen Sie *Verborgen* aus, wird das Feld nicht ausgegeben, sehr wohl aber mit abgespeichert, wenn das Formular ausgefüllt wird. Sie können so feststehende Felder mit fest vorgegebenem Text, den niemand sehen und ergo verändern kann, erzeugen. Das ist vielleicht sinnvoll, wenn Sie verschiedene Abteilungen schreiben lassen und für sich selbst als Notiz die Abteilung fest angeben. Sie können aber auch dynamische Felder auf *Verborgen* setzen, die dynamische Inhalte im Verborgenen erzeugen. Details dazu siehe bitte unter dem Kapitel 8 »Alle Felder«.

*Wert* gibt den Inhalt des *Standard Wertes* aus. Dieser kann nicht verändert werden

*Nur Leserechte* gibt auch den *Standard Wert* aus, jedoch innerhalb eines Feldes:



**Bild 11.3:** Die unveränderliche Ausgabe des *Standard Wertes* ist eine optische Entscheidung.

## 11.2 Nummer 2

Zwischen Eingabe- und Ausgabeformular liegen sehr große Unterschiede. Beziehen sich die Einstellungen zu *Nummer 2* im Admin- und Seitenformular auf vorgegebene Werte, geht es bei der Ausgabe in *Intro* und *Inhalt* um weiterführende Verweise.

## 11.2.1 Admin- und Seitenformular – Live und Live-Wert

Sie können Werte vorgeben. Das können Sie im Feld über *Standard Wert* machen, aber auch an dieser Stelle, wobei Sie unter *Nummer 2* mehr Möglichkeiten haben.

In Kapitel 3.2.1 hatten wir ein Kategoriefeld angelegt. Der *Standard Wert* direkt im Feld hätte sich fortan auf alle Felder vom Typ *Kategorie* ausgewirkt. Um das zu vermeiden, hatten wir unter dem *Live Wert* eine entsprechende Angabe gemacht. Das ist denn auch einer der großen Unterschiede: Eine *Standard Wert*-Zuweisung im Feld direkt hat Einfluss auf alle Felder dieses Typs. Eine Zuweisung über den *Live Wert* hat nur Einfluss auf eben dieses eine Feld in diesem einen Formular. Damit können Sie auch im *Admin Formular* einen anderen Standardwert (*Live Wert*) vorgeben als im *Seitenformular*.

In der ersten Spalte (*Live*) definieren Sie zunächst, welcher Art der *Live Wert* sein soll.

- *Standard*
- *User*
- *Variable*

*Standard* bietet Ihnen, je nach Feldtyp, eventuell weitere Möglichkeiten in der Konfiguration des *Live Wertes*. Haben Sie beispielsweise ein Auswahlfeld (Drop-down), dann erhalten Sie einen Pfeil, mit dem Sie einen Wert aus Ihrer Auswahl direkt zuweisen können.

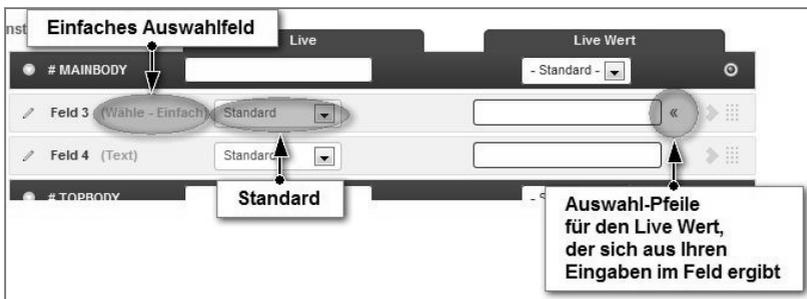


Bild 11.4: Auswahl für den *Live Wert*.

Wenn Sie beispielsweise ein einfaches Textfeld haben, dann haben Sie natürlich keine Auswahlmöglichkeit an dieser Stelle, können aber beliebigen Text als *Live Wert* eingeben

Wählen Sie unter *Live* den *User* aus, müssen Sie in der *Konfiguration* einen Datenbanktabellenspaltennamen aus der *#\_users* in *Eigentum* eintragen. Beispielsweise können Sie *username* in *Eigentum* eintragen und erhalten dann den jeweils aktuellen Usernamen angezeigt, also den Usernamen, der das Formular aufruft und ausfüllt. Die Spaltennamen erhalten Sie entweder mit einem Blick in die Datenbank oder indem Sie

das Feld editieren, im Speicherbereich *Benutzer* auswählen und sich dann die Spaltennamen anschauen.

Wählen Sie unter *Live* die *Variable* aus, können Sie Werte zusätzlich übergeben. Dazu wählen Sie zunächst den *Typ* aus, den Sie übergeben möchten. Falls Sie sich unsicher sind, finden Sie in der Datenbank bei den jeweiligen Feldern Hinweise dazu, um welchen Datentyp es sich handelt. *Mehrfach* gibt an, ob Sie eine Variable oder mehrere übergeben möchten. In *Variable* schreiben Sie den *Aliasnamen* des Feldes und in *Standard Wert* den zu übergebenden Wert. In *Ignore »0 / NULL«* geben Sie an, ob ein leeres Feld ignoriert werden soll oder nicht.

Auf Seite 162 erkläre ich im Tippkasten (»Bedeutung der Null«), was sich hinter *Null* verbirgt. In der *Encryption* können Sie *Base64* oder *Kein* einstellen. *Base64*<sup>28</sup> ist eine Codierung, die Sie bei verschlüsselten Daten benutzen.

In der Abbildung übergebe ich dem Kategoriefeld eine bestimmte Kategorie. Die Kategorie-ID ist vom Typ *Integer (Int)*.

Konstruktion <sup>1</sup>			
Typ	<input type="text" value="Int"/>	Mehrfach	<input type="text" value="Nein"/>
Variable	<input type="text" value="art_catid"/>	Ignore "0 / NULL"	<input type="text" value="Nein"/>
Standard Wert	<input type="text" value="10"/>	Encryption	<input type="text" value="- Kein"/>

**Bild 11.5:** Beispiel einer Variablenübergabe.

Eine andere Möglichkeit ist es, Inhalte von Variablen abzufragen. Legen Sie einen neuen Beitrag an, können Sie so beispielsweise Werte von Variablen abfragen und speichern. Beispielsweise finden Sie in der URL den Eintrag

```
001 option=com_cck
```

Gebe ich als Variable *option* an und lasse den *Standard Wert* frei, erhalte ich den Wert der Variablen. In meinem Beispiel ist das *com\_cck*:

<sup>28</sup> <http://de.wikipedia.org/wiki/Base64>

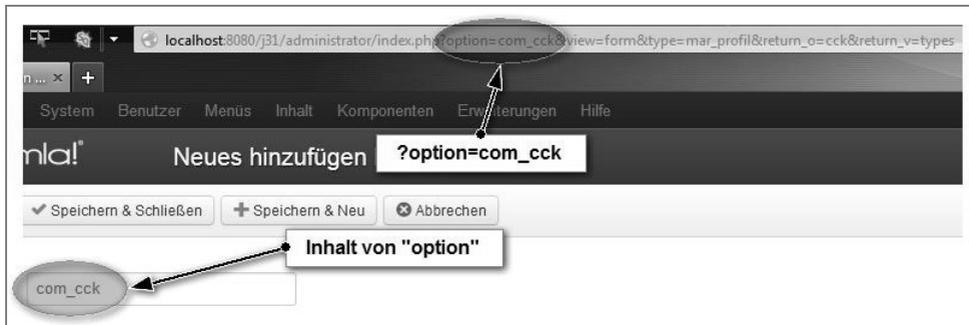


Bild 11.6: Inhalte von Variablen als vorgegebene Werte übernehmen.

## 11.2.2 Intro und Inhalt

In der linken Spalte können Sie verschiedene Verlinkungen vorgeben, wogegen Sie in der rechten Spalte Markups vergeben.

Markups sind CSS-Klassen. Das können sowohl eigene – also CSS-Klassen, die Sie formatieren – als auch vorgegebene Klassen sein. Die möglichen vorgegebenen Klassen finden Sie in Anhang A.

## 11.3 Nummer 3

### 11.3.1 Admin- und Seitenformular

In der ersten Spalte erhalten Sie einen Hinweis, ob das Feld *Optional* oder *Erforderlich* ist. Wenn Sie auf das Wort klicken, können Sie es entsprechend umschalten und so aus einem optionalen ein Pflichtfeld machen und umgekehrt. Außerdem können Sie auch eine Überprüfung der Eingabe machen. Für beides können Sie zudem einen Text vorgeben, der eingeblendet wird, wenn die jeweiligen Einstellungen vom Anwender ignoriert werden.

Handelt es sich um ein *erforderliches* Feld und ignoriert ein Anwender dieses, erscheint der Text, den Sie unter *Warnung* eingeben. Andernfalls erscheint der Standardtext.

The screenshot shows a form with three sections. At the top right, there are three buttons: 'Speichern & Schließen', 'Zurücksetzen', and 'Abbrechen'. The first section, 'Erforderlich', has a label 'Erforderlich', a dropdown menu with 'Nein' selected, and a 'Warnung' field. The second section, 'Bestätigung', has a label 'Bestätigung', a dropdown menu with 'Kein' selected and a list of options (Ajax, Alphanumerisch, Internet, Kontakt, Ohne Kategorie) visible, and a 'Warnung' field. The third section is empty.

**Bild 11.7:** Pflichtfeld und Überprüfung der Eingabe.

Unter *Bestätigung* können Sie auswählen, welcher Art der Text sein muss, der in das Feld eingegeben wird, und bei fehlerhafter Eingabe im *Warnungsfeld* einen Text vorgeben, der bei Fehleingabe erscheint. Geben Sie eine *Bestätigung* vor, erscheint in der Übersicht hinter *Optional* oder *Erforderlich* +1. Die *Bestätigungen* zeigen teilweise weitere Eingabefelder an, im Einzelnen:

- *Kein*: Keine Vorgabe – es kann beliebiger Inhalt eingegeben werden.
- *Ajax* – *Availability*: Mit dieser Überprüfung kann in Bezug auf eine Datenbanktabelle überprüft werden, ob der eingegebene Wert noch frei oder bereits belegt ist. Beispielsweise können Sie so überprüfen, ob der gewünschte Username noch frei ist oder ein Gutscheincode noch nicht vergeben wurde.

Im Beispiel habe ich in der *Tabelle* die *#\_\_users* ausgewählt und in *Spalte username* eingetragen. Leider müssen Sie das hier per Hand reinschreiben, weshalb Sie zuvor einen Blick in die Datenbank werfen müssen, um die Spaltenbezeichnung herauszufinden. Alternativ können Sie im Speicherbereich eines Feldes auch nachschauen, wie die Spalte heißt.

In *ID* muss der Spaltenname hinein, der die ID der Datenbanktabelle beinhaltet. Das ist ganz oft auch der Spaltenname *id*, aber leider nicht immer, weshalb Sie hier auch in die Datenbank schauen müssen.

Leider kann ich Ihnen nicht sagen, was das Feld *Felder* bewirkt. Bei meinen Tests hatte ich meist eine Fehlermeldung, oder es passierte nichts.

**Bild 11.8:** Ajax-Überprüfung, ob der Name noch frei ist.

- *Alphanumerisch - Nur Buchstabenanzahl:* Eine beliebige Zeichenkette aus alphanumerischen Zeichen. Alphanumerische Zeichen sind bei SEBLOD ausschließlich Ziffern und einfache Buchstaben. Der Versuch, ein Komma in der Zeichenkette einzugeben, was normalerweise bei alphanumerischen Zeichen erlaubt ist, schlug fehl. Auch werden keine Umlaute akzeptiert.
- *Internet - Url:* Es wird überprüft, ob es sich um eine URL handelt. Zu der URL gehört zwingend das Übertragungsprotokoll. Soweit ich es überprüfen konnte, wird dabei nur auf *http* und *ftp* geprüft.
- *Kontakt - Email:* Es wird geprüft, ob es sich um eine gültige E-Mail-Adresse handelt, wobei aber nur die Schreibweise geprüft wird, nicht, ob es die Mailadresse auch wirklich gibt.
- *Ohne Kategorie - Zusagen:* Das *Ohne Kategorie* hat mich eingangs mächtig verwirrt. Es bezieht sich aber lediglich auf die Auswahlbereiche des Pull-down-Buttons.

Sie können ein anderes Feld angeben (den Aliasnamen vom anderen Feld eintragen). Der Inhalt dieses Feldes muss identisch mit dem anderen Feld sein. Dabei wird zwischen Groß- und Kleinschreibung unterschieden. Bei Textfeldern ist die Überprüfung kein Problem. Der Versuch, mit einem Optionsfeld (Radiobutton) die Überprüfung abzugleichen, schlug bei mir fehl.

Mögliche Einsatzgebiete könnten das Überprüfen der Eingabe einer E-Mail mittels »Bestätigung« sein oder das erneute Eingeben eines Passworts.

In der zweiten Spalte können Sie ein mehrseitiges Formular anlegen. Gerade bei längeren Formularen kann es sinnvoll sein, das Formular in einzelne Abschnitte zu zerlegen, mittels eines Buttons am Seitenende zum nächsten Abschnitt oder zur nächsten Seite umzuschalten und am Ende alles zu speichern.

Sie müssen mehrere Schritte unternehmen, um ein mehrseitiges Formular anzulegen. In meinem Beispiel habe ich neben dem Titel im ersten Abschnitt jeweils ein einfaches Eingabefeld für den zweiten und dritten Abschnitt gesetzt und jeweils einen Button zum Umblättern.

Bild 11.9: Insgesamt drei Abschnitte für dieses Formular.

Der dritte Abschnitt ist der finale Abschnitt – weshalb hier *Ende* ausgewählt werden muss. Die Buttons, die auf die nächste Seite verweisen, sind *Abschicken*-Buttons, die keiner weiteren Einstellung bedürfen. Lediglich *Zeige Alternative* habe ich auf *Verbergen* gesetzt, sodass ich den *Abbrechen*-Link nicht angezeigt bekomme.

Damit das funktioniert, müssen Sie die *Konfiguration* bei den *Optionen* des Inhaltstyps aufrufen.

Bild 11.10: Die Konfiguration im Inhaltstyp.

Etwas weiter unten finden Sie in der *Konfiguration* den Block *Variante* und setzen den Zähler auf 3.

Bild 11.11: Anzahl der Abschnitte einstellen.

- Nach dem Speichern können Sie einen neuen Beitrag mit mehreren Abschnitten anlegen.



**Bild 11.12:** Ein Beitrag über mehrere Abschnitte/Seiten.

### 11.3.2 Intro und Inhalt

Mit *Typografie* ist die Gestaltung des Layouts gemeint, und die bezieht sich in erster Linie auf die Schrift und deren Gestaltung. Schließlich bezieht sich der Begriff ursprünglich auf die Kunst und das Handwerk des Druckens. Demzufolge können Sie an dieser Stelle einige HTML-Vorgaben einstellen, die sich teilweise auf die Schrift, aber auch auf andere Layoutbereiche auswirken.

Wenn Sie eine *Typografie* auswählen, erhalten Sie ein kleines Pluszeichen direkt neben dem Auswahlfeld, mit dem Sie bei einigen Eigenschaften weitere Einstellungen machen können. Im Einzelnen sind das:

- *Löschen*: Inhalt und Feld werden nicht angezeigt.
- *Html*: Wenn Sie das Pluszeichen anklicken, können Sie beliebiges HTML eingeben. Zum Testen habe ich einfach einen Link eingegeben, der anstelle des eigentlichen Inhalts abgedruckt wird:

```
001 <a href="datei.htm">Verweistext</a>
```

Geben Sie kein HTML an, wird der Inhalt des Feldes ganz normal ausgegeben.

- *Fett*, *Kursiv* und *Unterstrichen* geben den Inhalt des Feldes entsprechend aus. Weitere Einstellungsmöglichkeiten sind nicht vorhanden.
- *Überschrift*: Gibt den Inhalt des Feldes als Überschrift aus. Nach dem Klick auf das Pluszeichen können Sie das Überschriftenelement von *H1* bis *H6* eingeben und eine CSS-Klasse angeben.
- *Aktiviere Anker* bewirkt, dass der Inhalt des Feldes – in meinem Beispiel *Toller Text* – als CSS-*id* übergeben wird:

```
001 <h1 id="Toller_Text">Toller Text</h1>
```

- Legen Sie ein Image-Feld an, bei dem Sie ein Bild hochladen können, dann wird das standardmäßig als Textlink angezeigt. Erst durch die Auswahl *Bild* an dieser Stelle wird es auch als Bild angezeigt. Klicken Sie auf das Pluszeichen, können Sie angeben, welches Bild als *Standardanzeige* benutzt werden soll und welches Bild im *modalen Fenster* (Lightbox) nach dem Anklicken des *Standardbildes* angezeigt werden soll. Lassen Sie *Breite/Höhe* auf *Automatisch* stehen, wird die jeweilige

Bildgröße genommen, die Sie ausgewählt haben, andernfalls können Sie die Anzeigegröße vorgeben. Beim *Standardbild* kann das nur eine feste Größe sein, beim *modalen Fenster* wird zwischen *Fest*, *Innen* und *Aussen* unterschieden.

- Zwischen *Fest* und *Innere* ist kein Unterschied festzustellen. Die Modalbox wird jeweils in der maximalen Größe angezeigt und das Bild dort eingepasst.
- *Max*: Die Modalbox passt sich an die Größe des Bildes an bis maximal zur voreingestellten Größe.



**Bild 11.13:** Die breite Modalbox in der Einstellung *Fest* und *Innere*, die angepasste Modalbox in der Einstellung *Max*. Breite x Höhe jeweils 400 px x 200 px.

Der *absolute* oder *relative Pfad* wird direkt in das HTML-*img*-Element eingefügt.

```
001 Absolut: 
003 Relativ:
004 
```

- Zu guter Letzt das *Datum*: Klicken Sie auf das Pluszeichen, können Sie das Datumsformat einstellen. Ich denke, die meisten Einstellungsmöglichkeiten sprechen für sich. Vermutlich vermissen Sie auch das Datum mit dem Punkt, wie wir es hierzulande meistens schreiben. Wählen Sie einfach *Frei* aus und tragen Sie in das dann erscheinende Feld diesen Code ein:

```
001 d.F.Y
```

Dann haben wir unsere Punkte wieder. Details dazu können Sie der Tabelle in Kapitel 8.11.2 entnehmen.

Wenn Sie *vor* auswählen, wird es ein wenig mysteriös. Es wird immer die Zeit bis zum heutigen Tag als vergangene Zeit angezeigt. Wählen Sie ein Datum aus dem letzten Monat aus, erscheint als *Ausgabe 1 monat vorher*. Wählen Sie ein Datum vom kommenden Monat aus, erscheint der gleiche Text. Somit entfällt also die

Möglichkeit, sich einen kleinen Counter oder Ähnliches zu basteln. Aber ich denke, dass in kommenden Versionen dieser Bug, so es denn einer ist, beseitigt werden wird.

In der zweiten Spalte können Sie einstellen, ob die typografischen Angaben auch für die *Beschreibung* benutzt werden sollen. Dann wird beispielsweise auch das Labelfeld fett, kursiv oder als Überschrift ausgegeben.



**Bild 11.14:** Bei der *Beschreibung* wurde die Typografie nicht übernommen.

## 11.4 Nummer 4

Diese Nummer ist in allen Formularen gleich und regelt den Zugriff auf das jeweilige Feld. Wird hier nichts angegeben, gilt stets der *öffentliche Zugriff*. Aber Sie können jeden anderen Zugriff, den Joomla! vorgibt oder den Sie definiert haben, an dieser Stelle pro Feld vorgeben.

Somit können Sie mit einem Inhaltstyp unterschiedlichen Autorengruppen unterschiedliche Felder zugänglich machen oder in der Anzeige bestimmte Inhalte für bestimmte Zugriffsebenen sichtbar machen - ohne dass Sie erst einen neuen Inhaltstyp oder gesonderten Inhalt erstellen müssen.

## 11.5 Nummer 5

Ab hier gibt's nun nur noch zusätzliche Einstellungsmöglichkeiten für das Admin- und das Seitenformular.

Mit *Nummer 5* können Sie Abhängigkeiten und Berechnungen in den Feldern durchführen. Eine Abhängigkeit hatten wir gleich zu Beginn dieses Buches, als wir ein

kleines Bild neben der Newsüberschrift in Abhängigkeit von der ausgewählten Kategorie anzeigt. Aber wie sollte es anders in SEBLOD sein – es gibt natürlich noch viel mehr Möglichkeiten.

### Einstellungen testen

Wenn Sie verschiedene Einstellungen testen wollen, müssen Sie nicht nur den Feldtyp speichern, sondern den ganzen Inhaltstyp, sonst wird die Veränderung nicht angezeigt.

## 11.5.1 Voneinander abhängige Zustände

Ich habe zwei einfache Textfelder angelegt – Feld A und Feld B. Feld B soll abhängig von Feld A sein, weshalb ich auf Feld B klicke und eine Abhängigkeit definiere.

**Bild 11.15:** Abhängigkeiten erzeugen.

Es existieren unterschiedliche Möglichkeiten zur Definition von Abhängigkeiten, sie werden anhand eines Beispiels nun erklärt.

Einstellung *ist sichtbar* / *ist verborgen*:

- Dieses Feld *ist sichtbar*: WENN Feld A identisch ist mit *Apfelkuchen*.  
Erst wenn ich das Wort *Apfelkuchen* in Feld A eingebe, wird Feld B eingeblendet.
- Dieses Feld *ist verborgen*: WENN Feld A identisch ist mit *Apfelkuchen*.  
Hier verhält es sich andersherum: Geben Sie das Wort *Apfelkuchen* in Feld A ein, verschwindet Feld B.
- Sie können bei diesen beiden Status in das zweite Feld *fade* oder *slide* eingeben.  
Damit können Sie entsprechende Effekte für das Ein-/Ausblenden des Feldes festlegen.

Einstellung *ist ausgefüllt* / *ist leer*:

- Wählen Sie *ist ausgefüllt* aus, setzt SEBLOD automatisch ein *#form#* in das dritte Feld. Das gesetzte *#form#* ermittelt die ID für den Datensatz.  
Dieses Feld *ist ausgefüllt* *Birmeneis* *#form#* WENN Feld A identisch ist mit *Apfelkuchen*.

Erklärung: Wenn Sie in Feld A das Wort *Apfelkuchen* schreiben, wird in Feld B das Wort *Birneneis* eingetragen.

- Dieses Feld ist leer *Birneneis* #form# WENN Feld A ist identisch *Apfelkuchen*.  
Erklärung: Wenn Sie in Feld A das Wort *Apfelkuchen* schreiben, wird Feld B geleert, andernfalls steht das Wort *Birneneis* in dem Feld.

Einstellung *ist aktiviert* / *ist deaktiviert*:

- Dieses Feld *ist aktiviert* #form# WENN Feld A ist identisch *Apfelkuchen*.  
Solange Sie nicht *Apfelkuchen* in Feld A eintragen, bleibt das Feld B deaktiviert. Es kann also nur gelesen werden, wenn ein Defaultwert gesetzt ist. Eine Wertänderung ist nicht möglich.
- Dieses Feld *ist deaktiviert* #form# WENN Feld A ist identisch *Apfelkuchen*.  
Solange Sie nicht *Apfelkuchen* in Feld A eingetragen haben, können Sie auf Feld B zugreifen, d. h. das Feld kann beschrieben werden.

CSS-Formatierung für ein Feld beeinflussen:

- Dieses Feld hat Klasse **obst** WENN Feld A ist identisch *Apfelkuchen*.  
Wenn Sie in Feld A das Wort *Apfelkuchen* schreiben, erhält Feld A die CSS-Klasse **obst**. Wobei die Klasse dem umschließenden **div** vergeben wird und somit sowohl das Beschreibungsfeld als auch das Eingabefeld selbst betrifft.

```
001 <div id="cck1r_dev_feld_b" class="cck_forms cck_admin cck_text
    cck_dev_feld_b obst">
```

Sie können somit durch CSS-Formatierung das Feld besonders hervorheben, falls bestimmte Bedingungen auftreten.

- Dieses Feld hat keine Klasse **obst** WENN Feld A ist identisch *Apfelkuchen*.  
Solange Sie den *Apfelkuchen* nicht in Feld A schreiben, hat Feld B die CSS-Klasse **obst** – schreiben Sie jedoch *Apfelkuchen* in Feld A, wird die Klasse gelöscht. Somit können Sie beispielsweise eine Hervorhebung in der CSS-Formatierung nach einer bestimmten Abhängigkeit entfernen.

#### Feld-ID

#form# ermittelt die reale ID des Feldes. In vielen Abhängigkeiten ist es notwendig, dass in der Datenbank entsprechende Einstellungen gemacht werden. Dafür ist die ID des Feldes notwendig. Mit #form# brauchen Sie sich nicht um diese ID zu kümmern – SEBLOD übernimmt das für Sie.

Das Kästchen mit den beiden Zahlen 1 und 2 (wobei sich die 1 eigentlich auf das Feld links daneben bezieht) sorgt dafür, dass die Bedingungen sich in beide Richtungen beeinflussen, wenn es aktiviert ist. Wenn Sie beispielsweise *ist sichtbar* eingestellt haben, wird das Feld nur eingeblendet, wenn die Bedingung rechts erfüllt ist. Ist sie das nicht, wird das Feld ausgeblendet. Wenn Sie jedoch das Kästchen deaktivieren,

bleibt das Feld immer eingeblendet, weil die Bedingung rechts nicht entgegengesetzt mit einbezogen wird in der Abhängigkeit.

Es ist quasi eine Einbahnstraße. Das Feld wird unter bestimmten Abhängigkeiten eingeblendet. Soll es hingegen ausgeblendet werden, ist es erforderlich, die Straße wieder zurückzufahren, um dem Feld mitzuteilen, dass die Bedingung nicht erfüllt ist. Deaktivieren Sie das Kästchen in der Mitte, gibt es kein Zurück. Die Folge ist, dass das Feld niemals ausgeblendet wird.

Sie können für eine Abhängigkeit zwei *Auslöser* angeben, indem Sie unter *WENN* ein *und* oder *oder* hinzufügen und einen zweiten *Auslöser* angeben. Reichen Ihnen zwei Möglichkeiten nicht, können Sie oben rechts auf das kleine unscheinbare grüne Plus klicken – Sie erhalten dann so viele weitere Einstellungsblöcke, wie Sie benötigen. Unten das rote Minus löscht jeweils einen Block.

*Wähle* das Feld aus, das zu dem Inhaltstyp gehört und Auslöser für die Bedingung sein soll. Wenn Sie ein Feld mit mehreren Optionen auswählen, können Sie rechts direkt auf die vorhandenen Inhalte zugreifen, indem Sie auf die orangefarbenen Pfeile klicken. Beim Textfeld erhalten Sie lediglich ein Eingabefeld, aber da können Sie auch direkt in das Feld rechts schreiben.

Bei den *Auslösemöglichkeiten* gibt es einige zur Prüfung der Bedingung:

- *Ist identisch*: Exakte Übereinstimmung der Zeichen. Groß-/Kleinschreibung wird unterschieden (case sensitive).
- *Ist differenziert*: Ungleich zum Wert. Groß-/Kleinschreibung wird auch hier unterschieden (case sensitive). »apfelkuchen« ist also nicht identisch mit »Apfelkuchen«.
- *ist ausgefüllt*: Wenn Sie irgendetwas in das Feld schreiben, ist die Bedingung erfüllt.
- *ist leer*: Wenn nichts im Feld steht, ist die Bedingung erfüllt.
- *changed*: Wenn der Mauscursor aus dem Feld bewegt wird, ist die Bedingung erfüllt.
- *ist pressed*: Wenn der Mauscursor in das Feld gesetzt wird, ist die Bedingung erfüllt.
- *call function*: Wenn Sie mit JavaScript arbeiten, können Sie hier überprüfen, ob eine JavaScript-Funktion aufgerufen wurde. Wenn ja, ist die Bedingung erfüllt.

Wenn Sie alles zurückstellen möchten, können Sie oben auf *Zurücksetzen* klicken. Dann ist alles so wie am Anfang ...

## 11.5.2 Rechenregeln

Leider ist bei meiner Installation die deutsche Übersetzung noch nicht bis hierhin vorgezogen. Ich habe mich deshalb entschlossen, weitgehend die englischen Begriffe zu nennen.

Zum besseren Üben habe ich neben Feld A und Feld B noch ein Feld C eingefügt. Alles sind einfache Textfelder. Das reicht uns zunächst.

In meinem kleinen Beispiel wähle ich unter *Felder* Feld A und Feld B aus. Sie können ein Feld mit dem kleinen roten Minuszeichen jederzeit löschen oder mit dem blauen Punkt umsordieren. Das Umsortieren ist dann wichtig, wenn Sie Berechnungen in der Reihenfolge verändern wollen – Feld A, geteilt durch Feld B ist was anderes als das Ganze umgekehrt.

Im Feld *Format/Präcision* können Sie angeben, was mit dem Ergebnis gemacht werden soll:

- *Auto*: SEBLOD bestimmt, was mit der Zahl geschieht. Meist wird gerundet.
- *Aufrunden*: Es wird stets nach oben gerundet – es wird also die nächstgrößere ganze Zahl ermittelt.
- *Abbrunden*: Es wird die nächstkleinere ganze Zahl ermittelt.
- *Runden*: Die Zahl wird gerundet.
- *toFixed*: Die Anzahl der Nachkommastellen wird festgelegt.

Leider hat dieser Bereich in meiner Installation nicht wirklich funktioniert. Es wurde nichts gerundet, weder auf noch ab.

*Rule*: Mit den *globalen Einstellungen* werden die allgemeingültigen Mathematikregeln verwendet. Bei der Einstellung *Selbst* wird Ihre *benutzerdefinierte* Regel verwendet.

Mit dem *Trigger Event (Auslöse Ereignis)* legen Sie fest, wann die Berechnung durchgeführt wird. *Changed* berechnet erst, wenn Sie das Feld verlassen. Das kann durch die **Tab**-Taste erfolgen, durch **Enter** oder indem Sie in ein anderes Feld klicken. Mit *Taste ab* wird bereits direkt nach der Eingabe berechnet. Wenn Sie eine zweistellige Zahl eintragen, wird auch zunächst die erste Stelle berechnet und dann erst die zweistellige Zahl übernommen. Bei meinen Tests war die *Tasten-Methode* die genauere. Aber bei großen Zahlen kann das auch sehr verwirrend sein und bei komplexen Berechnungen die Performance beeinträchtigen.

### Ergebnisfeld deaktivieren

Es gab bei mir oft Fehler, wenn ich frecherweise das Feld, das das Ergebnis ausgeben sollte (Feld C), mit einem neuen Wert überschrieb. Es kann also sinnvoll sein, das Ergebnisfeld zu deaktivieren, was Sie mit der ersten Spalte, den Abhängigkeiten, erreichen können.

In *Berechnung* stehen Ihnen mehrere Rechenarten zur Verfügung, auf die ich gleich näher eingehen werde.

- *Average*: Ermittelt den Schnitt der Zahlen.
- *Count*: Zählt die Anzahl der Felder, die für die Berechnung zur Verfügung stehen.
- *Format*: Vielleicht muss man Mathematiker sein, um den Sinn dieser Einstellung herauszubekommen ...
- *Max*: Es wird die höchste Zahl ermittelt.
- *Min*: Es wird die niedrigste Zahl ermittelt.
- *Product* (\*): Die Zahlen werden miteinander multipliziert.
- *Sum* (+): Addiert die Zahlen der einzelnen Felder.
- *Concatenate*: Verbindet den Inhalt der Felder als Zeichenkette.  
Feld A: Apfel  
Feld B: kuchen  
Ergebnis in Feld C: Apfelkuchen
- *Benutzerdefiniert*: Zum einen können Sie hier einige fehlende Rechenarten auswählen, zum anderen können Sie eigene Berechnungsformeln eingeben, wenn Sie *Free* auswählen. Die Berechnungen werden in der Reihenfolge der ausgewählten Felder ausgeführt. Das erste Feld (in meinem Beispiel Feld A) entspricht dem »a«, das zweite Feld (Feld B), dem »b« und so weiter.

Wenn Sie neben einem ausgewählten Feld *Custom Attributes* (*benutzerdefinierte Attribute*) aktivieren, können Sie auch mit Auswahlfeldern arbeiten. Es wird dann entweder der Value berechnet, oder Sie haben einen angegeben, der übernommen wird. Haben Sie beispielsweise ein einfaches Auswahlfeld erstellt mit den beiden Optionen

- Ja = 1
- Nein = 2

wird bei der Berechnung die Value-Zahl übernommen.

Sie können aber bei diesen Auswahlfeldern auch eigene Attribute anlegen und diese wiederum namentlich angeben. Dann wird in der Berechnung der Inhalt Ihrer Attribute berücksichtigt.

## 11.6 Nummer 6

Sie können hier CSS-Klassen eintragen. Das können sowohl eigene sein – also CSS-Klassen, die Sie formatieren – als auch vorgegebene Klassen. Die vorgegebenen finden Sie in Anhang A.

## 11.7 Der Punkt

Der Punkt ist ein An- und Ausschalter und kann bei jeder Nummer ein- bzw. ausgeblendet werden. Wenn Sie den *Punkt* einschalten, dann sehen Sie jeweils in den einzelnen Positionsabschnitten ein *Breite x Höhe*-Feld anstelle des Feldes, das der Nummer entsprechend dort vorher stand. Aus diesem Grund sollten Sie nicht vergessen, den *Punkt* auch wieder auszuschalten, damit Sie vollen Zugriff auf die einzelnen Nummernfelder haben. Sie können also bei den einzelnen Abschnitten die Größe variieren, indem Sie Breite und Höhe jeweils vorgeben. Alternativ können Sie auch nur die Breite oder nur die Höhe angeben.

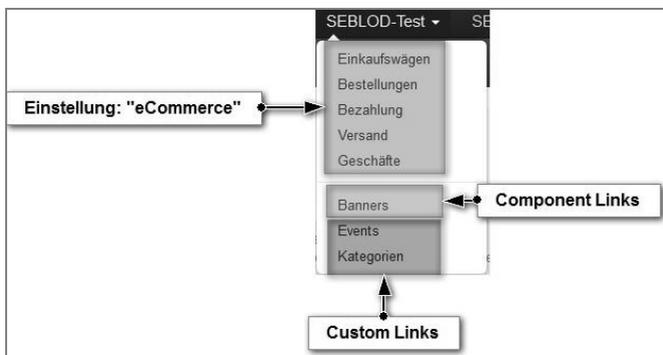
Insbesondere dann, wenn Sie mehrere Bereiche nebeneinander anzeigen möchten, sollten Sie hier eventuell etwas eintragen und somit Ihre Ausgabe individuell gestalten.

# 12

## Das eigene Adminmenü

Eine schöne Möglichkeit, etwas mehr Überblick über einige Erweiterungen oder seine SEBLOD-Apps und -Listen zu bekommen, ist das eigene SEBLOD-Adminmenü. Sie haben hier auch die Möglichkeit, sich eine Art »Schnellstartmenü« anzulegen.

Wenn Sie benutzerdefinierte Menüverweise definiert haben, bleiben diese auch bestehen, wenn Sie nicht *benutzerspezifisch* auswählen. Sie erhalten dann diese benutzerdefinierten Verweise stets am Ende des Menüs angezeigt.



**Bild 12.1:** Ein eigenes Adminmenü mit SEBLOD.

Wechseln Sie dazu zunächst in die Modulübersicht von Joomla! und dort in die Adminmodule.

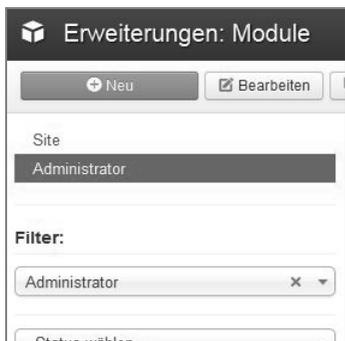


Bild 12.2: Module im Backend.

Über *Neu* können Sie das *SEBLOD-Administrationsmenü* aufrufen. Sie haben nun verschiedene Möglichkeiten, Inhalte in das Menü zu bekommen. Alle Modi haben einige Einstellungen gemeinsam:

Der *Menütitel* wird in der Menüleiste angezeigt, einerlei, wie Sie *Titel anzeigen* für das Modul eingestellt haben.

Mit der *Position* geben Sie an, wo genau das Menü angezeigt wird. *Auto* und *Left* waren bei mir allerdings identisch. Mit *Right* wird das Menü rechts außen direkt rechts neben dem »Zahnrad« angezeigt.

Alles andere wird entweder nachfolgend erklärt oder ist identisch mit den Joomla!-spezifischen Einstellungen.

Falls es nicht ohnehin bereits voreingestellt ist, wählen Sie als *Modulposition: menu*.

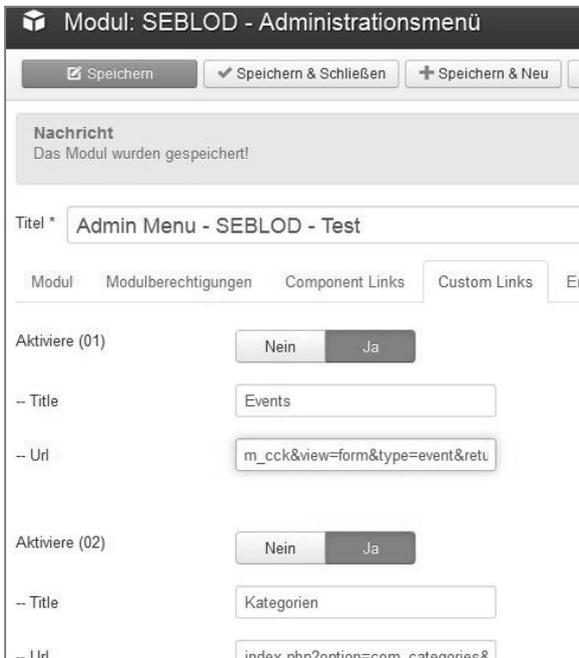
## 12.1 Add-ons

Wenn Sie Add-ons für SEBLOD installiert haben, können Sie sie immer im Komponentenmenü unter *SEBLOD 3.x* aufrufen. Oder Sie legen sich hiermit ein eigenes Menü für diese Add-ons an und können Sie so direkt aufrufen

## 12.2 Benutzerspezifisch

Wenn Sie diese Einstellung auswählen, klicken Sie auf den Tab *Custom Links*.

Sie können nun bis zu zehn verschiedene eigene Verweise angeben und die Menüs selbst benennen. So habe ich beispielsweise einen direkten Aufruf angegeben, um ein neues Event zu erstellen:



**Bild 12.3:** Benutzerdefinierte Menüverweise.

Die URLs bekommen Sie heraus, indem Sie einmal den Vorgang regulär aufrufen und dann die URL des Browsers kopieren.

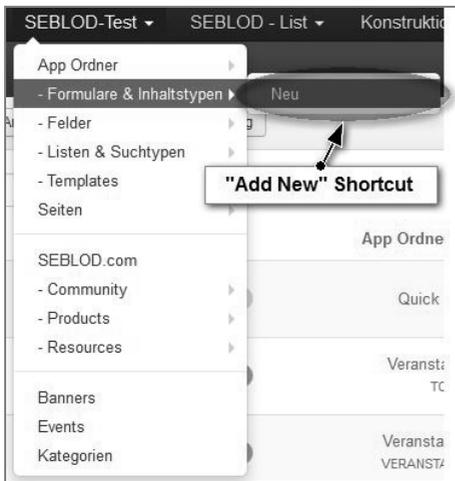
Sie können zusätzlich oder stattdessen auch direkt Komponenten aufrufen. Dazu einfach den Tab *Component Links* anklicken und nach dem *Aktivieren* die Komponente der Wahl aufrufen. So können Sie bis zu fünf Komponenten mit ihren Untereinträgen aufrufen.

Natürlich kommen Sie auch stets über das Joomla!-Menü *Komponenten* zu selbigen. Aber falls Sie die eine oder andere Komponente haben, auf die Sie öfter zugreifen, ist das eine sehr schöne Möglichkeit, es mit dem SEBLOD-Administrationsmenü zu tun.

## 12.3 Konstruktion

Diese Einstellung ist identisch mit dem bereits vorhanden SEBLOD-Menü.

Sie können allerdings im Modul einstellen, ob ein »Add New« *Shortcut* hinzugefügt werden soll. Wählen Sie *Anzeigen* aus, erhalten Sie direkt neben den einzelnen SEBLOD-Menüverweisen die Möglichkeit, *Neu* auszuwählen.



**Bild 12.4:** Add New-Shortcut in der Konstruktion.

Wenn Sie das generell möchten, empfehle ich, das bereits vorhandene Modul zu editieren und entsprechend anzupassen.

## 12.4 E-Commerce

Damit können Sie ein Menü erstellen, das Ihnen Verweise zu verschiedenen Bereichen eines Shopsystems gibt. Leider habe ich die dafür notwendige Komponente nirgends finden können und vermute, dass es sich entweder um eine für die Zukunft geplante Entwicklung handelt oder dass das SEBLOD-Team davon ausgeht, dass Sie selber so etwas programmieren können. Zumindest führt das Anlegen solch eines Menüs zu der Mitteilung, dass die Komponente `com_cck_ecommerce` nicht vorhanden ist, und der Aufruf eines Menüverweises zu einer 404-Statusmeldung.

## 12.5 Formulare

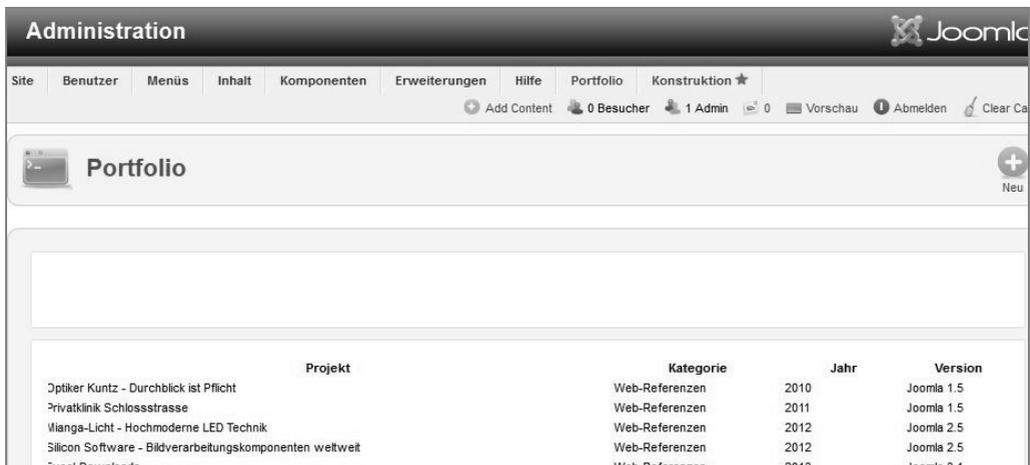
Sie erhalten eine Liste der angelegten Formulare als Direktaufruf im Menü und wechseln bei Aufruf direkt in *Neues hinzufügen* des jeweiligen Formulars.

Haben Sie mehr Formulare, als Sie in der Liste darstellen möchten bzw. benötigen, setzen Sie Formulare in der Übersicht von SEBLOD auf *Eintrag verstecken* im Status. Wenn Sie das allerdings mit dem *Article* von SEBLOD machen, dann können Sie nach wie vor mit Anklicken von *Neu* einen neuen Beitrag aus der Beitragsübersicht auswählen, jedoch nicht auf dem kurzen Weg über das Pull-down-Menü *Inhalt/Beiträge/Neuer Beitrag*. Dort erhalten Sie dann die Meldung, dass keine Felder vorhanden sind.

## 12.6 Lists

Sie erhalten eine Liste der angelegten *Listen & Suchtypen* zum direkten Aufrufen. Die Liste wird so angezeigt, wie Sie sie definiert haben. Falls Sie eine Suche dazu definiert haben, wird sie ebenfalls angezeigt. Zudem gibt es einen *Neu*-Button, und Sie können aus der Liste heraus direkt einen neuen Beitrag verfassen.

Im nachfolgenden Screenshot sehen Sie einen Ausschnitt aus meinem Portfolio, das ich für Joomla! 2.5 mit SEBLOD erstellt habe.



**Bild 12.5:** Portfolioliste im Backend (Joomla! 2.5). Bei mir hat übrigens die Sortierung nach Jahreszahl nicht funktioniert.

Sie haben beim Menümodul die Möglichkeit, im Tab *Modul* weiter unten einzustellen, ob das Menü *Inline* angezeigt werden soll. Der übliche Weg ist, dass es als Pull-down-Menü angezeigt wird. Sie können aber auch explizit dafür sorgen, dass der Eintrag nicht als »Aufklappmenü« erscheint, sondern dass die Inhalte direkt in der Menüleiste angezeigt werden. Das ist bei sehr wenigen Suchlisten vielleicht sinnvoll, führt jedoch bei vielen Suchlisten schnell zu einer unübersichtlichen Vergrößerung der Menüleiste.

## 12.7 Fazit zum Administrationsmenü

Was zunächst wie eine nette kleine Spielerei aussieht, kann bei größeren Webseiten, die vielleicht auch für Kunden erstellt sind, für einen schnellen Zugriff auf speziell erstellte Listen, Beiträge, Kategorien und User sorgen. Gerade dem Kunden, der eventuell nicht die Sicherheit in Joomla! hat, um alles Benötigte ohne Sucherei sofort zu finden, wird hier eine immense Arbeitserleichterung an die Hand gegeben werden.

Ich habe bereits Webseiten gesehen, die eine Vielzahl von SEBLOD-Eingabemasken hatten, und durch ein entsprechendes Menü wurde vieles erleichtert.

# **Teil 3 - Der praktische Einsatz**

Was mit SEBLOD möglich ist und wie man es vielfältig einsetzen kann, ist immer noch am besten erklärt, wenn man es in der Praxis anwendet. In diesem dritten Teil geht es mir weniger um fertige Lösungen, die Sie nur noch herunterladen und zu installieren brauchen, um dann fertige Erweiterungen zu haben, sondern es geht mir darum, anhand einfacher Beispiele einige Möglichkeiten aufzuzeigen. Sie werden aber ohne nennenswerte Schwierigkeiten die Beispiele an Ihre Bedürfnisse anpassen können.





## Veranstungskalender

Die Überlegung ist, dass ich mehrere Städte habe, in diesen Städten Veranstaltungslöke sind und Künstler dort auftreten. Meine Website soll über die Künstler berichten, weshalb die Künstlerbeiträge als Joomla!-Beiträge abgespeichert werden. Sowohl die Stadtauswahl als auch die einzelnen Veranstalter werden lediglich in der Datenbank gespeichert, sollen aber als Auswahllisten in den Künstlerbeiträgen zur Verfügung stehen.

Dazu benötige ich drei Inhaltstypen:

- Eingabe der Städte, die anschließend dynamisch ausgewählt werden.
- Eingabe der Veranstalter, die bei den Events ebenfalls dynamisch ausgewählt werden.
- Das Event.

Alles kommt in die App *Veranstaltungen*.

	Titel	App Ordner	# Admin Formular	# Seitenformular	# Intro	# Inhalt
	Event event	Veranstaltungen TOP	3	-	2	3
	Veranstalter veranstaltungsort	Veranstaltungen TOP	3	-	-	-
	Veranstaltungsstädte eingeben veranstaltungen	Veranstaltungen TOP	1	-	-	-

**Bild 13.1:** Drei Inhaltstypen für die *Veranstaltungen*.

In der Abbildung sehen Sie, dass ich lediglich dem Inhaltstyp *Event* Ausgaben im Frontend gebe, die beiden anderen Inhaltstypen bekommen nur ein Adminformular.

## 13.1 Die Städte

Die Städte möchte ich als Auswahlliste bei den Veranstaltern haben. Später will ich eine Suchliste erstellen und mir so alle Veranstaltungsorte in einer Stadt anzeigen lassen. Dazu erstelle ich zuerst eine neue Tabelle in meiner Datenbank. Dabei gehe ich so vor, wie in Kapitel 4.8 »Eigene Tabelle erstellen« beschrieben. Neben der ID lege ich eine Spalte mit dem Namen *stadt* an und nenne die Tabelle *#\_\_veranstaltungen*.

Nun wird ein neuer Inhaltstyp angelegt, der App *Veranstaltungen* zugeordnet und ein Feld von Typ *Text* im *Admin Formular* angelegt. Im Speicherbereich wähle ich zunächst *Standard* aus und dann *Frei*, damit ich meine gerade angelegte Datenbanktabelle und die Tabellenspalte *stadt* auswählen kann.

**Speicher <sup>2</sup>**

Format / Position    Standard ▾    Frei ▾    fj511\_veranstaltungen ▾

stadt    «     Wechseln

**Bild 13.2:** Speicherung des Textfeldes.

Das war es bereits, um die Städte einzugeben. Wenn Sie nun in die Beiträge gehen und dieses Formular aufrufen, sehen Sie ein unscheinbares Eingabefeld. Völlig unspektakulär. Dadurch, dass wir aber in einer eigenen Tabelle speichern, erscheinen die einzelnen Städte nicht in der Beitragsübersicht.

## 13.2 Veranstalter

Für die Veranstalter lege ich wieder eine neue Datenbanktabelle an, diesmal aber neben der ID mit drei weiteren Spalten:

- *id*
- *stadt*
- *veranstaltungsname*
- *anschrift*

+ Optionen										
← T →			id	veranstaltungsname	stadt	anschrift				
<input type="checkbox"/>		Bearbeiten		Kopieren		Löschen	10	Pavillon	Hannover	Weiß-Kreuz-Platz
<input type="checkbox"/>		Bearbeiten		Kopieren		Löschen	11	Oper	Hannover	Opernplatz
<input type="checkbox"/>		Bearbeiten		Kopieren		Löschen	12	Theater am Aegi	Hannover	Aegidientorplatz
<input type="checkbox"/>		Bearbeiten		Kopieren		Löschen	13	Tollhaus	Karlsruhe	

**Bild 13.3:** Die Datenbanktabelle mit einigen Einträgen.

Die drei weiteren Spalten sind alle als *Text* angelegt. Genannt habe ich die Tabelle `#__veranstalter`.

Im *Admin Formular* erstelle ich nun drei neue Felder. *Name* und *Anschrift* sind einfache Textfelder, wobei Sie die Anschrift auch als *Textarea* anlegen können. Alle drei Felder speichern Sie als *Standard* und *Frei* in die neu angelegte Tabelle.

Die *Stadt* lege ich vom *Typ: Dynamisch* an. Damit kann ich auf die Datenbanktabelle mit den Städten zugreifen und diese als Auswahlliste anzeigen.

Lassen Sie die *Query* auf *Konstruktion* stehen und tragen Sie in *Table: #\_\_veranstaltungen* ein – genau so, wie ich es hier geschrieben habe, also mit `#__` (dem doppelten Unterstrich). In *Optionsname* kommt der Spaltenname, dessen Inhalte Sie angezeigt bekommen möchten, und in *Optionswert*, welcher Inhalt abgespeichert werden soll. In unserem Fall ist das sehr einfach, weil wir nur eine Tabellenspalte haben, die wir gebrauchen können: die Spalte mit den Städten.

Gemacht ist das so, weil es ja gut sein könnte, dass Sie neben dem Stadtnamen auch noch die PLZ haben. Dann können Sie die Stadt beispielsweise per Auswahlliste anzeigen, die zugehörige PLZ aber beim Veranstalter speichern. Näheres zu diesem Feldtyp erhalten Sie im Kapitel 8 »Alle Felder«. In unserem Fall kommt also in beiden Feldern *stadt* hinein.

Alternativ können Sie auch bei *Query* auf *Frei* umschalten und die SQL-Syntax direkt eingeben:

```
001 SELECT stadt FROM fj51l_veranstaltungen
```

Dann müssen Sie aber das Präfix Ihrer Datenbanktabellen mit angeben. In der *Konstruktion* brauchen Sie das nicht.

Wenn Sie nun in der Beitragsübersicht einen neuen Veranstalter eingeben wollen, erhalten Sie eine dynamische Auswahlliste der bisher eingetragenen Städte.

**Bild 13.4:** Dynamische Auswahllisten.

Auch diese Einträge werden nicht als Joomla!-Beiträge gespeichert, weil wir sie in einer eigenen Datenbanktabelle abgespeichert haben.

### 13.3 Das Event

Das eigentliche Event wird nun endlich unser Beitrag, den wir entsprechend ausgeben können. In diesem kleinen Beispiel mache ich es mir einfach und lege lediglich den Künstler und das Auftrittsdatum an. Den Veranstaltungsort hole ich mir wieder als *dynamisches Feld*, allerdings diesmal von der Datenbanktabelle `#_veranstalter`. Der *Optionsname* ist der *veranstaltername*, und diesmal speichere ich die *id* ab, die ich im *Optionenwert* eintrage.

Query	Konstruktion ▾	Wähle Bezeichnung	Wähle einen Veranstaltungsort
Table	#_veranstalter	Optionsname	veranstaltername
Wo		Optionenwert	id

**Bild 13.5:** Die *id* wird abgespeichert.

Das mache ich deswegen, weil ich in der Ausgabe nicht nur auf den Veranstalternamen, sondern auch auf die anderen Spalten zugreifen möchte.

*Veranstaltungsort* und *Datum* speichere ich jeweils *benutzerdefiniert* im *introtext* des *Artikels*. Den *Künstlernamen* speichere ich als *Standard* im *Artikel* und in der Spalte *title* ab. Damit kann ich meine einzelnen Events optimal in der Beitragsübersicht wiederfinden. Das *Intro* und den *Inhalt* gestalten Sie nach eigenem Gusto.

Rufen Sie nun einen Beitrag im Frontend auf, dann erhalten Sie zwar die ID des Veranstaltungsorts, aber mehr leider nicht. Um Stadt, Anschrift und Name des Veranstaltungsorts anzuzeigen, benötigen wir ein Plug-in, das leider nicht standardmäßig bei SEBLOD dabei ist. Aber unter der Product-Site<sup>29</sup> auf SEBLOD gibt es das Plug-in *SD Databaser* zum kostenlosen Download.

Mit dem *SD Databaser* können Sie Ausgaben von Datenbanktabellenzellen machen, auf die Ihr Beitrag zwar verweist, die aber nicht direkt ausgegeben werden können. Also genau das, was wir vorliegen haben: Wir haben eine ID gespeichert, die auf einen Datensatz verweist, den wir aber nicht direkt anzeigen können.

Das Plug-in installieren Sie so, wie in Joomla! üblich - also über *Erweiterungen/Erweiterungen/Paketinstallation*. Eventuell müssen Sie das Plug-in in der Plug-in-Übersicht von Joomla! noch aktivieren. Wechseln Sie nun zu Ihrem *Formular & Inhaltstyp* und rufen Sie den *Inhalt* Ihres Eventformulars auf (wahlweise das *Intro*). Klicken Sie hier die *Nummer 3* an. Damit wechseln Sie in die *Typografie*. Bei dem Feld für den Veranstaltungsort wählen Sie in der Spalte *Typografie* ganz unten *SD Databaser* aus. Sie erhalten nun ein kleines Pluszeichen, das Sie bitte anklicken.



**Bild 13.6:** Auswahl des *SD Databaser*.

Das Plug-in ist eines der wenigen Plug-ins auf SEBLOD, das wirklich sehr gut dokumentiert ist. Das Video beim Download dieses Plug-ins zeigt einige Möglichkeiten auf. Bei mir sieht leider das Plug-in nicht so hübsch aus wie auf dem Video, erfüllt aber dennoch seinen Zweck.

Wir benötigen den Veranstalternamen, die Stadt des Veranstaltungsorts und die Anschrift. Somit müssen wir zuerst eine SQL-Anweisung formulieren und in das Feld *SQL Query* eingeben. In unserem Fall lautet sie wie folgt:

```
001 select veranstaltername, stadt, anschrift from #__veranstalter where
    id=*value*
```

<sup>29</sup> <http://www.seblod.com/products/2239>

Sie können hier mit #\_\_ arbeiten. Die Datenbanktabelle hatten wir zuvor selber angelegt. Übergeben müssen wir die ID, die wir gespeichert haben. Das tun wir mit der Variablen `value`, die in Sternchen eingefasst werden muss. Wenn Sie das speichern und ins Frontend zurückkehren, werden Sie leider immer noch unsere abgespeicherte ID sehen. Aber in *Column(s) AS Value* tragen wir nun endlich ein, welches oder welche Felder ausgegeben werden sollen, in unserem Fall also die drei Felder, die auch in der SQL-Abfrage bereits genannt sind. Allerdings müssen wir die drei Felder mit zwei senkrechten Strichen trennen (|). Sie kommen an die senkrechten Striche, indem Sie auf der Windows-Tastatur `AltGr` + `↵` drücken. Der korrekte Eintrag lautet:

```
001 veranstaltername||stadt||anschrift
```

Damit die einzelnen Daten untereinanderstehen, tragen Sie in *Column(s) Separator* einen HTML-Zeilenumbruch ein:

```
001 <br/>
```

*Multiple Row Separator* benötigen Sie, wenn Sie mehrere Datensätze ausgeben und die einzelnen Datensätze optisch voneinander trennen möchten. Beispielsweise könnten Sie unsere Veranstalterdaten auch mit einem Komma trennen, indem Sie als *Column(s) Separator* ein Komma mit einem nachfolgenden Leerzeichen eintragen und in *Multiple Row Separator* den HTML-Zeilenumbruch. Zum Testen ändern Sie die SQL-Abfrage am Ende auf `id >= 1`:

```
001 select veranstaltername, stadt, anschrift from #__veranstalter where
id>=1
```

Vorausgesetzt, Sie haben mindestens zwei Veranstalter eingetragen, werden diese auch entsprechend ausgegeben.

Der zerbrochene	
Veranstaltungsort 1	Veranstaltungsort 2
Veranstaltungsort	Pavillon, Hannover, Weiße-Kreuz-Platz
Oper, Hannover, Opernplatz	Theater am Aegi, Hannover, Aegidientorplatz
Tollhaus, Karlsruhe,	und so weiter...
Hotel Ehrwürdig, Braunschweig, Platz der Großen	
Künstler:	Der zerbrochene Krug
Auftrittsdatum	2013-11-29
Kategorie	Apfelkuchen

**Bild 13.7:** Mehrere Veranstaltungsorte mit dem *SD Databaser* ausgeben.

**Fazit:** Ein kleines Beispiel, wie Sie mit mehreren Tabellen Ihre Daten sauber trennen und am Ende wieder zusammenfügen können. Einziger Nachteil bei dieser Methode: Sie müssen sich ein wenig mit SQL vertraut machen. SQL selber ist aber einfach zu erlernen, und für unsere Belange reicht Ihnen das Grundwissen, um alles Benötigte umsetzen zu können.



## Portfolio

Auf meiner Firmenseite *time4mambo.de* brauchte ich eine Referenzliste. Die angebotenen Erweiterungen gefielen mir nicht, also habe ich mir eine eigene, nach meinen Bedürfnissen, mit SEBLOD gemacht. Zunächst war mir ein schneller Überblick wichtig, in dem neben einer Grafik eine Kurzinformation zur verwendeten Joomla!-Version und dem Erstellungsdatum angezeigt wird, um eine gewisse Kontinuität aufzuzeigen. Dann sollte *Weiterlesen* eingeblendet werden, um bei Interesse weitere Details sowie mehrere Screenshots aufzuzeigen.

## sprich Charisma - Körpersprache mit Köpfchen

[Zurück zur Übersicht](#)



Jahr: 2013  
Version: Joomla 2.5

---

**sprich Charisma - Körpersprache mit Köpfchen**

Um Körpersprache darstellen zu können, bedarf es unter anderem Bilder. Und die sollten groß und aussagend sein. Also wurde ein Slider in den Hintergrund gelegt, der sich der Browserauflösung des Besuchers anpasst. Dazu noch eine umfangreiche Galerie und Text sowieso.

Abgerundet mit einem Buchungssystem für Seminare

---

**Integrierte Erweiterungen:**

- Seminar Manager
- Integration und Anpassung eines Templates von *Joomspirit*
- Die Kommentarfunktion von Komento war eine kleine Herausforderung, weil nicht all zuviel Platz zur Verfügung stand. Insbesondere für das Modul "der letzten Kommentare"
- Das Zusammensuchen und Anpassen der Bilder sieht deutlich einfacher aus, als es in Wirklichkeit war.

Link zur Website: <http://www.sprich-charisma.de>

< Zurück
Weiter >

**Bild 14.1:** Detailansicht des Portfolios.

Eine Schwierigkeit war der *Zurück zur Übersicht*-Verweis, da ich zwei Kategorien habe, Webdesign und Programmierung, und diese beiden Bereiche im Portfolio trennen wollte. Also zwei verschiedene Menüverweise jeweils als Kategorieblog.

Doch der Reihe nach:

Eine neue App mit dem Titel *Portfolio*. Alle *Elemente* und sowohl *App Root* als auch *Haupteintrag* auf *Ja*.

Entnehmen Sie die Felder der Abbildung:

Konstruktion <small>(für Admin Formularansicht)</small>	
<input type="radio"/>	# MAINBODY
<input type="checkbox"/>	Kurztext <small>(Textbereich)</small>
<input type="checkbox"/>	Projektbeschreibung <small>(WYSIWYG-Editor)</small>
<input type="checkbox"/>	Technik <small>(WYSIWYG-Editor)</small>
<input type="radio"/>	# TOPBODY
<input type="checkbox"/>	Projektkategorie <small>(JForm - Kategorie)</small>
<input type="checkbox"/>	Version <small>(Wähle - Einfach)</small>
<input type="checkbox"/>	Jahr <small>(Text)</small>
<input type="radio"/>	# BOTTOMBODY
<input type="checkbox"/>	Link zur Site <small>(Link)</small>
<input type="checkbox"/>	Projektbilder <small>(Feld X)</small>
<input type="radio"/>	# TOP-A
<input type="checkbox"/>	Introbild <small>(Hochladen - Bild)</small>
<input type="radio"/>	# TOP-B
<input type="checkbox"/>	Projekttitle <small>(Text)</small>
<input type="radio"/>	# BOTTOM-A
<input type="checkbox"/>	Zurück-Link <small>(Link)</small>
<input type="checkbox"/>	Oder-Zurück-Link <small>(Link)</small>
<input type="radio"/>	# SIDEBODY-A

Bild 14.2: Die Felder im Adminformular.

Die Felder für die Versionen sind mit den gängigen Joomla!-Versionen vorbelegt. Für das Jahr habe ich ein Textfeld gewählt, da mir die Eingabe der Jahreszahl mit dem Datumsfeld zu umständlich erschien. Für die Projektbilder kommt ein *Feld X* zur Anwendung, um beliebig viele Screenshots eingeben zu können, wobei ich den *Standard* auf 5, *Maximum* auf 10 und *Minimum* auf 0 gesetzt habe. Nachfolgend eine Detailbeschreibung der Felder. Bis auf die Felder *Projekttitle* und *Projektkategorie* wurden die Felder als *Benutzerdefiniert - Article - introtext* gespeichert.

- **Kurztext** - *Textbereich* - Spalten: 30, Reihen: 10, Mindestlänge: 0, Maximallänge: 255, Neue Zeilen: Absätze (?), Lösche Leerzeilen: Nein

Mit diesem Feld gebe ich den Text im Intro an, also einen kurzen Übersichtstext. Der sollte nicht weiter gestaltet werden können, weshalb ich auf den WYSIWYG-Editor verzichtet habe.

- **Projektbeschreibung** - *WYSIWYG-Editor* - Ort: Standard, Breite x Höhe: 600 x 400, Importierer: Nein  
Dieser Text wird dann in der Detailansicht (*Inhalt*) angezeigt. Da ich eventuell einmal etwas fett oder eine Aufzählung machen möchte, brauche ich einen Editor.
- **Technik** - *WYSIWYG-Editor* - wie *Projektbeschreibung*.  
Meine Gedanken dazu sind, dass ich einen Absatz machen wollte, in dem installierte Erweiterungen unter dem eigentlichen Beschreibungstext angezeigt werden sollen. Damit erreiche ich bei jedem Projekt ein einheitliches Aussehen.
- **Projektkategorie** - *Kategorie*  
**Speicherung** - *Standard* - *Artikel* - *catid*
- **Version** - *Einfachauswahl* - Standard Wert: Joomla! 3.2, Optionen: [Die jeweiligen Joomla!-Versionen]
- **Jahr** - *Text*
- **Link zur Site** - *Link* - Zeige Text: Verberge, Zeige Ziel: Zeige, Standard Ziel: \_blank, Zeige Klasse: Verberge, Zeige Vorschau: Zeige, Größe: 32, Mindestlänge: 0, Maximallänge: 255
- **Projektbilder** - *Feld X* - Feld (Formular): einzelbild, Standard: 5, Maximum: 10, Minimum: 0, alles andere auf *Ja*
- **Einzelbild** (wird nicht im Formular angezeigt) - *Bild* - Ordner: images/portfolio/, 1 Ordner / Inhalt: Nein, 1 Ordner / Benutzer: Nein, Erlaubte Erweiterungen: Image, Maximalgröße: 5 MB, Zeige Vorschau: Miniaturbild 1, Zeige Löschen-Kästchen: Zeige, Zeige spezifischen Pfad: Verberge, Größe: 32, Erweiterter Modus: Nein
- **Miniaturbilderstellung**: Beim Hochladen  
Anzeige als Standard: Miniaturbild 1  
Bild: Original  
Miniaturbild 1: Max Anpassung - 150 x 150 px  
Miniaturbild 2: Max Anpassung - 1024 x 800 px  
Miniaturbild 3: Max Anpassung - 400 x 250 px
- **Introbild** - *Bild* - Ordner: images/portfolio/introbilder/, 1 Ordner / Inhalt: Ja, 1 Ordner / Benutzer: Nein, Erlaubte Erweiterungen: Image, Maximalgröße: 5 MB, Zeige Vorschau: Miniaturbild 5, Zeige Löschen-Kästchen: Zeige, Zeige spezifischen Pfad: Verberge, Größe: 32, Erweiterter Modus: Ja (JSON-formatierter Wert), Zeige Titel: Titel, Zeige »alternative« Beschreibung: Zeige alt-attribut

- **Miniaturbilderstellung:** Immer, wenn nicht vorhanden

Anzeige als Standard: Miniaturbild 2

Bild: Original

Miniaturbild 1: Max Anpassung - 150 x 150 px

Miniaturbild 3: Max Anpassung - 1000 x 800 px

Miniaturbild 5: Max Anpassung - 300 x 200 px

- **Projekttitlel** - *Text*

**Speicherung** - *Standard - Artikel - title*

- **Zurück-Link** - *Link* - Alles auf *Verberge* gesetzt, Standard Text: Zurück zur Übersicht, Standard Ziel: *\_self*, Größe: 70, Mindestlänge: 0, Maximallänge: 255

- **Oder-Zurück-Link** - Exakt gleiche Einstellung wie der »Zurück-Link«, einschließlich des gleichen Standardtextes. Erläuterung etwas weiter unten.

Da ich als Einziger für das Portfolio zuständig bin, gibt es nur ein *Admin Formular*, ein *Seitenformular* wird nicht benötigt.

Das *Intro* umfasst die Felder: Kurzttext, Jahr, Version und Introbild.

Der *Inhalt* erhält die Felder: Projektbeschreibung, Technik, Link zur Site, Jahr, Version, Zurück-Link, Oder-Zurück-Link und Projektbilder.

Verlinkt wird das Ganze als Kategorieblog. Und an dieser Stelle wurde ein Problem sichtbar: Ich wollte natürlich beim *Zurück zur Übersicht* automatisch die richtige Verlinkung haben. Ergo musste ich SEBLOD kundtun: Wenn in der Kategorie *Webdesign* ausgewählt ist, nimm diese Übersichts-URL, und wenn die Kategorie *Programmierung* ausgewählt ist, nimm die andere Übersichts-URL.

Eigentlich sollte das mit den abhängigen Feldern kein Problem darstellen. Doch leider kann keine SONST-Bedingung definiert werden. Also: WENN das, DANN nimm dieses, SONST das - ist etwas, was nicht funktioniert in SEBLOD. Also blieb nur, zwei Felder zu nehmen und bei beiden Feldern eine Abhängigkeit auf die Kategorie einzustellen.

Das eine Feld prüft, ob die Kategorie *Webdesign* heißt. Wenn ja, wird eine bestimmte URL in das Textfeld geschrieben. Wenn nicht, wird es einfach leer gelassen. Das andere Feld macht das Gleiche, prüft lediglich die Kategorie auf *Programmierung* ab. Glücklicherweise werden leere Felder nicht angezeigt, somit wird nur der eine korrekte Link zur Übersicht ausgegeben.

Um das alles einzustellen, klicken Sie im *Admin Formular* auf die *Nummer 5* und klicken beim *Zurück-Link* auf *Hinzufügen*. Geben Sie nun Folgendes ein:

*Dieses Feld ist ausgefüllt [Ihr Link zu Ihrer Übersicht] #form# [Haken gesetzt] WENN Projektkategorie is Equal 43*

43 ist bei mir die Kategorie-ID für *Programmierung*. Bei Ihnen wird das vermutlich abweichen. Klicken Sie der Einfachheit halber die orangefarbenen Pfeile an und wählen Sie die passende Kategorie aus.

Exakt das Gleiche müssen Sie bei dem *Oder-Zurück-Link* noch einmal machen, allerdings mit der Kategorie-ID für das *Webdesign*.

**Bild 14.3:** Die richtige URL für den Zurück-Link.

Ich habe beide Felder im *Admin Formular* auf *Nur Leserechte* gesetzt, da die URLs letztlich fest vorgegeben sind und ich mich darum auch nicht mehr kümmern will.

**Bild 14.4:** Wird die Kategorie *Programmierung* ausgewählt, erhält das obere Feld den passenden Link, und das untere bleibt leer. Bei der Kategorie *Webdesign* ist es umgekehrt. Im Frontend wird nur das jeweils beschriebene Feld ausgegeben.

## 14.1 Gestaltung

Im *Intro* habe ich zunächst beim *Jahr* die *Nummer 3* ausgewählt und dann in der *Typographie Kursiv*. Beim *Introbild* wird die *Typographie Bild* ausgewählt. Hier klicken Sie bitte auf das kleine Pluszeichen und stellen Folgendes ein:

- Anzeige als Standard: Miniaturbild 5 – Breite/Höhe: Automatisch
- Modales Fenster: Miniaturbild 3 – Breite/Höhe: 1000 x 800 px
- Pfad: Relative

Das *Intro* ist der erste Überblick, den Sie sehen, wenn Sie den Link direkt aufrufen. Das Bild können Sie anklicken, und es öffnet sich in einem modalen Fenster, dazu ein Button, mit dem Sie in die Detailansicht wechseln können. *WEITERLESEN* ist eine Joomla!-Eigenschaft. Eventuell müssen Sie die Joomla!-Optionen bei den Beiträgen oder den Menüverweis entsprechend einstellen, sollte dieser nicht angezeigt werden.



**Bild 14.5:** Das *Intro* mit anklickbarem Bild, das sich im modalen Fenster öffnet (Lightbox), und einem *WEITERLESEN*-Button, um in die Detailansicht zu kommen.

Im *Inhalt* habe ich bei der *Typographie* (Nummer 3) bei *Jahr* und *Version* ebenfalls *Kursiv* ausgewählt und bei den *Projektbildern* auch wieder das *Bild*, wobei ich hier als Standard: Miniaturbild 1 und für das modale Fenster das Miniaturbild 5 gewählt habe. Breite und Höhe bleiben auf *Automatisch*.

Beim Menüaufruf habe ich bei den beiden Referenzverweisen im Menü eingestellt, dass die *Kategoriebeschreibung* (Tab *Kategorieoptionen*) angezeigt, und im Tab *Beitragsoptionen*, dass *WEITERLESEN* angezeigt werden soll, da ich dieses in den globalen Optionen jeweils ausgeschaltet habe.

Was das Ganze im Frontend betrifft, also die Gestaltung mit CSS, kommen Sie nicht umhin, sich mit CSS und dessen Auffinden näher zu befassen. Sie finden auf meiner Tutorialsite<sup>30</sup> eine Einführung in CSS und auch, wie CSS-Formatierungen und zugehörige Klassen und IDs gefunden werden können. Alles andere würde den Rahmen dieses Buches sprengen.

Sie können sich das Praxisbeispiel auf meiner Website *time4mambo.de* ansehen.

<sup>30</sup> <http://www.time4joomla.de> – dort unter »Tutorials zu 1.6 bis 3.5 / Templates«





## Beiträge nach eigenem Gusto

Vielleicht benötigen Sie die vielen Einstellungen, die ein Joomla!-Beitrag mitbringt, nicht. Das gilt oft genug insbesondere für die Frontend-Einreichungen von Autoren. Zwar bietet Joomla! ein paar wenige Einschränkungen für die Autoreneinreichungen, aber sehr viele Möglichkeiten haben Sie da nicht.

SEBLOD bringt die Standards im eigenen Gewand mit. Sie können somit Beiträge, Kategorien und User direkt bearbeiten, ohne dass Sie die Core-Elemente direkt anfassen müssen.

Ob Sie einzelne Felder einfach löschen oder umbenennen, sie verstecken oder die ganze Form umgestalten, bleibt Ihnen überlassen. Aber Sie haben zunächst alle Felder, die Joomla! auch anbietet, zur Verfügung und können somit bequem alles Ihrem Geschmack anpassen, sofern Sie weiter mit den Joomla!-Beiträgen arbeiten möchten.

### **Sicherheitskopie der Original-App**

Laden Sie sich vorher die Haupt-App mit dem Artikel auf Ihre Festplatte – dann haben Sie immer eine Sicherheitskopie der Original-App.

Wenn Sie nun in den *Optionen* einstellen, dass bei einem Klick auf *Neu* bei der Beitragsübersicht automatisch Ihre Artikelformulare geladen werden, fällt der Umstieg auf die SEBLOD-Mimik kaum ins Gewicht.

**Vorlageneffekt**

Ein zusätzlicher Effekt der fertigen Vorlagen: Viele Felder werden benutzt, die Sie in Ihren eigenen Formularen unterbringen können. Manchmal hilft es ungemein, einfach einmal nachzuschauen, wie das SEBLOD-Team mit den Feldern umgegangen ist und was eigentlich wie und warum passiert.



## Formulare für jeden Zweck

Christian Schuelling, viele kennen ihn als Whykiki von YouTube, war nicht der Erste, der Videos zu SEBLOD anbot, aber der Erste, der im deutschsprachigen Raum tiefer und konsequenter in die Materie einstieg und über die Einsteigervideos hinaus tiefer gehende SEBLOD-Einstellungen per Videotutorial erläuterte.

Ich freue mich sehr, dass er sich bereit erklärt hat, hier ein Gastkapitel zu übernehmen. Kiki hat sich dafür sehr umfassend mit den E-Mails beschäftigt und ein neues Kontaktformular erstellt, das deutlich mehr kann als das Joomla!-eigene. Sie können mit wenigen Anpassungen letztlich jedes x-beliebige Formular damit erstellen, das versendet werden soll. Lassen wir ihn nun selber zu Wort kommen.

### **Whykiki-Videostream**

Dieses und weitere Videos von ihm, nicht nur zu SEBLOD, finden Sie auf seinem Videostream:  
[www.youtube.com/user/ScreendesignWhykiki](http://www.youtube.com/user/ScreendesignWhykiki)

## 16.1 Warum ein Kontaktformular?

Wie Sie in den vorangegangenen Kapiteln bereits bemerkt haben, lassen sich mit SEBLOD beachtliche Erweiterungen für Joomla! schaffen, die auf jeden speziellen Fall zugeschnitten sind. Deshalb wird SEBLOD von vielen Entwicklern als eine Art »Eier legende Wollmilchsau« beschrieben.

Mit SEBLOD werden nämlich zahlreiche Erweiterungen, die man sich sonst installiert hätte, absolut überflüssig. Ob Galerie, Marktplatz, Newssystem oder ganz abwegige Inhaltstypen, die es so derzeit nicht als Joomla!-Komponente oder Plug-in gibt: mit SEBLOD kein Problem.

Aber auch einfachere Sachen, wie zum Beispiel ein simples Kontaktformular, sind mit SEBLOD problemlos möglich. Der Vorteil liegt dabei auf der Hand: Viele Erweiterungen erfordern einen ungleich höheren Wartungsaufwand, denn vor allem Erweiterungen sind oftmals ein Einfallstor für Hacker. Daher ist es oberstes Gebot im Rahmen von Joomla!, immer alle Erweiterungen auf dem aktuellsten Stand zu halten.

Baut man aber alles auf SEBLOD auf, so ist nur noch Joomla! und eben das Content Construction Kit zu pflegen. Ich möchte Ihnen nun an einem einfachen Beispiel zeigen, wie man mit SEBLOD ein eigenes Kontaktformular aufbaut und somit Erweiterungen, wie das bekannte *dfContact* oder *FlexiContact*, überflüssig macht.

### 16.1.1 Die Vorbereitung: Einrichtung

Wir bauen an dieser Stelle auf dem Wissen auf, das Sie bereits in den letzten Kapiteln erworben haben, und beginnen zuerst mit dem Erstellen eines eigenen Ordners für unsere neue Kontaktformularapplikation. Das ist sinnvoll, weil Sie dann alle Inhalte für jede Applikation in einem eigenen Ordner gespeichert haben. Sie sollten sich bei der Arbeit mit SEBLOD, schon allein der Übersichtlichkeit halber, immer angewöhnen, alles genau durchzustrukturieren.

Klicken Sie auf *App Ordner* und legen Sie dort einen neuen Ordner mit dem Namen *Kontaktformular* an.

Achten Sie darauf, dass Sie den Ordner als 1. Ebene der App festlegen, außer Sie möchten das Kontaktformular im Rahmen mehrerer Applikationen nutzen. Setzen Sie zudem den Haupteintrag auf *Ja*, somit können Sie den angelegten Ordner als Grundlage für Ihr neues Kontaktformular benutzen.

Wählen Sie nun eine Farbe, um den Ordner eindeutig zu kennzeichnen, sowie eine Zeichenfolge und eine Farbe für den darin befindlichen Text. Zudem können Sie ein eigenes Symbol für den Ordner festlegen, das jedoch bereits auf dem Server liegen sollte.

Wählen Sie bei *Elemente* alle vier Typen aus, auch wenn Sie sie aktuell eigentlich gar nicht alle brauchen. Somit erhalten Sie sich jedoch die Freiheit dem Ordner, später andere, jetzt vielleicht noch nicht absehbare Elemente hinzufügen zu können.

The screenshot shows the Joomla! administrator interface for creating an order. The title is 'Kontaktformular' and the parent is 'Top'. The options are as follows:

- App Root (1. Ebene):  Nein,  Ja (1 Ebene der Applikation (App))
- Haupteintrag:  Nein,  Ja (Gekennzeichneter Ordner kann als Gerüst für neuen Inhaltstyp ausgewählt werden)
- Symbol: media/cck/apps/joomla\_article/images/icon.png
- Elemente:
  - Formular & Inhaltstyp, Farbe: #000000
  - Feld, Zeichen: KO
  - Liste & Suchtyp
  - Template, Zeichenfarbe: #ffff

Bild 16.1: Optionen beim Erstellen des Ordners.

Nun erstellen wir die Felder, die wir für unser Kontaktformular benötigen. Hierbei handelt es sich um die Felder, die der Kunde zur Eingabe in unser Formular benötigt, und die Felder, die wir im Hintergrund benötigen, um unser Formular zu nutzen. Werfen wir erst einmal einen Blick auf das Formular und darauf, welche Felder in der Eingabemaske vorhanden sein müssen.

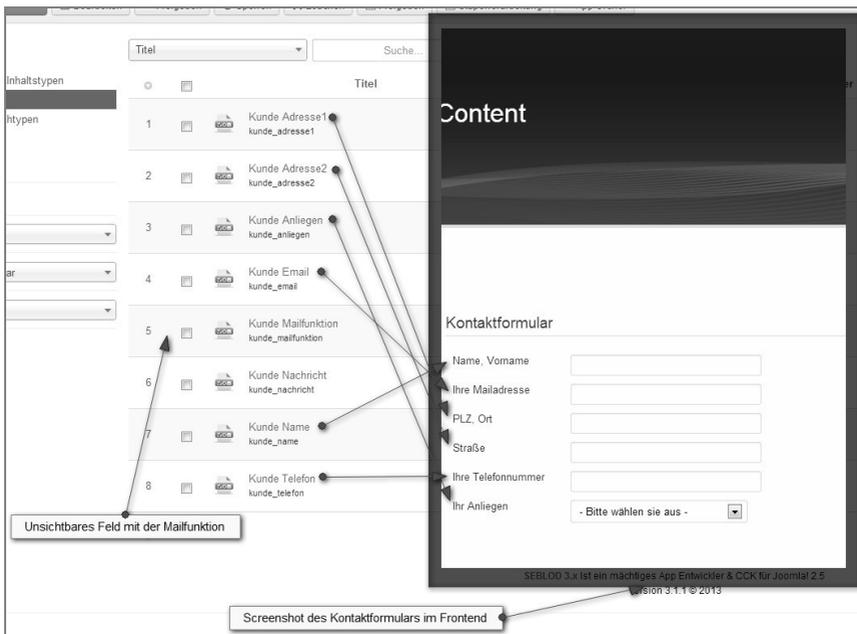


Bild 16.2: Die Felder und das Kontaktformular (Site Form) im Frontend.

Wie Sie auf der Abbildung erkennen, haben wir in unserem Beispiel insgesamt acht Felder angelegt, die wir für unser Formular benötigen. Selbstverständlich haben Sie an dieser Stelle die Möglichkeit, bei Bedarf weitere Felder hinzuzufügen oder deutlich weniger Felder einzubauen.

Optional können Sie sogar Anhänge weiterleiten, worauf wir an späterer Stelle noch einmal zurückkommen wollen. Aus unserem Kontaktformular lässt sich nämlich auch problemlos eine Beschwerdedatenbank einrichten.

Wir belassen es für den Anfang aber bei einem recht einfachen Aufbau und wollen dem Besucher unserer Seite die Möglichkeit geben, mit uns in Kontakt zu treten. Dafür benötigen wir an dieser Stelle seinen Namen, seine Mailadresse, seine Adresse (optional), seine Telefonnummer sowie die Art seines Anliegens und einen Text, den uns der Nutzer hinterlassen kann.

All diese Felder werden uns nun per Mail übermittelt, und wir können ganz gezielt auf das Anliegen unseres Kunden reagieren.

Bis auf Feld 5, *Kunde Mailfunktion*, und Feld 3, *Kunde Anliegen*, handelt es sich bei allen weiteren Feldern um simple Textfelder bzw. einen Textbereich.

Der Unterschied zwischen *Text* und *Textbereich* liegt darin, dass Sie bei Ersterem nur die Möglichkeit haben, eine Zeile zu hinterlassen, bei einem Textbereich jedoch mehrzeilige Inhalte übertragen können, wie zum Beispiel einen kurzen Text.

Erstellen Sie also erst einmal insgesamt sechs Felder vom Typ *Text* und eines vom Typ *Textbereich* jeweils mit einem aussagekräftigen Titel, der es Ihnen auch später bei großen und komplexen Formularen erlaubt, den Überblick zu behalten.

Wählen Sie dafür bei dem Feld als *App Ordner* den zuvor erstellten Ordner aus, den wir zu Beginn dieses Kapitels erzeugt haben.

Das Feld *Name* wird automatisch befüllt und ist ähnlich wie der bekannte Joomla!-Alias zu betrachten. Wichtig ist hierbei, dass wir diesen Namen später auch benötigen, um unsere Felder gezielt anzusprechen. Beachten Sie, dass Sie das Feld *Name* bei Bedarf auch manuell ausfüllen können.

Auf dem zuvor gezeigten Screenshot habe ich Ihnen eine Zuordnung der Felder zu den jeweiligen Feldern im Frontend gezeigt. Darauf sehen Sie auch jeweils den *Feld Namen* oder den *Alias* des Feldes, der kleingeschrieben unter dem Titel zu finden ist.

Ich erwähne das an dieser Stelle noch einmal explizit, weil die Feldnamen absolut wichtig sind für die Arbeit mit unserem Kontaktformular und für die Arbeit mit SEBLOD allgemein.

Nun können Sie eine Bezeichnung für Ihr Feld festlegen. Dabei handelt es sich um das sogenannte *Label*, das der Kunde später als Auszeichnung des Formularfeldes sieht.

Bei *Standard Wert* können Sie einen Wert bzw. einen Text festlegen, der schon als Inhalt im Textfeld zu finden ist und der dann vom Kunden überschrieben werden

müsste. Natürlich handelt es sich bei diesem Wert auch um den Wert, der per Mail übertragen wird, wenn der Kunde nichts anderes einträgt. Wir verzichten an dieser Stelle auf das Eintragen eines Standardwertes und lassen das Feld frei.

Die Mindest- und Maximallänge beim Erstellen des Textfeldes bezieht sich jeweils auf die Menge der Ziffern oder Buchstaben, mit denen das Feld befüllt werden kann. Es ist vollkommen in Ordnung, wenn Sie das auf der Standardeinstellung belassen.

Nur wenn Sie sich sicher sind, wie zum Beispiel bei einem Feld *Postleitzahl*, das erwartungsgemäß nie mit mehr als fünf Ziffern befüllt wird, würde ich dazu raten, einen kleineren Wert einzutragen.

Befüllt der Kunde das Feld mit mehr Zeichen hintereinander, wird eine Fehlermeldung ausgegeben.

Die *Größe* gibt an, wie breit das Feld im Formular auf der Webseite dargestellt wird. Wir haben später natürlich auch noch die Möglichkeit, via CSS darauf zuzugreifen und das Feld designtechnisch zu formatieren.

Zuletzt müssen Sie sich entscheiden, wo der Inhalt des Feldes gespeichert werden soll. Da es sich um ein simples Kontaktformular handelt, dessen Inhalt uns bereits per Mail zugestellt wird, verzichten wir an dieser Stelle darauf, seine Inhalte auch in der Datenbank abzulegen. Wählen Sie deshalb bei *Format/Position* *Kein* als Speicher-methode aus. Vor allem wenn viele Kontakthanfragen zu erwarten sind, ist diese Einstellung sinnvoll, weil wir es so vermeiden, dass unsere Datenbank und unser Joomla!-System mit zahllosen Anfragen gefüllt werden.

Denken wir an dieser Stelle schon einen Schritt weiter Richtung »Beschwerde-datenbank«, ist es sinnvoll, den Inhalt der Felder in der Datenbank abzulegen. Aber dazu später mehr.

Ähnlich verhalten sich die Einstellungen beim Feldtyp *Textbereich*. Wir erstellen also zuerst ein neues Feld, vergeben den Namen *Kunde Nachricht* und speichern es im Ordner *Kontaktformular*. Wir geben dem Feld eine *Bezeichnung/ein Label* und verzichten wieder auf einen *Standard Wert*.

Die *Anzahl der Spalten* dient dem Einstellen der Breite des Textfeldes, die Reihen dienen dem Ausrichten der Höhe. Um es zu verdeutlichen: Bei einem Feld vom Typ *Textbereich* handelt es sich um eine aus HTML bekannte Textarea.

Die *Mindest- und Maximallänge* gibt an dieser Stelle an, wie viele Zeichen man hintereinander schreiben kann, bevor man eine Fehlermeldung erhält.

Um Ihre Mail später ein bisschen aufzuräumen, stehen Ihnen unter *Prozesse* einige Möglichkeiten zur Verfügung. Sie können hier einstellen, wie Zeilenumbrüche im Textfeld per Mail übergeben werden.

Sie haben einfache Zeilenumbrüche und Abstände zur Verfügung sowie verschachtelte Zeilenumbrüche. In der Mail würden die Zeilen via HTML mit

```
001 <br />
```

oder

```
001 <p></p>
```

umbrochen. Auch der Textbereich soll natürlich nicht in der Datenbank gespeichert werden, und wählen wir als *Format/Position* wieder *Kein* aus.

Als kleine Besonderheit bauen wir in unser Formular eine Drop-down-Auswahl ein, in der man die Art seines Anliegens auswählen kann. Das macht es später einfacher, das Anliegen zuzuordnen und zu bearbeiten. Dazu erstellen wir ein Feld vom Typ *Wählen: Einfach (Simple)*. Hierbei handelt es sich um ein Auswahlfeld, in dem der Kunde eine aus mehreren Möglichkeiten auswählen darf. Selbstverständlich können Sie an dieser Stelle auch eine Mehrfachauswahl einsetzen, die Sie unter *Wähle: Mehrere* finden.

Wir erstellen verschiedene Auswahloptionen, wie *Beschwerde*, *Lob*, *Preis Anfrage*, *Kontakt*.

Optional lassen sich einfache Texte eintragen, die Sie auch mit Werten verknüpfen können. Schreiben Sie beispielsweise *Beschwerde=1*. Wird an dieser Stelle vom Kunden *Beschwerde* ausgewählt, wird per Mail der Wert *1* übermittelt. Die Übergabe von Werten hat dann Sinn, wenn Sie die Felder auswerten und damit gegebenenfalls weitere Felder triggern oder wenn man mit den Werten (zum Beispiel in einer Datenbank) weiterarbeitet.

Im nachfolgenden Bild finden Sie noch einmal eine Übersicht der Felder, die wir erstellt haben.

Zu sehen sind die *Feld Namen*, die *Feld Titel*, der *Typ*, die *Speicherung* und vieles mehr. Diese Ansicht ist immer dann sinnvoll, wenn Sie sich noch einmal einen Überblick über alle Felder Ihrer App verschaffen möchten. Stellen Sie dazu die Filterung bei *Felder* auf Ihren *App Ordner* ein, um nur die *Felder* Ihrer App anzuzeigen. Da Sie mit dem *Feld Namen* später weiterarbeiten und die korrekte Schreibweise sehr wichtig ist, können Sie sicherheitshalber alle *Feld Namen* noch einmal in der Übersicht abgleichen oder gegebenenfalls sogar per Copy-and-paste nutzen.

	Titel	App Ordner	# Speicher	Typ	Status	ID
1	Kunde Adresse1 kunde_adresse1	Kontaktformular TOP	-	Text	<input checked="" type="checkbox"/>	5005
2	Kunde Adresse2 kunde_adresse2	Kontaktformular TOP	-	Text	<input checked="" type="checkbox"/>	5006

**Bild 16.3:** Übersicht aller Felder mit *Feld Name*, *Feld Typ*, *Feld Titel* und *Speicherort*.

## 16.1.2 Erstellen der Mailfunktion

Nun kommen wir endlich zum Herzstück unseres Kontaktformulars, denn alle Felder haben wenig Sinn, wenn sie nicht auch per Mail an uns übertragen werden, um auf das Anliegen des Kunden eingehen zu können.

Auch dafür gibt es in SEBLOD bereits ein vordefiniertes Feld, das eine Mailfunktion beinhaltet. Natürlich ist es unabdingbar, dass wir auch diese Mailfunktion erst konfigurieren und unseren Wünschen anpassen.

Erstellen Sie also ein neues Feld, benennen Sie den *Titel* mit *Kunde Mailfunktion* und vergeben Sie natürlich auch einen aussagekräftigen *Alias*, wenn Sie nicht auf den automatisch generierten *Alias* bzw. Namen zurückgreifen möchten. Wählen Sie als *App Ordner* wieder den bereits erstellten Ordner des Kontaktformulars aus und wählen Sie den Typ *Email* aus der Hauptgruppe *Formular*.

Es stehen Ihnen zahlreiche Optionen zur Verfügung, die auf den ersten Blick etwas verwirrend wirken. Aber keine Sorge, wir gehen auf alle diese Punkte ein.

## 16.1.3 Feldtyp Email

Wie bei allen anderen Feldtypen ist auch beim Typ *Email* die Konstruktion das A und O. Hier legen wir eindeutig fest, wie unser Feld bearbeitet wird bzw. welche Funktionen es hat.

Wir vergeben zuerst eine Bezeichnung, die an dieser Stelle jedoch nicht so wichtig ist. Es handelt sich um das sogenannte Label. Da die Mailfunktion für den User später jedoch auf unsichtbar geschaltet wird, ist eigentlich auch das Label egal. Denken Sie jedoch daran, dass Sie trotzdem stringent und sinnvoll benennen sollten, auch wenn man darauf verzichten könnte. Wie schon zuvor erwähnt, werden Sie sich vor allem bei komplexen Formularen freuen, wenn Sie nach monatelanger Untätigkeit an diesem Formular wieder hineinschauen und alles sauber benannt haben.

Bevor ich Ihnen weitere Details erkläre, schauen Sie sich erst einmal den nachfolgenden Screenshot an, damit Sie einen Überblick darüber bekommen, welche Optionen Ihnen an dieser Stelle zur Verfügung stehen.

Konstruktion <sup>1</sup>

Bezeichnung	<input type="text" value="Mailfunktion"/>	Schicke E-mail	<input type="button" value="Einreichen"/> <input type="text" value="mein_feld"/>
Von	<input type="text" value="Feld (Formular)"/>	Von (E-mail / Feld)	<input type="text" value="kiki@kiki.de"/>
Absender	<input type="text" value="Name"/>	Von (Name / Feld)	<input type="text" value="Christian"/>
Objekt	<input type="text" value="#kunde_anliegen#"/>	an	<input type="text" value="kiki@kiki.de, info@kiki.de"/>
Nachricht	<input type="text" value="Editor"/>	An (Felder)	<input type="text" value="kunde_mail, inhaber_mail"/>
Nachricht (Feld)	<input type="text" value="[fields] oder #kunde_nachricht#"/>	An (Administratoren)	<input type="text" value="admin"/>
Schicke Anhang (Feld)	<input type="text" value="kunde_anhang"/>	Anhang (Felder)	<input type="text" value="kunde_anhang1, kunde_anhang2"/>
Kopie an	<input type="text" value="Kein"/>		
Blindkopie an	<input type="text" value="Kein"/>		
Größe	<input type="text" value="32"/>		

**Bild 16.4:** Der Feldtyp *Email* im Überblick.

Nachdem Sie eine *Bezeichnung* festgelegt haben, müssen Sie entscheiden, wie und wann die Mail versendet wird. Dazu stehen Ihnen unter *Schicke E-Mail* insgesamt vier verschiedene Optionen zur Verfügung, die ich Ihnen kurz erläutern möchte. Bedenken Sie: Die Mailfunktion dient nicht nur der Erstellung simpler Kontaktformulare, sondern wird auch benutzt, um das Redaktionssystem zu steuern.

SEBLOD arbeitet in den meisten Fällen auf Artikelbasis, das heißt, dass üblicherweise auch ein Artikel erstellt wird. In unserem Fall verzichten wir darauf jedoch. Trotzdem ist es wichtig, sich dies ins Gedächtnis zu rufen, denn nur dann haben die vier folgenden Auswahloptionen tatsächlich Sinn.

- *Niemals (Never)*: Es wird in keinem Fall eine Mail verschickt, auch nicht, wenn ein neuer Artikel erstellt wird.
- *Einreichen (Submission)*: Eine Mail wird versendet, wenn der User einen neuen Artikel erstellt bzw. einreicht. Diese Funktion ist auch bei unserem Formular sinnvoll, denn nun wird in jedem Fall eine Mail versendet, wenn der User das Formular mit dem Send-Button verschickt.

- *Edition (Edition)*: Bei *Edition* wird immer dann eine Mail versendet, wenn ein Beitrag editiert wird. Dies hat im Fall unseres Formulars jedoch keinen Sinn.
- *Immer (Always)*: *Immer* erklärt sich von selbst. Es wird immer eine Mail versendet, egal ob man einen Artikel editiert, neu erstellt etc.  
Auch diese Möglichkeit können wir für unser Formular benutzen.

Zusätzlich finden Sie dahinter eine Eingabemöglichkeit, in der Sie einen *Feldnamen* hinterlegen können. Benötigt wird das für unser simples Formular nicht. Jedoch möchte ich es nicht unerwähnt lassen. Dieses Feld dient der dynamischen Weiterleitung unserer Mail.

Sie können einen *Feldnamen* eintragen, und in diesem Feld muss eine *Nummer*/ein *Integer* übergeben werden. Dabei steht:

- *Never/Niemals* = 0
- *Submission/Einreichen* = 1
- *Edition* = 2
- *Always/Immer* = 3

Sie müssen also in diesem Feld immer eine Nummer hinterlegen, die dafür sorgt, dass eine Sendemethode gewählt wird.

#### 16.1.4 Absender des Formulars festlegen

Im nächsten Schritt legen wir den Absender des Mailformulars fest. Dazu stehen Ihnen insgesamt drei Auswahlmöglichkeiten zur Verfügung.

- *Standard*: Als Absender-E-Mail-Adresse wird die Adresse genommen, die Sie in der Joomla!-Konfiguration hinterlegt haben. Sie können sie unter *System/Konfiguration/Server* im Register *Mailing* selbst verändern.
- *E-Mail*: Sie können die Mailadresse im Format *ihremail@mail.de* selbst eingeben. Dies ist sicherlich die einfachste Variante.
- *Feld (Formular)*: Hier haben Sie die Möglichkeit, eine Mailadresse aus einem Feld auszulesen. Haben Sie zuvor zum Beispiel ein Feld definiert, in dem der User seine Mailadresse eingeben muss, können Sie hier den Namen des Feldes eintragen, und Sie bekommen die Kontaktmail von der Mailadresse des Kunden geschickt.

#### Beispiel:

Der Kunde trägt seine Mailadresse in das Feld *kunde\_mail* ein, und Sie nutzen diese eingegebene Adresse, um sie per Formular direkt zu übermitteln.

*Feldtyp »Text« -> Feldname »kunde\_email«*

*Feldtyp »Email« -> Einstellungen -> »Von« Feld (Formular) : kunde\_email*

Bitte beachten Sie, dass der Mailversand natürlich über Ihren Server läuft und die Joomla!-interne Mailingfunktion genutzt wird. Der Mailversand läuft also nicht über den Computer des Kunden.

Ein Vorteil dieser Methode ist unter anderem, dass Sie in Ihrem Mailprogramm direkt auf die Kontaktmail des Kunden antworten können, ohne zuvor seine Mailadresse noch einmal gesondert zu erfassen.

Ähnlich arbeitet *Absender*, hierbei handelt es sich schlicht um den angezeigten Namen, der Ihnen per Formular übermittelt wird. Auch die Optionen sind recht ähnlich:

- *Standard*: Als *Name* wird der Name verwendet, den Sie in der Joomla!-Konfiguration hinterlegt haben. Sie können ihn unter *System/Konfiguration* - *Server* im Register *Mailing* selbst verändern.
- *Name*: Sie können an dieser Stelle selber einen Namen vergeben, z. B. »Webseite XY Kontaktmailer«. Somit können Sie Mails in Ihren Mailprogrammen ganz genau dem Absender zuordnen und sie danach sortieren lassen.
- *Feld (Formular)*: Wie bei *Von* haben Sie hier die Möglichkeit, den Namen aus einem Feld auszulesen. Eine ausführliche Erklärung dazu habe ich Ihnen ja schon beim vorangegangenen Feld geliefert. Beide sind deckungsgleich.
- Kommen wir zum *Objekt*, bei dem es sich um nichts anderes handelt als den *Betreff*. Hier haben Sie wiederum die Möglichkeit selbst, einen statischen Text einzugeben, Sie können aber auch den Inhalt eines einzelnen Feldes auslesen. Hierbei muss beachtet werden, dass der Feldname in Rauten (#) gesetzt wird. Wollen wir das Feld *Kunde Anliegen* als *Betreff* übermitteln, muss bei *Objekt* *#kunde\_anliegen#* eingetragen werden.

### 16.1.5 Das Feintuning

So langsam kommen wir ans Eingemachte. Daher will ich Ihnen vorab noch einmal ein Bild an die Hand geben, denn tatsächlich sind die nachfolgenden Optionen etwas willkürlich verteilt, sodass teilweise falsche Eindrücke entstehen könnten.

**Bild 16.5:** Das Feintuning des Mailfeldes.

Bisher haben wir jeweils zur Auswahl auf der linken Seite eine Entsprechung auf der rechten Seite finden können. Nun aber ist der Inhalt ein wenig durchmischt. Das hängt vor allem damit zusammen, dass uns SEBLOD nun mehrere Möglichkeiten bietet, an wen die Mail verschickt werden soll.

Es ist wichtig, dass die Mail tatsächlich an alle Empfänger verschickt werden kann. Sie können also problemlos alle *An*-Felder befüllen. Sie haben dabei die Wahl, die Mailadresse direkt statisch als Text einzugeben, sie an eine Mailadresse zu senden, die Sie via Feld abgreifen, oder sogar an diverse hinterlegte Administratoren. Wir bleiben also erst einmal auf der rechten Seite des Fensters und schauen uns die verschiedenen Optionen an.

Im Feld *An* haben Sie die Möglichkeit, statisch Mailadressen einzutragen. Die Mailadressen werden dabei nach der üblichen Schreibweise eingetragen, *info@info.de*, und Sie können mehrere Mailadressen durch Kommata trennen.

Bei *An (Felder)* können Sie verschiedene Felder, ebenfalls durch Kommata getrennt, eingeben. Hier werden dann die Mailadressen aus den Feldern ausgelesen. Das ist zum Beispiel dann sinnvoll, wenn man das Kontaktformular zur Bestätigung noch einmal an den Kunden senden möchte.

Sehr eindeutig ist die letzte Option *An (administratoren)*. In dem nebenstehenden Feld werden Ihnen alle Administratoren aus der Benutzerverwaltung angezeigt. Per Klick können Sie auswählen, welchem Administrator eine Mail geschickt werden soll. Eine Mehrfachauswahl ist natürlich möglich.

Die Mailadresse der einzelnen Administratoren ändern Sie problemlos in der Joomla!-internen Nutzerverwaltung.

Bevor wir uns mit dem eigentlichen Inhalt, der sogenannten Nachricht oder Message, befassen, wollen wir uns die restlichen Einstellungen ansehen, die von Relevanz sein können.

Da ist zum einen der Anhang. Kunden haben die Möglichkeit, über das Feld *Schicke Anhang (Feld)*, auch *Attachment* genannt, Anhänge an uns zu übermitteln.

Dabei haben wir die Auswahl, entweder mehrere Felder zu übertragen, siehe auf der rechten Seite *Anhang (Felder)*, oder nur ein einzelnes Feld.

Anhänge können Grafiken, Dokumente und vieles mehr sein. Beachten Sie, dass die Mailfunktion nach wie vor von Joomla! bzw. Ihrem Server bereitgestellt wird. Je nach Größe des Dateianhangs kann das also zu Problemen führen. Bei Time-outs empfiehlt sich ein Anruf bei Ihrem Hoster.

Die Felder *Kopie an* und *Blindkopie an* dürften Ihnen bekannt sein. Sie haben an dieser Stelle die Möglichkeit, die Mail per Kopie oder Blindkopie an weitere Empfänger zu versenden. Sie haben wiederum drei Auswahlmöglichkeiten, die ich Ihnen bereits erläutert habe. Dazu zählen *Kein*, *E-Mail* und *Feld (Formular)*. Sie können also dort entweder eine feste, statische Mailadresse eintragen oder die Mailadresse aus einem vordefinierten Feld auslesen.

#### Kurzer Zwischenruf

Auch wenn ich mich an dieser Stelle wiederhole, aber vergessen Sie nicht, dass die Mailfunktion für mehr gedacht ist als »nur« für ein einfaches Kontaktformular. Sie können mit den vorhandenen Feldern und Möglichkeiten Benachrichtigungen für nahezu jeden Fall erstellen, ob Sie dem Administrator eine Meldung schicken wollen, wenn jemand ein neues Bild in eine Galerie eingefügt hat, oder ob ein User eine neue Beschwerde in die Datenbank eingereicht hat. Sie können sogar einen kleinen Onlineshop realisieren und zum Beispiel eine Mail versenden, wenn der Artikelbestand weniger als zehn Artikel beträgt. Sie sehen, es gibt unzählige Möglichkeiten, und wir behandeln nur eine davon.

Ich empfinde diesen Hinweis als sehr wichtig, denn erst dann haben die zahlreichen Felder und ihre Möglichkeiten innerhalb der Mailfunktion auch wirklich Sinn.

### 16.1.6 Das Erstellen der E-Mail-Nachricht

Kommen wir nun zur Essenz unseres Formulars. Wir haben uns vor allem den Grundlagen gewidmet, Felder erstellt und dafür gesorgt, dass die Inhalte unseres Formulars auch versendet werden. Am wichtigsten ist jedoch, dass wir einen Inhalt übermitteln. Sonst hat die ganze Übung ja keinen Sinn. Daher wollen wir uns nun ganz ausführlich den beiden letzten, offenen Eingabemöglichkeiten widmen:

- *Nachricht (Feld)* – In diesem Bereich haben Sie die Möglichkeit, ein einzelnes Feld festzulegen, das übermittelt wird. Dies kann zum Beispiel auch eine Gruppe sein. Außerdem können Sie durch die Eingabe von *[fields]* alle Felder des Formulars gesammelt rendern.

- *Nachricht (Editor)* – Während *Nachricht (Feld)* vor allem den Inhalt einzelner Felder bzw. alle Felder unformatiert rendert, haben Sie bei *Nachricht (Editor)* deutlich tiefer greifende Möglichkeiten. Über den Editor können Sie den Inhalt der Mail Ihren Wünschen anpassen. Dabei handelt es sich um eine Version des Ihnen bereits bekannten *Tiny Editors*, der standardmäßig bei Joomla! dabei ist. Sollten Sie im Backend für Ihren User einen anderen Editor ausgewählt haben, so wird dieser hier angezeigt. Sie haben aber in den SEBLOD-Optionen jederzeit die Möglichkeit, auf einen anderen Editor umzustellen, vorausgesetzt natürlich, dass Sie einen weiteren Editor installiert haben.

Bei der Erstellung des Mailtextes stehen uns diverse Möglichkeiten zur Verfügung. So können Sie mit dem Editor beispielsweise PHP-Code ausführen und sich über die PHP-Date-Funktion das Sendedatum ausgeben lassen. Dies ist nur ein Beispiel, PHP lässt sich hier vortrefflich einsetzen, um noch weit mehr Informationen zu übergeben.

Zudem können Sie auf Joomla!-eigene Werte zugreifen, wie z. B. den Seitennamen [*sitename*] oder die Seiten-URL [*siteurl*]. Die Werte können auch via PHP ausgelesen und genutzt werden. Es gibt eine Liste mit nutzbaren Parametern<sup>31</sup>. Dort finden Sie weitere Informationen zu den Konfigurationswerten. Unglücklicherweise findet sich jedoch derzeit keine Liste mit allen nutzbaren Joomla!-Werten.

Neben der Nutzung von PHP-Code oder internen Werten steht Ihnen aber auch die Möglichkeit zur Verfügung, auf SEBLOD-Werte zuzugreifen. Sie können somit z. B. das Label/die Bezeichnung eines Feldes, seinen Wert oder viele weitere Attribute auslesen. Der Aufruf erfolgt dabei jeweils über folgenden Code:

```
001 $cck->getAttribute('fieldname');
```

Dabei müssen Sie **Attribut** durch den Namen des jeweiligen Attributs ersetzen. Wollen Sie also zum Beispiel das Label eines bestimmten Feldes auslesen, müssen Sie den Code anpassen:

```
001 $cck->getLabel('name_ihres_feldes');
```

Sie haben so die Möglichkeit, auch Werte auszulesen, die sich nicht direkt auf den Inhalt des Feldes beziehen. Das mag für unser Kontaktformular nun wenig Sinn haben. Es gibt aber bei komplexeren Formularen oder Anwendungen durchaus vielfältig Möglichkeiten, diese Werte sinnvoll zu nutzen.

Zu guter Letzt will ich Ihnen an dieser Stelle noch eine Möglichkeit zeigen, wie Sie die Werte der Felder auslesen und per Mail sauber formatiert ausgeben. Sicherlich ist das an dieser Stelle der wichtigste und auch interessanteste Weg, den Inhalt seiner Mail zu befüllen. Wie Sie schon in den vorangegangenen Abschnitten gemerkt haben, ließen sich auch einfach alle Felder oder ein einzelnes Feld unformatiert auslesen und versenden.

---

<sup>31</sup> <http://docs.joomla.org/JApplication/getCfg>

Wollen Sie nun aber eine sauber strukturierte Mail erhalten, müssen Sie den Editor mit einem bestimmten Format füttern.

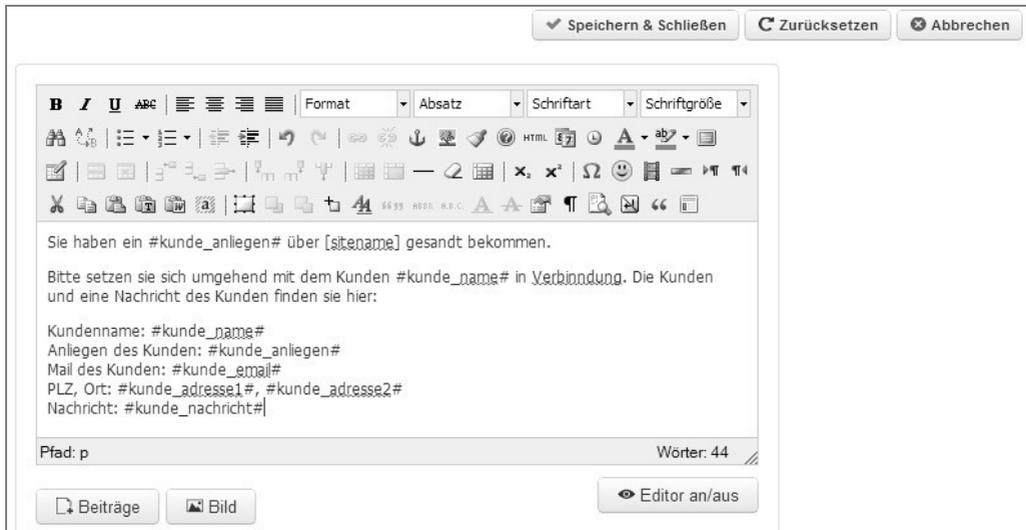
Einzelne Felder lesen Sie aus, indem Sie den Namen der Felder mit zwei Rauten »einkleiden«. Sie können sich die Rauten wie ein öffnendes und ein schließendes Tag aus dem HTML-Bereich vorstellen. Felder werden also innerhalb des Editors immer in folgendem Format ausgelesen: `#name_des_feldes#`.

Nicht zu vergessen ist jedoch, dass Sie natürlich auch statischen Text nutzen können, um den Text unserer Mail ansprechend zu gestalten. Im Folgenden möchte ich Ihnen einen Beispieltext vorstellen, den ich Ihnen nachstehend genauer erklären werde.

### 16.1.7 Der Inhalt im Detail

Mailinhalt im Editor:

```
001 Sehr geehrter [admin],
002 Sie haben eine Nachricht der Webseite [sitename] erhalten. Der Kunde
    #kunde_name# möchte Kontakt zu Ihnen aufnehmen. Er hat folgendes
    Anliegen #kunde_anliegen#.
003 Der Kunde hat Ihnen folgende Nachricht hinterlassen:
004 #kunde_nachricht#
005 Kontaktdaten des Kunden:
006 Name: #kunde_name#
    Adresse, Ort: #kunde_adresse1#, #kunde_adresse2#
    Telefonnummer des Kunden: #kunde_telefon#
    Mail des Kunden: #kunde_email#
007 Bitte setzen Sie sich umgehend mit dem Kunden in Verbindung.
```



**Bild 16.6:** Der Inhalt der Mail im Editor.

Sie haben nun eine Mischung diverser verschiedener Werte und Texte vorgenommen. Sie starten mit einem statischen Text und dem Aufruf des Joomla!-Parameters für den Namen des Admins. Dieser wird die Mail gesendet bekommen, wir sprechen ihn somit direkt an. In den folgenden Zeilen teilen wir ihm mit, dass er eine Mail der Webseite bekommen hat. Den Namen der Webseite lesen wir wieder aus den Joomla!-Parametern aus.

Dann folgt das Auslesen der ersten Felder. Wir geben den Inhalt des Feldes *kunde\_name* weiter. So übermitteln wir das Anliegen des Kunden. Wir erinnern uns: Es handelt sich um eine einfache Drop-down-Auswahl, deren Wert wir als Text per Mail übermitteln. So verhält es sich auch mit den weiteren SEBLOD-Feldern, die wir alle auslesen und in den Inhalt unserer Mail verpacken.

Das Gute an dieser Arbeitsweise mit dem Editor-Feld ist, dass Sie den kompletten Inhalt der Mail nach eigenen Vorstellungen formatieren können.

Bitte denken Sie daran, dass Ihnen neben den hier vorgestellten Möglichkeiten natürlich auch die Formatierungsoptionen des Editors zur Verfügung stehen, denn das Ergebnis ist letztlich reines HTML. Sie können also sogar ein Bild einbauen, Sie können Text fett oder kursiv setzen, Absätze einfügen oder gar Tabellen einbauen, wenn Ihnen der Sinn danach steht.

Wie bei SEBLOD im Allgemeinen gilt: Auch mit der Mailfunktion und dem Editor ist nahezu alles Vorstellbare machbar.

Damit kommen wir dem Ziel schon einen ganz großen Schritt näher. Wir haben alle Felder erstellt, das Formular (*Site Form*) zusammengesetzt und die Mailfunktion bis

ins Detail ausgearbeitet. Nun fehlt eigentlich nur noch die Anzeige auf unserer Webseite, mit der wir uns im letzten Abschnitt beschäftigen wollen.

### 16.1.8 Darstellung des Formulars

Damit die Kunden unser erstelltes Formular auch nutzen können, müssen wir es natürlich im Frontend aktivieren und darstellen. Das wird vielen von Ihnen wahrscheinlich keinerlei Probleme machen. Ich möchte der Vollständigkeit halber diesen Schritt allerdings kurz erklären.

Erstellen Sie bitte einen neuen Menüpunkt, dem Sie wie üblich einen Namen geben. Wählen Sie als Menütyp *SEBLOD/Formular* und dann den Namen Ihres Inhalts- bzw. Formulartyps aus. In unserem Fall handelt es sich um den Inhaltstyp *Kontaktformular*.

Nun haben Sie alle Schritte unternommen, und Ihr Formular sollte auf der Seite zu sehen sein. Wechseln Sie bitte in das Frontend Ihrer Seite und prüfen Sie, ob alles so funktioniert, wie es sollte.

### 16.1.9 Berechtigungen

Und jetzt stehen Sie vermutlich vor einem Problem: Denn Sie bekommen kein Formular zu sehen, sondern nur eine Fehlermeldung, dass Sie derzeit keine Berechtigung haben, den Inhalt der Seite anzusehen.

Genau so wird es auch Ihren Kunden gehen, die mit einem Gast- oder einem öffentlichen Profil Ihre Seite besuchen. Alle registrierten User bzw. Admins werden das Formular im eingeloggten Zustand sehen. Das hat jedoch bei dieser Variante nur bedingt Sinn, wir wollen den User ja nicht zu einer Registrierung zwingen.

Daher müssen wir uns die Berechtigungen noch einmal anschauen. Diese sorgen dafür, dass die Formularfelder nicht sichtbar sind. Ein kleiner Hinweis an dieser Stelle: Gleiches gilt, wenn Sie dem Kunden die Möglichkeit bieten, einen Text zu übermitteln, zum Beispiel wenn Sie ein kleines Redaktionssystem aufgebaut haben. Sobald Sie dem Kunden Texteingabemöglichkeiten bieten, müssen Sie auch die Berechtigungen anpassen.

Beachten Sie, dass es sich hierbei um die Berechtigungen in SEBLOD selbst handelt. Diese hängen zwar auch mit den Joomla!-Berechtigungen zusammen, jedoch sind beide getrennt steuerbar. Allerdings ist das *Berechtigungen*-Feld in SEBLOD 3.x sehr gut versteckt, was oftmals zu Irritationen führt.

Schauen Sie sich den unteren Screenshot an, auf dem ich Ihnen zeige, wohin Sie klicken müssen, um die richtige Ansicht zu erhalten

The screenshot shows the configuration page for a contact form in SEBLOD. At the top, there are fields for 'Titel \*' (Kontaktformular), 'App Ordner' (Kontaktformular), 'Beschreibung' (Editor), and 'Status'. Below this is a navigation bar with tabs: 'Admin Formular', 'Seitenformular', 'Intro', 'Inhalt', 'Konfiguration' (selected), 'Felder', and 'Template'. Under 'Konfiguration', there are sections for 'Ansichten' and 'Optionen'. The 'Optionen' section includes 'Alias (Optional)', 'Inhaltsobjekt' (Artikel), 'Berechtigungen', 'Menüzuweisung ...', 'Inhaltserstellung' (Erlaubt), and 'Suche' (Intro). The 'Konfiguration (für Seiten Formularansicht)' section includes 'Nachrichtstil' (Nachricht), 'Weiterleitung' (aktuell), and a 'Nachricht (Danke)' text area containing 'Test'.

**Bild 16.7:** Berechtigungen festlegen.

Sie müssen nun bei *Inhaltserstellung* den Wert auf *Erlaubt* setzen und zudem noch einmal auf den rechten Button *Berechtigungen* klicken. Dort wählen Sie *Öffentlich* aus und setzen bei Aktion *Erstellen* den Wert ebenfalls auf *Erlaubt*. Jetzt sind die Einstellungen korrekt, und das Formular wird angezeigt.

#### Erstellen von Inhalt

Ein Hinweis zu diesem Thema, denn es mag etwas missverständlich sein: Wir sollten uns immer ins Gedächtnis rufen, dass das Formular in gewisser Weise Inhalte erstellt, selbst wenn es sich nur um den Versand einer Mail handelt. Für SEBLOD ist das »Erstellen von Inhalt«. Und wenn dazu die Berechtigungen nicht ausreichen, dann ist das *Erstellen* unmöglich.

Nun haben Sie Ihr eigenes Kontaktformular mit SEBLOD erstellt und gelernt, was Sie mit dem Mailfeld alles anstellen können.

## 16.2 Erweiterung zum Beschwerdeformular

Nachdem ich Ihnen auf den letzten Seiten vorgestellt habe, wie man ein Kontaktformular aufbaut, möchte ich Ihnen nun näherbringen, wie Sie dieses Formular erweitern und zu einem Beschwerdeformular ausbauen können. Beschwerdeformular ist exemplarisch zu verstehen, um Ihnen weitere Möglichkeiten darzulegen. Natürlich lassen sich auf diesem Wege auch viele andere Anwendungen erstellen. Allerdings sind wir mit dem Beschwerdeformular zumindest thematisch sehr nah an unserer vorangegangenen Anwendung.

Aufgrund der thematischen Nähe und da wir einige Felder, die wir auf den letzten Seiten bereits erstellt haben, wieder benutzen werden, führen wir beide Anwendungen zusammen.

Erstellen Sie also bitte zuerst einen neuen *App Ordner*, den Sie als Unterordner des bereits vorhandenen Ordners *Kontaktformular* erstellen. So haben Sie alle Felder beider Formulare gesammelt in einem »großen« Hauptordner. Und nur die zusätzlichen Felder, die wir für unser neues Formular benötigen, werden in den neuen Ordner *Beschwerdeformular* gespeichert.

Dies hat den Vorteil, dass Sie später – für den Fall, dass Sie Ihre App weitergeben oder anderweitig nutzen wollen – alles sauber und kompakt abgebildet haben und sehr einfach herunterladen und wieder neu installieren können.

### 16.2.1 Grundlagen des Beschwerdeformulars

Nachdem Sie einen Ordner erstellt haben, benötigen Sie einen neuen Inhaltstyp. Wir beginnen wieder mit dem Erstellen der verschiedenen Felder. Da wir viele Felder schon vorab für unser Kontaktformular erstellt haben, brauchen wir nur noch einige zusätzliche Felder, mit denen der Kunde seine Beschwerde genau kategorisieren kann.

**Beschwerdeformular**

Name, Vorname

Ihre Telefonnummer

PLZ, Ort

Straße

Anliegen des Kunden

Art der Beschwerde

Ihre Nachricht

oder [Abbrechen](#)

**Bild 16.8:** Das Beschwerdeformular.

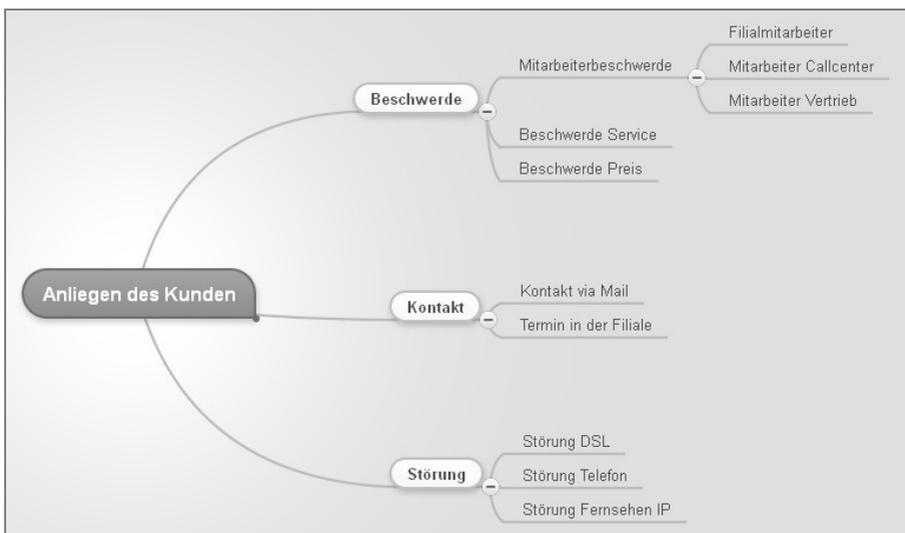
Bevor Sie jedoch frei heraus einfach Felder anlegen, möchte ich Sie an dieser Stelle einmal kurz bremsen.

Unser Kontaktformular war eine einfache Sache, wenige Felder, einfach und übersichtlich. Geht es wie jetzt jedoch um ein Beschwerdeformular oder sonstige Formulare, die einen komplexeren Hintergrund haben, kann man schnell die Übersicht verlieren.

Daher ist es sinnvoll, das ganze Vorhaben in Form einer MindMap (siehe Screenshot) oder ganz einfach auf Papier zu skizzieren. So haben Sie einerseits einen Überblick über alle zu erstellenden Felder, zudem ist es gut möglich, dass Ihnen beim Skizzieren oder Notieren Ihres Formulars Unstimmigkeiten auffallen oder Verbesserungen, die Sie direkt einfließen lassen können.

Auch wenn das natürlich zusätzlicher Aufwand ist – Sie werden sich, sobald Sie loslegen und sich verzettelt haben, freuen, wenn Sie alles ganz genau notiert haben.

Daher zeige ich Ihnen an dieser Stelle einen Screenshot, den ich mit dem kostenlosen Tool *Mindmeister*<sup>32</sup> erstellt habe, damit können Sie einfach solche Maps erstellen, die mir beim Erzeugen meines Formulars geholfen hat oben.



**Bild 16.9:** Der logische Aufbau des Beschwerdeformulars.

Sie können anhand des Screenshots den Aufbau des Formulars erkennen. Wir nutzen jeweils nur einfache Drop-down-Auswahlen, natürlich können Sie Ihr Formular mit eigenen Feldern befüllen. Ihrem Einfallsreichtum sind keinerlei Grenzen gesetzt.

Ich will das oben gezeigte Schema noch einmal in Worte fassen, um Ihnen die Funktionsweise des Formulars zu verdeutlichen:

<sup>32</sup> <http://www.mindmeister.com/de>

Der Kunde startet zuerst mit einem einzigen Drop-down-Feld, in dem er auswählen kann, ob es sich um eine Beschwerde, einen Kontakt oder eine Störungsmeldung handelt. Je nachdem, wofür er sich entscheidet, werden - davon abhängig - neue Felder angezeigt.

Wählt der Kunde beispielsweise »Beschwerde« aus, erscheint ein weiteres Drop-down-Feld mit den Auswahlmöglichkeiten »Mitarbeiterbeschwerde«, »Beschwerde Service« und »Beschwerde Preis«. Und auch hier ergeben sich später neue Möglichkeiten, je nachdem, was der Kunde auswählt.

Damit wird schon an dieser Stelle deutlich, was bei dieser Art von Formular am wichtigsten ist: die Abhängigkeit verschiedener Felder voneinander.

Erstellen wir zuerst unsere Felder. Im Fall des Beschwerdeformulars müssen dafür mindestens sechs neue Felder erstellt werden.

Es handelt sich dabei jeweils immer um eine simple *Einfachauswahl*. Da wir den Rahmen an dieser Stelle nicht sprengen wollen, erläutere ich nur für ein einzelnes Feld den Aufbau.

Bei allen anderen Feldern ist der Aufbau ähnlich, da es sich ja jeweils um den gleichen Feldtyp handelt, nur der Inhalt der Auswahl ist bei jedem Feld anders.

Wir vergeben also zuerst einen Namen bzw. einen *Titel* für das Feld, woraus dann passend unser *Alias* entsteht. Auch an dieser Stelle will ich noch einmal mahnend eingreifen: Bitte vergeben Sie vor allem bei komplexen Formularen sinnvolle Feldtitel, denn vor allem wenn Sie später die Abhängigkeiten der Felder festlegen müssen, werden Sie froh sein, wenn Sie erkennen können, welches Feld mit einem anderen Feld verbunden ist.

Unser erstes Feld lautet *Anliegen Level 1*, es wird gespeichert im Ordner *Beschwerdeformular*. Als *Typ* wählen Sie an dieser Stelle bitte *Wählen -> Einfach*.

Im Bereich *Konstruktion* legen Sie die Feldinhalte fest, also das, was dem Kunden zur Auswahl stehen soll.

In unserem Fall sind das die Werte *Beschwerde*, *Kontakt* und *Störung*. Natürlich vergeben wir auch bei der *Bezeichnung* einen aussagekräftigen Titel: *Anliegen des Kunden*. Im Feld *Wähle Bezeichnung* legen Sie fest, was in Ihrem Feld zu sehen sein soll, wenn der Kunde noch keine Auswahl getroffen hat.

Im Bereich *Standard Wert* brauchen wir aktuell keinerlei Angaben zu machen. Wir hätten hier aber die Möglichkeit, eine Vorauswahl für den Kunden zu treffen.

### 16.2.2 Speichern oder nicht?

Der nächste Punkt ist essenziell. Sie erinnern sich sicherlich, dass wir bei unserem Kontaktformular keine Inhalte gespeichert haben. Es hat ja keinen Sinn, Kontaktanfragen in der Joomla!-Datenbank zu speichern. Etwas anders könnte es sich bei Beschwerden gestalten.

Denn einerseits könnten wir die Beschwerden ganz simpel per Mail versenden (was wir auch auf jeden Fall tun werden). Zusätzlich hätten wir auch die Möglichkeit, alle eingehenden Beschwerden in Beiträgen zu speichern und diese zum Beispiel im Listenformat wieder auszugeben. Das wäre vor allem dann sinnvoll, wenn man Joomla! bzw. seine Webseite auch als Beschwerdedatenbank benutzen möchte.

Das würde in diesem Fall den Rahmen der Erklärung sprengen, trotzdem will ich es nicht ganz unerwähnt lassen.

Sie können beispielsweise alle Beschwerden in einer Kategorie speichern, diese Kategorie nur den Beschwerdebearbeitern sichtbar machen (Berechtigungen!) und genau diese Kategorie dann im Blog- oder einem speziell erstellten Listenformat zugänglich machen.

All dies ist problemlos möglich, erfordert aber zusätzliche Bearbeitung, die wir uns an dieser Stelle ersparen wollen. Ich gehe davon aus, dass Sie mit dem bereits erworbenen Grundwissen in diesem Buch selbst in der Lage sind, derlei Listenansichten zu erstellen.

Um es also kurz zu machen: Auch im Fall des Beschwerdeformulars speichern wir unsere Inhalte nicht, sondern übermitteln sie ganz einfach per Mail.

Wählen Sie also im Bereich *Speicher* unter *Format/Position* *Kein* aus. Somit werden die übermittelten Inhalte nicht in der Datenbank gespeichert.

### 16.2.3 Der Grundaufbau des Formulars

Erstellen Sie zuerst die weiteren benötigten Felder nach dem bereits bekannten Schema. Natürlich können Sie dieses Formular nach Belieben erweitern oder vereinfachen.

Wir erstellen also zunächst einen neuen Inhaltstyp, den wir *Beschwerdeformular* nennen, und können dort damit beginnen, die einzelnen Felder dem *Seitenformular* zuzuweisen.

Wie ich schon eingangs erwähnte, nutzen wir einige Felder aus dem Kontaktformular mit. Denn den Namen des Kunden und seine Telefonnummer haben wir als Feld bereits erstellt, und wir wollen unsere Arbeit so einfach wie möglich halten.

Auch dabei stehen uns mehrere Möglichkeiten zur Verfügung, denn wir könnten an dieser Stelle auch mit Gruppen arbeiten und die bereits bekannten Felder in eine Gruppe auslagern. Aber auch das will ich hier nur erwähnen, damit Sie es nicht vergessen. Im Rahmen dieses Kapitels bleiben wir beim konventionellen Weg und ziehen die Felder alle händisch in das *Seitenformular*.

Damit ist unsere Hauptaufgabe an dieser Stelle eigentlich erledigt. Wie schon beim Kontaktformular wird Ihnen aber auffallen, dass Sie das neue Beschwerdeformular nicht sehen können, sobald Sie es mit einem Menüpunkt verknüpft haben.

Das liegt an den Berechtigungen, die tatsächlich für jeden neuen Inhaltstyp neu definiert werden müssen.

#### 16.2.4 Abhängigkeiten von Feldern

Unser Formular ist nun erstellt, aber wie Sie nach dem Festlegen der Berechtigungen wohl festgestellt haben, werden alle unsere eingebundenen Drop-down-Felder angezeigt. Und genau so soll unser Formular ja nicht sein. Wenn wir uns an den Screenshot erinnern, den ich Ihnen zu Beginn gezeigt habe, dann wird deutlich, dass einige Felder erst erscheinen sollen, wenn in einem Feld davor ein bestimmter Wert festgelegt wurde. Das bedeutet: Die verschiedenen Felder in den verschiedenen Ebenen müssen voneinander abhängig gemacht werden.

Zum Glück bietet SEBLOD auch diese großartige Funktion, die es uns ermöglicht, sehr komplexe Formulare sehr übersichtlich für den Kunden darzustellen.

Wir begeben uns an dieser Stelle wieder in das Joomla!-Backend und rufen dort über *Konstruktion* und *Formulare & Inhaltstypen* unseren Inhaltstyp *Beschwerdeformular* auf.

Wechseln Sie in das Register *Seitenformular* und klicken Sie anschließend auf die Nummer 5. Nun erscheinen im Bereich der Felder Verweise, die Sie anklicken können. Wenn Sie noch keine Abhängigkeit festgelegt haben, erscheint dort zuerst *Hinzufügen*, später findet sich der Verweis *Bearbeiten*, wenn Sie bereits Abhängigkeiten festgelegt haben.

#### 16.2.5 Abhängigkeiten im Detail

Das *Abhängigkeiten*-Feld ist auf den ersten Blick etwas irreführend, denn tatsächlich müssen Sie »das Pferd von hinten aufzäumen«. Soll heißen, der hintere, rechte Bereich ist der Trigger oder Auslöser, der dafür sorgt, dass etwas passiert.

Wenn der Kunde im Feld *Anliegen* den Wert *Beschwerde* auswählt, soll ein neues Drop-down-Feld mit dem Titel *Anliegen Level 2* erscheinen.

Somit ist die Auswahl des Wertes *Beschwerde* der *Trigger* oder *Auslöser*, der dafür sorgt, dass etwas passiert. Getriggert wird jedoch, und das ist im ersten Moment noch etwas verwirrend, das neu erscheinende Feld. Das bedeutet, Sie müssen den *Trigger* natürlich im Feld *Anliegen Level 2* festlegen.

Dieser *Trigger* findet sich im *Abhängigkeiten*-Feld auf der rechten Seite. Schauen Sie sich dazu einfach den nachfolgenden Screenshot an, der das Ganze noch besser illustriert.

The screenshot shows a configuration window with two main sections: 'Status' and 'Auslöser'.  
 Under 'Status', there are three rows. Each row starts with 'Dieses Feld' followed by a dropdown menu. The first dropdown is set to 'ist sichtbar'. To the right of each dropdown is a text input field and a checkbox with a checkmark. The second checkbox has a '2' as a superscript.  
 Between the 'Status' and 'Auslöser' sections is a 'WENN' section with a 'und' dropdown menu.  
 Under 'Auslöser', there are two rows. Each row starts with a dropdown menu. The first dropdown is set to 'Anliegen Level1'. To the right of each dropdown is another dropdown menu. The first dropdown in the second row is set to 'COM\_CCI'. To the right of each dropdown is a button. The first button is labeled 'Beschwen' and has a double arrow icon to its right. The second button is empty and also has a double arrow icon to its right.

**Bild 16.10:** Abhängigkeiten von Feldern in SEBLOD.

Im ersten Drop-down-Feld legen Sie fest, welches der erstellten Felder als *Trigger* dienen soll. Wir erinnern uns: Wir möchten, um das zweite Level bzw. die zweite Ebene freizuschalten, die erste Ebene als *Trigger* benutzen. Wählen Sie also nun das Feld *Anliegen Level1* aus.

Als Nächstes legen Sie im Feld danach die Bedingung fest. Die Felder sind zum Teil nicht sauber oder gar nicht übersetzt, lassen Sie sich davon jedoch nicht irritieren. Derzeit arbeitet ein größeres Team der SEBLOD-Developer an einer sauberen deutschen Übersetzung, damit werden auch die etwas kryptischen Bezeichnungen durch sauberen Text ersetzt.

Mit ein wenig Englischkenntnissen oder notfalls einem Übersetzungstool lässt sich dies jedoch leicht nachvollziehen.

In unserem Fall nutzen wir die Vergleichsfunktion *is equal*, was so viel bedeutet wie *ist gleich*. Bevor wir alles zusammensetzen, kommen wir zum letzten Feld. Das befüllen Sie, indem Sie auf die kleinen orangefarbenen Pfeile klicken, worauf sich ein Pop-up-Fenster öffnet, das Ihnen die Werte aus Ihrem Feld *Anliegen Level1* anzeigt. Wir wählen hier *Beschwerde* aus. Um auch das noch einmal deutlich zu zeigen, habe ich einen weiteren Screenshot hinzugefügt.



Bild 16.11: Bedingungen von Feldern.

Damit haben wir eine erste Bedingung erstellt. Wenn also der Wert im Feld *Anliegen Level 1* gleich *Beschwerde* ist, passiert etwas. Was dann passiert, legen Sie auf der linken Seite unter *Status* fest. Dort haben Sie die Möglichkeit, aus verschiedenen vorgegebenen Werten auszuwählen. Auch dazu zeige ich Ihnen ein Bild.

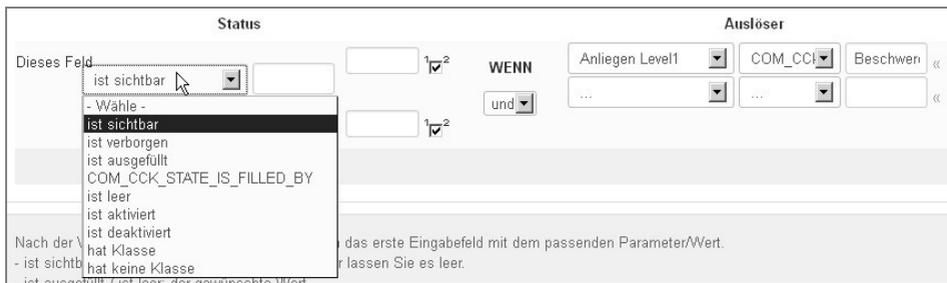


Bild 16.12: Bedingungen von Feldern 2.

Größtenteils sind die Status selbsterklärend. Um das noch einmal auf unser Beispiel zu beziehen: Unsere Bedingung ist, dass im Feld *Anliegen Level 1* der Wert gleich *Beschwerde* sein muss. Dann kann man auswählen, dass das Feld (nur zur Erinnerung, wir befinden uns gerade im Feld *Anliegen Level 2*) sichtbar ist, verborgen wird etc.

Logischerweise sorgen wir an dieser Stelle dafür, dass der Wert *ist sichtbar* ausgewählt wird. Somit ist dieses Feld auch nur dann sichtbar, wenn die entsprechende Bedingung erfüllt ist.

Nur am Rande und zur Vertiefung: Wenn Sie mit Textfeldern arbeiten würden, dann könnte so eine Abhängigkeit zum Beispiel auch so genutzt werden, dass bei Eintragen eines bestimmten Wertes in ein Textfeld in einem anderen Textfeld ein anderer Wert angezeigt wird. Wie so oft haben Sie mit SEBLOD einfach unzählige Möglichkeiten.

### 16.2.6 Das Beschwerdeformular fertigstellen

Sie haben auf den letzten Seiten so viel Wissen angehäuft, dass ich Ihnen nun bei den letzten Schritten des Beschwerdeformulars nicht mehr auf der Schulter sitzen muss, oder?

Legen Sie die Abhängigkeiten für alle weiteren Felder nach dem Beispiel fest, das ich Ihnen erläutert habe. Vermutlich werden Sie jetzt auch verstehen, warum ich die ganze Zeit darauf gepocht habe, dass Sie Ihre Feldnamen sauber vergeben, denn jetzt wird es unter Umständen kompliziert, wenn Sie mehrere Level bzw. Ebenen miteinander verknüpfen wollen.

Zum Abschluss möchte ich ein weiteres Mal darauf hinweisen, dass sowohl das Kontaktformular als auch das Beschwerdeformular jeweils sehr einfache Beispiele sind, die sich massiv erweitern lassen.

Sie haben nun die ersten Schritte gelernt und haben mit diesem Wissen die Möglichkeit, nahezu alle Arten von Formularen problemlos umzusetzen. Ab sofort brauchen Sie dafür sicherlich keine eigene Komponente mehr, alles lässt sich mit SEBLOD bauen.

Viel Spaß dabei ...



# 17

## Beziehungslisten

Manchmal möchten Sie vielleicht Listen in einem Beitrag ausgeben, die von dem Beitrag selber abhängen. Beispielsweise möchten Sie alle Beiträge eines Autors anzeigen. Das geht natürlich auch in SEBLOD, erfordert allerdings einige Vorüberlegungen und Bedingungen. Es funktioniert nur, wenn es sich um Beiträge handelt, und nur, wenn Sie dabei einen Wert übergeben, der in der URL steht. Ich möchte Ihnen das an einem kleinen, einfachen Beispiel veranschaulichen:

Zunächst lege ich zwei neue Kategorien an. Die eine nenne ich *Autoren*, die andere *Beiträge*. Danach wechsele ich zu *Formulare & Inhaltstypen* und lege zunächst einen neuen Inhaltstyp namens *Autor* an. Im *Admin Formular* lege ich ein *Textfeld* an, das ich als *Standard - Artikel - title* abspeichere, also als Beitragstitel. Dazu geselle ich das bereits fertige Feld *Article Category Id*, das Sie finden, wenn Sie rechts den ersten Filter auf *-Article* setzen. Dann wird das Kategoriefeld aufgelistet, und Sie können es in Ihr Formular hinüberziehen. Wählen Sie die *Nummer 2* aus und setzen Sie als Standardwert die Kategorie *Autor*.

Unter der *Nummer 1* setzen Sie das Feld auf *Verborgnen*. Sie speichern dann alle neuen Beiträge zu den Autoren automatisch unter der gleichnamigen Kategorie ab, ohne dass Sie sich darüber Gedanken machen müssen. Ein versehentliches Ändern der Kategorie ist nicht möglich, da das Feld gar nicht sichtbar ist. Unter *Einleitung/Intro* wählen Sie das gerade angelegte Feld *Autor* aus und unter *Inhalt* ebenfalls. Hier

kommt ein neues Feld hinzu: Wählen Sie im *Typ* das *Modul* aus. Als *Modus* setzen Sie *Position*, und unter *Position* oder *Name, Titel* wählen Sie die *Position beziehung*. Speichern und schließen Sie nun dieses Formular.

Wir benötigen ein zweites Formular. (Wobei Sie auch den vorhandenen *Article*-Inhaltstyp benutzen und entsprechend modifizieren können. Der Einfachheit halber und um das Prinzip zu verdeutlichen, lege ich aber einen neuen Inhaltstyp an). Für diesen Inhaltstyp benötigen wir in unserem Beispiel nur das *Admin Formular*. Dort legen Sie bitte zunächst ein *Article-Title*- und wieder ein *Article Category Id*-Feld an – dafür können Sie wieder die bereits vorhandenen Felder unter *-Article* benutzen. Dazu gesellt sich *Artikel (Ähnliche)*, das Sie in der Gruppe *Joomla!* des *Typs* finden. Unter *Kategorien* wählen Sie *Autoren* aus. Wenn Sie das alles speichern, erhalten Sie ein Auswahlfeld Ihrer Autoren, wenn Sie neue Beiträge anlegen, vorausgesetzt, Sie haben bereits ein paar Autoren angelegt.

**Bild 17.1:** Beitrag, in dem die Autoren direkt ausgewählt werden können.

Damit wir alle Beiträge des jeweiligen Autors anzeigen lassen können, benötigen wir zunächst einen *Listen & Suchtyp*. Ich habe ihn *Alle Beiträge* in meinem Beispiel genannt. Unter dem bereits vorhandenen Feld *CCK*, das *SEBLOD* benötigt, schieben Sie bitte das vorhandene Feld mit der *Autorenauswahl* hinein, also das *Joomla!-Artikel-Feld (Artikel (Ähnliche))*, das Sie zuvor angelegt haben.

Unter der *Nummer 2 - Live + Live Wert* wählen Sie in der ersten Spalte *Variable* aus und klicken dann auf *Konfigurieren*. Der *Typ* ist ein *Int*, und in *Variable* schreiben Sie *id*. Alles andere lassen Sie so stehen. Weder *Mehrfach* noch *Standard Wert* oder eine *Ver-schlüsselung* benötigen wir.

Unter *Liste* wählen Sie in *COLUMN-A* das *Article Title*-Feld aus.

Bevor ich Ihnen die Zusammenhänge erkläre, benötigen wir ein Modul, um unsere Suchliste anzeigen zu können. Denn Sie hatten zuvor bei den Autoren ja ein *Modul-Feld* eingebaut.

Dazu wechseln Sie bitte in die Modulübersicht von Joomla! und wählen das *SEBLOD*-Modul *SEBLOD-Liste* aus. In *Suchtyp* wählen Sie die eben angelegte Suchliste aus. In meinem Beispiel ist das *Alle Beiträge*. In *Position* tragen Sie *beziehung* ein und veröffentlichen das Modul. Eventuell müssen Sie es unter *Menüzuweisung* noch *Allen Seiten* zuweisen.

Nun brauchen Sie noch einen Menüverweis, damit Sie die Früchte Ihrer Arbeit sehen können. Dazu wechseln Sie in das Menü und legen ein *Kategorieblog* auf die »Autoren« an.

Wenn Sie nun im Frontend auf das Kategorieblog der Autoren klicken, sehen Sie ein normales Blog mit Ihren Autoren. Klicken Sie einen einzelnen Autor an (*Titel verlinken* muss auf *Ja* stehen, oder Sie haben einen *Weiterlesen*-Button im Beitrag), sehen Sie den Autor mit all seinen bisher geschriebenen Beiträgen.



Bild 17.2: Beziehungslisten in SEBLOD.

## 17.1 Beziehungs-Weise

Was genau ist passiert?

Wir haben zunächst zwei verschiedene Inhaltstypen angelegt, aber die beiden durch das *Artikel (Ähnliche)* miteinander verbunden. Somit gibt es im Beitragsinhaltstyp einen direkten Bezug zu den Autoren. Per Menüverweis werden die Autoren aufgerufen. Nach Anklicken eines einzelnen Autors haben Sie dessen ID in der URL stehen:

```
001 http://www.irgendeinedomain.de/index.php?option=com_
content&view=article&id=12:axel&catid=10&Itemid=108
```

Diese ID (*id=12*) wird im Suchtyp verwertet. Sie haben als Variable die *id* vom Typ *Int* übergeben. *Int* steht übrigens für Integer – einen ganzzahligen Wert. Mit dieser ID lässt sich der Autor klar und einwandfrei bestimmen, und in der *Liste* werden alle Beitragstitel zu dieser ID, zu diesem Autor, ausgegeben.

Die Ausgabe selber jedoch findet im Modul statt. Und die Modulposition, in der die Suchliste veröffentlicht wird, wurde zuvor im Inhaltstyp *Autor* zugewiesen.

Natürlich funktioniert das alles auch mit eingeschaltetem SEO. Die ganzen Variablen werden auch weiterhin vergeben – Sie sehen das lediglich nicht, da SEO die URLs anders darstellt.

**Keine Ergebnisse?**

Bei Autoren, die noch nichts geschrieben haben, erhalten Sie die Mitteilung, dass »keine Ergebnisse« vorliegen. Um das zu verhindern und stattdessen einfach eine leere Liste anzuzeigen, wählen Sie in *Listen & Suchtypen* die *Konfiguration* aus und stellen ganz unten unter *Kein Resultat* im *Nachrichten-Stil Keine* ein.

# 18

## Vorbelegung der Sprache bei Mehrsprachigkeit

Wenn Sie Mehrsprachigkeit auf Ihrer Seite haben und Ihren Autoren die aktuell ausgewählte Sprache als Standardwert im Feld für die *Inhaltssprache* übergeben wollen, werden Sie feststellen, dass sich kein Standardwert übergeben lässt. Entweder ist das Feld leer, oder es ist stets *Alle* ausgewählt. Mit einem kleinen Trick können Sie dennoch die aktuell ausgewählte Sprache übergeben. Im Beispiel arbeite ich mit zwei Sprachen: Deutsch und Englisch.

Zunächst müssen Sie in Joomla! die Mehrsprachigkeit einstellen, also Sprachinhalte eingeben und das Plug-in aktivieren, ebenso das Modul, damit Sie zwischen den Sprachen umschalten können.

Erstellen Sie nun zusätzlich ein einfaches Textfeld in der Seitenform Ihres Artikels. In meinem Beispiel habe ich es *Länderkürzel finden* genannt. Bei Speichern stellen Sie *Kein* ein. Wir benötigen dieses Feld lediglich zum Ermitteln der aktuell eingestellten Sprache und werten es sofort aus. Wir benötigen keine Speicherung.

Sie müssen nun *Nummer 2* anklicken und *Variable* auswählen. In der rechten Spalte klicken Sie *konfiguriere* an und tragen dann im modalen Fenster *lang* bei *Variable* ein. Alles andere lassen Sie, wie es ist.

Konstruktion 1

Typ: Zeichenkette

Mehrfach: Nein

Variable: lang

Ignore "0 / NULL": Nein

Standard Wert:

Encryption: - Kein

**Bild 18.1:** Abfrage der aktuell eingestellten Sprache.

Joomla! übergibt die aktuell eingestellte Sprache an die Variable *lang*. Sehen können Sie das in der URL bei deaktivierter SEO:

```
001 http://localhost:8080/j32/index.php?lang=de
```

Sie fragen den Inhalt dieser Variablen ab, und der Wert wird als Standardwert in das Feld eingetragen.

Nun wechseln Sie zur *Nummer 5* und klicken beim Feld *Inhaltssprache* auf *Hinzufügen*. Sie stellen eine Abhängigkeit zum eben angelegten Feld, in dem die aktuell benutzte Sprache gespeichert ist, her. Dazu fügen Sie an erster Stelle

```
001 ist ausgefüllt en-GB #form# WENN Länderkürzel finden is Equal en
```

ein. Dabei gehe ich von den Standardeinstellungen bei den Sprachen aus. *en-GB* ist das Sprach-Tag. Sie müssen stets das Tag übergeben, auch dann, wenn Sie im Auswahlfeld von *Inhaltssprache* etwas anderes stehen haben. Bei mir steht hier:

- Alle
- Deutsch
- English (UK)

Das Sprach-Tag führt aber dazu, dass die richtige Sprache im Auswahlfeld vorbelegt wird.

In der zweiten Position geben Sie

```
001 ist ausgefüllt de-DE #form# WENN Länderkürzel finden is Equal de
```

ein. Wichtig ist, dass Sie den Haken nicht in den Checkboxen setzen. Denn dann führt eine Rücküberprüfung stets zu dem Ergebnis, dass mindestens eines der Länderkürzel nicht zutrifft und somit der Inhalt gelöscht wird, was ja in unserem Fall nicht Sinn und Zweck ist.

ABHÄNGIGE ZUSTÄNDE Speichern & Schließen Zurücksetzen Abbrechen

### Article Language

# 1.

	<b>Status</b>		<b>Auslöser</b>	
Dieses Feld	ist ausgefüllt <input type="checkbox"/> en-GB	#form# <input type="checkbox"/> 1 <input type="checkbox"/> 2	WENN und	Länderkürzel finden <input type="checkbox"/> is Equal / <input type="checkbox"/> en
	... <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> 1 <input type="checkbox"/> 2		... <input type="checkbox"/> ... <input type="checkbox"/>

# 2.

	<b>Status</b>		<b>Auslöser</b>	
Dieses Feld	ist ausgefüllt <input type="checkbox"/> de-DE	#form# <input type="checkbox"/> 1 <input type="checkbox"/> 2	WENN oder	Länderkürzel finden <input type="checkbox"/> is Equal / <input type="checkbox"/> de
	... <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> 1 <input type="checkbox"/> 2		... <input type="checkbox"/> ... <input type="checkbox"/>

**Bild 18.2:** Abhängigkeiten der Sprachvorbelegung.

Nun legen Sie noch das erste Feld, *Länderkürzel finden*, auf *Verborgen*. Dennoch wird das Länderkürzel ermittelt und kann ausgewertet werden, es taucht aber im Eingabebereich der Artikelform nicht auf. Wenn nun Ihre Autoren einen neuen Beitrag schreiben möchten, erhalten sie als Vorauswahl Ihre aktuell eingestellte Sprache, die Sie nachträglich natürlich auch ändern können, wenn Sie in einer anderen Sprache schreiben möchten.



# 19

## Neue und andere Module

In der Modulübersicht von Joomla! (*Erweiterungen/Module*) finden Sie vier Module von SEBLOD, die Sie verwenden können. Die Module bieten teilweise sehr interessante Aspekte.

Bis auf die Breadcrumbs sind bei allen anderen Modulen die Einstellungen identisch. Auf die Joomla!-spezifischen gehe ich hier nicht näher ein, die *Overrides (Live/Variation)* sind bereits im Kapitel 8 »Alle Felder« besprochen worden.

Neu ist der Tab *URL Assignment*. Sie können die URL prüfen und das Modul bei bestimmten Variablen verstecken bzw. anzeigen. Beispielsweise können Sie auf die gewählte Sprache prüfen, das funktioniert übrigens auch bei eingeschaltetem SEO. Geben Sie beispielsweise bei *URL Assignment (Hide)* `lang=de` ein. Wenn jemand die englische Sprache eingeschaltet hat, dann wird das Modul nicht angezeigt. Das Gleiche bei *URL Assignment (Show)* angeben, und es wird angezeigt.

Eine praktischer Anwendungsfall kann sein, dass Sie eine internationale Website mit Autoren aus verschiedenen Ländern betreiben. Vielleicht haben Sie ein Eingabeformular, das in bestimmten Sprachen angezeigt werden soll, obwohl das Modul selber der Sprache nicht zugeordnet ist.

Oder das Modul soll nicht angezeigt werden, wenn der User einen bestimmten Beitrag angeklickt hat. Dann prüfen Sie die *ItemID*. Oder Sie möchten bestimmte

Kategorien ausschließen, was die *catid* wäre. Das ist auch dann interessant, wenn vielleicht kein bestimmter Menüverweis zugrunde liegt, weil Sie auf *Weiterlesen* geklickt haben, oder Sie haben zwei unterschiedliche Bedingungen, wie die Beitrags-ID und die Kategorie-ID, was nicht unbedingt immer mit der direkten Menüzuteilung funktioniert. Sie haben mit diesem *URL Assignment* zusätzliche Möglichkeiten, ein Modul unter bestimmten Voraussetzungen anzuzeigen oder zu verstecken.

Da Sie Module nicht nur in den dafür vorgesehenen Modulpositionen des Templates, sondern auch unter/in/über einem Beitrag mit *loadposition*<sup>33</sup> anzeigen lassen können, erhalten Sie weitere spannende Gestaltungsmöglichkeiten. Ein Beispiel dazu finden Sie im Kapitel 17 »Beziehungslisten«.

## 19.1 Modul: From – SEBLOD

Mit diesem Modultyp erhalten Sie das *Seitenformular* zum ausgewählten Inhaltstyp.

Somit können Sie statt eines Menüverweises zur Eingabe neuer Inhalte das Modul anzeigen. Zusammen mit dem *URL Assignment* ergeben sich da vielleicht ein paar interessante Möglichkeiten ...

Bei mir klappten leider Pull-down-Felder wohl auf, aber gleich wieder zu, wenn ich die Maus in den Auswahlbereich bewegen wollte, bevor ich dort etwas auswählen konnte. Wenn Sie die linke Maustaste festhalten, können Sie diesem Effekt entgegensteuern. Eventuell sollten Sie einen freien Text mit diesem Hinweis für Ihre Autoren in das Formular eintragen, sollten Sie den gleichen Effekt bei sich auch beobachten.

## 19.2 Modul: List – SEBLOD

Mit diesem Modul zeigen Sie eine zuvor angelegte Liste an, die Sie im *Suchtyp (Liste)* auswählen. Wenn Sie einen Beschreibungstext für Ihre Liste eingegeben haben, haben Sie die Möglichkeit, diese Beschreibung im Modul explizit auszugeben oder die globale Einstellung zu wählen. Wenn der Beschreibungstext ausgegeben werden soll, wählen Sie, ob er über oder unter den Feldern angezeigt werden soll.

Interessant ist der »*More*« *Link*, mit dem Sie zu einem vorhandenen Menüverweis einen Link setzen können. Dieser wird am Ende der Liste angezeigt.

---

<sup>33</sup> Im Beitrag in der WYSIWYG-Ansicht diesen Code eingeben: `{loadposition mein_positionname}`  
Danach ein Modul an dieser Stelle positionieren (*mein\_positionname*).

List - SEBLOD	
Titel	Kategorie
Beitrag Nummer 2	Apfelkuchen
Damit wir was zum finden haben	Bauklötze
Der erste Beitrag	Apfelkuchen
Noch einen	Uncategorised
Testusers Beitrag	Bauklötze

"More" Link → View All

Bild 19.1: »More« Link am Ende der Liste.

Zu dem »More« Link können Sie eine CSS-Klasse angeben, um den Verweis individuell zu formatieren. Mit »More« Link Variables können Sie an die URL eine oder mehrere Variablen hängen. Beispielsweise können Sie so die Sprache wechseln, wenn Sie `lang=en` eintragen. Ist die bisherige Sprachauswahl Deutsch, wird automatisch zu Englisch gewechselt. Sie können aber auch IDs oder andere Variablen, auch mehrere, mitgeben. Wollen Sie mehrere Variablen an die URL hängen, müssen Sie ab der zweiten am Anfang ein kaufmännisches Und (&) schreiben. Beispielsweise:

```
001 lang=en
002 &catid=12
003 &itemid=5
```

Ob Sie in jede Zeile eine Variablendeklaration oder alles in eine Zeile schreiben, ist hingegen egal.

In *Zählen* geben Sie an, wie viele Datensätze in der Liste ausgegeben werden.

Die *Standardsortierung* ist, so denke ich, klar. Die *endgültige Sortierung* erscheint mir persönlich überflüssig, da tatsächlich nur eine Sortierung angezeigt wird. Wenn Sie in der *endgültigen Sortierung Shuffle* einstellen, werden die ausgewählten Listeneinträge jeweils in unterschiedlicher Reihenfolge dargestellt.

Damit hebeln Sie eine Einstellung bei *Sortierung* aus. Wählen Sie *Random* aus, werden die Listeneinträge stets so angezeigt, wie Sie sie in *Sortierung* eingestellt haben, es werden aber zufällig angezeigte Listeneinträge angezeigt. Sehen können Sie das nur, wenn Sie in *Zähler* eine kleinere Zahl als tatsächlich vorhandene Listeneinträge einstellen.

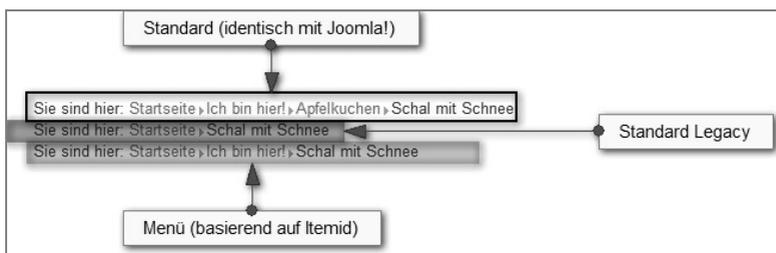
## 19.3 Modul: Search – SEBLOD

Sie können mit diesem Modul eine Suche anzeigen, die Sie angelegt haben. Wählen Sie in *Suchtyp (Liste)* eine Ihrer angelegten Suchen aus. Im *Menüelementziel* müssen

Sie einen Menüverweis auswählen, mit dem Sie eine Suche oder Liste darstellen können, die das Suchergebnis nutzt.

## 19.4 Modul: Breadcrumbs – SEBLOD

Im Großen und Ganzen handelt es sich dabei um das Joomla!-eigene Modul für die Breadcrumbs<sup>34</sup>. Sie können hier noch eine CSS-Klasse angeben und haben ein paar Möglichkeiten zur Pfadgestaltung. Je nachdem, wie Sie Ihre eigenen Formulare und Listen verschachteln, kann es mit den hauseigenen Breadcrumbs eventuell nicht befriedigend enden. Das SEBLOD-Modul bietet etwas mehr Möglichkeiten.



**Bild 19.2:** Die verschiedenen Pfad-Einstellungen mit aktivierter »Startseite«.

Die Pfad-Einstellungen:

- *Standard (identisch mit Joomla!)* – Gibt Ihnen den Pfad wie gewohnt aus.
- *Standard Legacy* – Gibt Ihnen eine verkürzte Darstellung des Pfades wieder. Eventuelle Pfade zwischen der Startseite und der aktuellen Seite werden weggelassen.
- *Menü (basierend auf Itemid)* – Zeigt Kategorien nicht an. In der Abbildung ist *Ich bin hier!* ein Beitrag, bei *Apfelkuchen* handelt es sich um eine Kategorie. Der Beitrag wird im Pfad angezeigt, die Kategorie nicht.

### 19.4.1 Ein Easteregg?

In meiner Übersetzung für das Modul taucht der Begriff »Paniermehl« auf. Ob es sich da wohl um ein *Easteregg*<sup>35</sup> handelt?

<sup>34</sup> Zu Deutsch »Brotkrumen«. Wie einst bei Hänsel und Gretel werden Brotkrumen ausgelegt, um den Weg zurück zu finden, also den Pfad vom aktuellen Inhalt zur Startseite.

<sup>35</sup> Als »Easteregg« werden versteckte und undokumentierte Besonderheiten in Softwareprogrammen oder Spielen bezeichnet. Meist handelt es sich dabei um witzige Merkmale oder Geheimlevels.



Bild 19.3: Das Paniermehl-Modul.

## 19.5 Module ändern

Die bestehenden Joomla!-Module können Sie nicht ändern. Sie können aber neue Listen und Formulare anlegen, die bestehende Aufgaben übernehmen, und sie nach Ihrem Gusto anlegen. Möchten Sie die neuesten Beiträge mit einem kleinen Bild und der Anzeige der zugehörigen Kategorie darstellen? Dann »bauen« Sie sich eine entsprechende Liste in SEBLOD und zeigen diese mit dem SEBLOD-Modul für Listen an. Da Sie statt der Joomla!-Beitragsmasken die von SEBLOD für neue Beiträge nutzen können, haben Sie relativ bequem die Möglichkeit, Module, die zur Anzeige von Beiträgen dienen, nach Ihren Vorstellungen zu ändern. Sie müssen lediglich den Umweg über die Listen und die SEBLOD-Vorgaben nehmen.

So können Sie nahezu jedes Joomla!-Modul neu erfinden und Ihrer Website ein unverwechselbares Aussehen geben.





## Multisites mit SEBLOD

Die Multisites sind auf den ersten Blick sehr verwirrend, und ich selber habe einige Zeit benötigt, bis ich das Konzept verstand. Wenn aber erst einmal der berühmte Groschen gefallen ist, dann sind die Multisites mit SEBLOD eine sehr interessante Möglichkeit, entweder mehrere Domains mit einer Joomla!-Datenbank oder mehrere Aspekte einer Website auf mehrere Subdomains aufzuteilen.

Große Webseiten können beispielsweise bequem aufgeteilt werden. Angenommen, Sie haben eine große Website über Hunde, Katzen und Pferde und dazu einen Onlineshop. Statt einer Website für alles machen Sie vier Websites daraus. Eine für Hunde, eine für Katzen, eine für Pferde und eine für E-Commerce. Der Vorteil ist, dass Sie nach wie vor eine Hauptseite, eine *Mastersite*, haben, von der aus Sie alles verwalten, einstellen und entsprechend zuordnen, aber vier weitere Domains, Subdomains oder Aliasseiten haben, die zwar alle auf die Hauptseite verlinkt sind, aber Unterschiedliches anzeigen.

Oder Sie erstellen ein Intranet<sup>36</sup> und möchten ein Extranet<sup>37</sup> integrieren. Dann liegt es doch nahe, das sauber zu trennen.

---

<sup>36</sup> Das Intranet ist ein nicht öffentliches Rechnernetz, beispielsweise ein firmeninternes Netz.

<sup>37</sup> Das Extranet ist Bestandteil des Intranets, aber für eine Personengruppe festgelegt. Beispielsweise könnte es sich dabei um einen Bereich für Lieferanten oder für die Führungsetage handeln.

Oder Sie haben eine Mehrsprachenseite und möchten nicht die Mehrsprachigkeit von Joomla! nutzen, sondern für jede Sprache eine eigene Domain einrichten.

Das dahinter stehende Konzept ist, dass Sie beliebig viele Domains, Subdomains oder Aliasdomains erstellen, die alle auf die Haupt-Joomla!-Installation verweisen. Das machen Sie in aller Regel in der Domainverwaltung Ihres Hosters.

frankreich.chalambar.de	 Website unter mastersite
italien.chalambar.de	 Website unter mastersite
multimaster.chalambar.de	 Website unter mastersite
spanien.chalambar.de	 Website unter mastersite

**Bild 20.1:** Drei Subdomains, die alle auf das gleiche Verzeichnis der Hauptdomain zeigen, in dem Joomla! liegt (Auszug aus Plesk – bei Ihnen könnte die Ansicht abweichen).

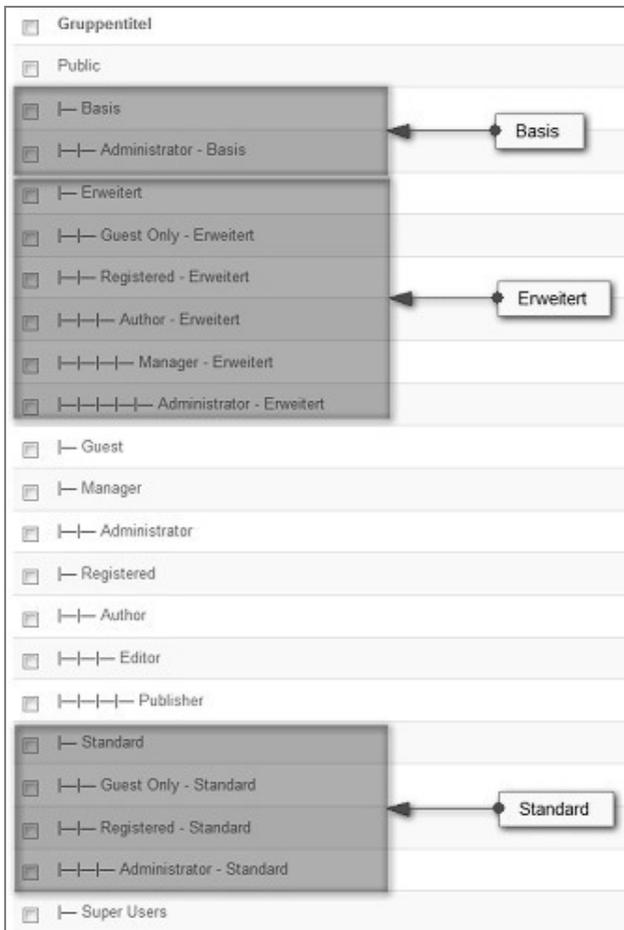
Wenn Sie dann eine neue Seite in SEBLOD anlegen, werden automatisch neue Benutzergruppen und Zugriffsebenen angelegt und der jeweiligen Multisite zugeordnet. Letztlich haben Sie ein entsprechendes Rechtesystem, mit dem Sie das alles verwalten. Allerdings bietet Ihnen SEBLOD darüber eine deutlich bessere Übersicht und Verwaltung und die eine oder andere Möglichkeit mehr, die auf dem herkömmlichen Weg schwerer umzusetzen wäre. Und SEBLOD ordnet die jeweiligen Benutzergruppen und Zugriffsebenen einer eindeutigen Website zu. Das können Sie so mit Joomla! nicht umsetzen.

## 20.1 Einrichtung einer neuen Multisite

Rufen Sie in SEBLOD die *Seiten* auf. Sie erreichen diesen Bereich am einfachsten über das *Konstruktion-Menü*. Gehen Sie über den Komponentenaufruf, klicken Sie wahlweise links im Menü auf *Seiten* oder im Iconbereich auf *Seiten Manager*. Danach ein beherrzter Klick auf *Neu*.

Die erste Auswahl fragt nach *Basis*, *Standard* oder *Erweitert*.

SEBLOD steuert die Multisites und zugehörige Beiträge, Kategorien, User, Menüs etc. über die Rechteverwaltung. Für jede einzelne Multisite werden mindestens zwei neue Benutzergruppen, zwei neue Benutzer und mindestens eine neue Zugriffsebene angelegt.



**Bild 20.2:** Die neuen Benutzergruppen.

- *Basis* geht davon aus, dass Sie eine relativ überschaubare Website haben, die mit einem registrierten Benutzer und einem Administrator auskommt.
- *Standard* geht bereits von einer etwas komplexeren Website aus, die nicht nur einen unterschiedlichen registrierten Bereich hat, sondern auch ein eigenes öffentliches Frontend.
- *Erweitert* geht von einer komplexen Website mit öffentlichem Frontend, Bereichen für registrierte Benutzer und einem Autorensystem aus. Für das Backend wird neben dem Administrator auch ein Manager angelegt.

Die eigentliche Multisite ändert sich zunächst nicht – es geht lediglich darum, dass SEBLOD Ihnen etwas Arbeit abnimmt, indem es die entsprechenden Berechtigungen für Sie voranlegt.

Sollte Ihnen im Moment noch unklar sein, was genau hier passiert – ich werde gleich die Fäden im Wirrwarr ordnen.

Doch zunächst klicken Sie auf *Erstellen*, nachdem Sie für den ersten Versuch *Standard* auswählen. Diese Auswahl ist auch voreingestellt. In der *Standard*-Auswahl können wir sowohl das Frontend als auch den Bereich für registrierte Benutzer direkt beeinflussen.

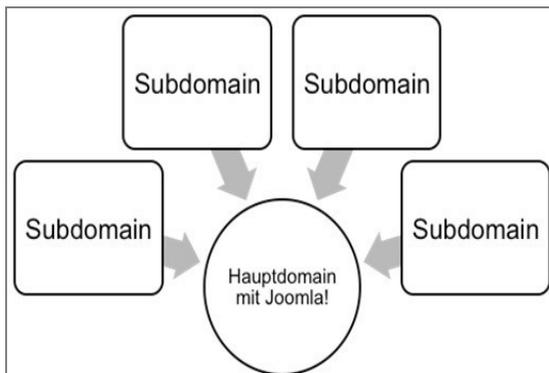
Der *Titel* ist für Sie und wird auch in der Rechteverwaltung benutzt und sollte eine konkrete Aussage über den Inhalt der Multisite machen.

Der *Seiten Name* hingegen wird in der Site selber benutzt und kann mit *Inkludiere Seiten-Name* wahlweise *Vor* oder *Nach dem Seitentitel* ausgegeben werden oder überhaupt nicht. Diese Einstellung ist identisch mit der Joomla!-Einstellung für den Seitennamen, wo Sie auch einstellen können, ob er im Browserfenster oder in den Browser-Tabs angezeigt werden soll.

Neben dem *Titel* finden Sie die *Url*. Dort geben Sie die Domain der Multisite ein.

Als Beispiel simuliere ich eine Reisesite. Dafür habe ich drei Domains angelegt:

- italien.domain.de
- frankreich.domain.de
- spanien.domain.de



**Bild 20.3:** Eine Joomla!-Datenbank und beliebig viele Subdomains, die darauf verweisen.

#### URL-Eingabe

Lassen Sie bei der *URL-Eingabe* das »http://« am Anfang weg, da SEBLOD es auf jeden Fall vor die URL schreibt. Wenn Sie es dort hinschreiben, steht es zweimal am Anfang der URL, was natürlich nicht funktioniert.

Alle diese Domains – nur noch einmal zur Erinnerung – zeigen auf das gleiche Verzeichnis: die Joomla!-Installation. Wenn Sie diese Domains direkt in Ihrem Browser

aufrufen würden, würden Sie also jedes Mal die gleiche Website sehen. SEBLOD legt nun seine Berechtigungen an, die alle mit dem *Seitennamen* enden, und kann somit diese Berechtigungen einwandfrei der in der *URL* angegebenen Website zuordnen. Wenn Sie nun beispielsweise einen neuen Menüverweis anlegen und eben dieser neuen Zugriffsebene zuordnen, dann haben Sie einen Menüverweis, der nur auf dieser zugeordneten Website angezeigt wird.

Sie können für diese Multisite eine eigene *Meta-Beschreibung* und *Meta-Schlüsselworte* angeben. Andernfalls wird die Metabeschreibung der Mastersite genommen. Als Mastersite bezeichne ich die eigentliche Joomla!-Installation.

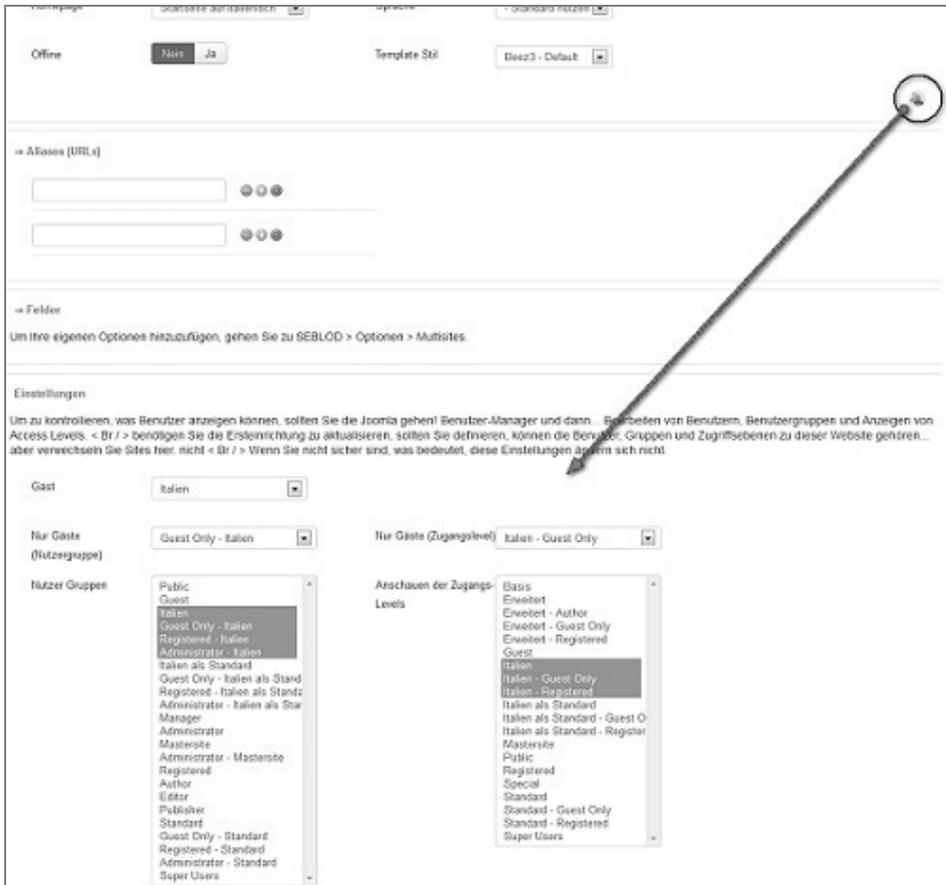
- In *Homepage* tragen Sie die Startseite ein. Lassen Sie die Einstellung auf *Standard nutzen* stehen, wird die Defaultstartseite benutzt. Oder Sie stellen eine individuelle Startseite für diese Domain ein.
- In *Sprache* können Sie diese Domain, diese Multisite, einer bestimmten Sprache zuordnen, sofern Sie Mehrsprachigkeit nutzen. Auch können Sie für diese Site ein eigenes Template zuordnen.

Falls Sie jetzt denken, dass das doch sowieso alles möglich ist in Joomla!, haben Sie natürlich recht. Aber dafür müssten Sie auf jeder einzelnen Domain ein eigenes Joomla! installieren und alle einzeln administrieren. Auch können die einzelnen Domains nicht kooperieren, beispielsweise dass alle Benutzer sich überall einloggen können. In SEBLOD haben Sie nur eine einzige Datenbank und eine einzige Joomla!-Installation.

Sie können zu jeder einzelnen Multisite, die Sie angelegt haben, eine eigene Seite in SEBLOD anlegen und entsprechend individuell gestalten. Aber vielleicht möchten Sie auch zwei oder mehr Multisites mit den gleichen Einstellungen in SEBLOD übernehmen. Dann können Sie in *Aliases (URLs)* weitere Multisites angeben, die mit den gleichen Einstellungen arbeiten wie die, die unter *Url* angegeben ist. Angenommen, ich trage in *Url*: *frankreich.domain.de* und unter *Aliases (URLs)*: *italien.domain.de* ein, dann gelten die Einstellungen für beide Domains/Multisites, und zwar sowohl die unmittelbaren Einstellungen in dieser Eingabemaske als auch die Berechtigungen.

Zu den *Feldern* schreibe ich weiter unten etwas ausführlicher. Wenn Sie entsprechende *Felder* angelegt haben, können Sie sie hier auswählen und der Multisite explizit zuordnen.

Wenn Sie auf das kleine *Benutzericon* klicken, das Sie erst nach dem ersten Speichern sehen können und das Sie dann im Bereich *Optionen* unten rechts vorfinden, erhalten Sie ganz unten einen Überblick über die *Einstellungen* der einzelnen *Benutzergruppen und Zugriffsebenen*. Sie können diese hier ändern, indem Sie sie entsprechend auswählen. Mit Großschreibtaste und linker Maustaste können Sie mehrere Einträge markieren. Sie finden hier die standardmäßig angelegten Berechtigungen und die von SEBLOD. Wenn Sie bereits eigene Berechtigungen angelegt haben, finden Sie diese hier auch und können somit sehr individuell das Rechtesystem den einzelnen Multisites zuordnen.



**Bild 20.4:** Die Einstellungen der Berechtigungen für jede Multisite auf einen Blick.

## 20.2 Die Multi-Berechtigungen

Es wird Zeit, dass wir die neuen Berechtigungen konkret in der Praxis einsetzen, um das Prinzip zu durchleuchten.

Sie haben generell folgende Zugriffsebenen:

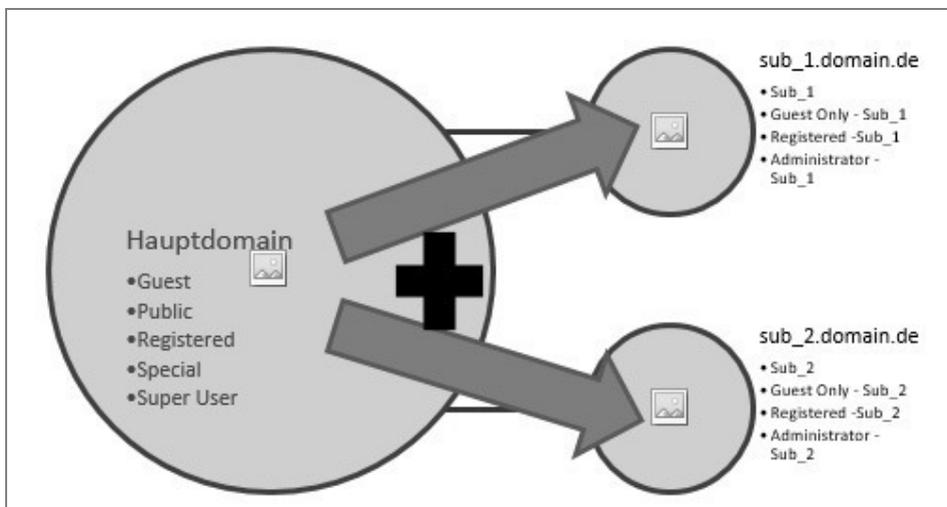
- Öffentlich
- Registriert
- Administrator

In den einzelnen Bereichen können sich weitere Untergruppen, wie beispielsweise die Autoren oder Manager, befinden. Diese Hauptbereiche haben Sie generell auch in den Multisite-Berechtigungen.

*Standard* und *Erweitert* haben eine *Guest Only*-, eine *Registered*- und eine *Administrator*-Gruppe. *Erweitert* hat zusätzlich den *Autor* und den *Manager*. *Basis* hat nur den *Administrator*.

*Basis*, *Standard* und *Erweitert* sind jeweils die Zugriffsrechte auf alle genannten Gruppen. So können Sie beispielsweise Menüs anlegen, die von allen gesehen werden können – egal ob eingeloggt oder nicht –, indem Sie die Zugriffsebene *Standard* setzen. Oder Sie legen Menüs an, die nur für registrierte und eingeloggte Benutzer sichtbar sind, indem Sie die Zugriffsebene *Registered - Standard* setzen, oder nur für Administratoren mit *Administrator - Standard*. *Standard* bezieht sich auf die Auswahl, die Sie beim Anlegen einer neuen Multisite getroffen haben. Statt des Worts *Standard* steht in der Zugriffsliste und den Benutzergruppen der *Titel*.

Dadurch, dass Sie bei jeder angelegten Multisite ein eigenes Rechtesystem nur für diese eine Site kreieren, haben Sie alle Freiheiten, Beiträge, Menüs, Kategorien und Module den einzelnen Multisites zuzuordnen.



**Bild 20.5:** Die Standardrechte plus eigene Rechte für jede Multisite.

Alles, was mit den Standardzugriffen von Joomla! versehen ist, wird auf allen Multisites angezeigt. Somit können Sie Daten nur für bestimmte Seiten sowie auch für alle Seiten zugänglich machen. Da Sie weiterhin eigene Gruppen definieren können, die unterschiedliche Zugriffsebenen erhalten, ist es auch möglich, dass Sie bestimmte Inhalte nur auf ausgewählten Multisites anzeigen lassen. Beispielsweise könnten Informationen über Schutzimpfungen nur auf Domains mit Reiseberichten zu relevan-

ten Ländern auftauchen, nicht jedoch, wenn Sie von Reisezielen in Deutschland berichten.

Schauen wir uns das Ganze am besten in der Praxis an: Dazu legen wir am einfachsten einen Beitrag an, den wir danach per Kategorieblog auf unsere Startseite bringen.

Um das Ganze für meine Reiseseiten vernünftig auseinanderzuhalten, habe ich meine Subdomain für die Italien-Reiseberichte mit dem *Titel Italien* versehen und mir passende Kategorien angelegt: *Italien, Spanien, Frankreich*. Für meine Mastersite möchte ich mit der Kategorie *Reisen* arbeiten.

Für den Anfang schreibe ich zwei Beiträge: einen für Italien und einen für das Reisen an sich. Nun wechseln Sie bitte in das *Main Menu*. Der vorhandene Eintrag *Home* (sofern Sie ein frisch angelegtes Joomla! haben) bietet sich als Startseite für die Mastersite weiterhin an. Markieren Sie einfach den »Reisen an sich«-Beitrag als Haupteintrag.

Danach legen Sie einen neuen Menüeintrag an. Als *Menüeintragstyp* bietet sich das *Kategorieblog* in unserem ersten Beispiel an. Als Kategorie wird *Italien* eingestellt. Bei der *Zugriffsebene* wählen Sie *Italien* aus. Nun das Ganze speichern und im Browser sowohl die Master-Domain als auch die Italien-Domain aufrufen. Sie sehen bei der Master-Domain lediglich *Home* als Startseite und einen einzigen Menüverweis. Auf der Italien-Domain haben Sie neben *Home* auch den eben angelegten *Italien*-Menüverweis.

So weit – so gut, wir haben nun einen Menüverweis, der nur auf einer der beiden Domains angezeigt wird. Wenn Sie jedoch mehrere Menüverweise angelegt haben, ist es ja sinnvoll, für jede Multisite ein eigenes Startmenü festzulegen. Wenn Sie das nun tun, werden Sie zu Ihrer Verwunderung vermutlich zweimal den gleichen Menüeintrag auf der Multisite sehen: einmal in der regulären Reihenfolge des Menüs und einmal ganz oben als Startseite.

Bei der Betrachtung des Quellcodes werden Sie auch feststellen, dass es sich tatsächlich um den gleichen Menüverweis handelt, der von SEBLOD dupliziert wird. Das ist natürlich weder schön noch sinnvoll. Abhilfe schaffen Sie, wenn Sie ein eigenständiges Menü mit entsprechendem Menümodul anlegen, das Sie nur der Zugriffsebene für die Multisite zuordnen.

So habe ich ein neues Menü namens *Italien* erstellt und es einem gleichnamigen Modul zugewiesen, das die Zugriffsebene *Italien* erhalten hat. Nun wird zum einen dieses Menü tatsächlich nur und ausschließlich auf der passenden Multisite angezeigt, und Sie können auch einen *Homepage-Eintrag* festlegen, der nur einmal angezeigt, aber bei jedem Aufruf der Multidomain als Erstes ausgegeben wird.

Sie können nun Kategorien, Beiträge, Menüverweise, Module und Plug-ins nach Belieben Ihren Subdomains zuordnen und somit jede Website individuell gestalten. Alles läuft über eine Datenbank. Damit haben Sie auch die Möglichkeit, Ihre registrierten Benutzer den Seiten zuzuweisen. Diese müssen sich nicht mehrmals

anmelden, weil überall ein eigenes Joomla! läuft, sondern werden zentral verwaltet. Sie können einem einzelnen Benutzer eine oder mehrere Benutzergruppen zuordnen. Somit können Sie sehr individuell bestimmen, auf welchen Seiten ein Benutzer sich anmelden kann.

Ebenso verhält es sich mit Beiträgen oder Menüs. Sie können die Rechteverwaltung von Joomla! mit allen Möglichkeiten nutzen und erhalten somit mit Joomla! und SEBLOD ein mächtiges Werkzeug an die Hand.

## 20.3 Post von der Mastersite

Wenn Sie eine neue Multisite in SEBLOD anlegen, erhalten Sie einige E-Mails von Ihrer Hauptdomain. Je nachdem, was Sie bei der Seitenerstellung ausgewählt haben – *Basis*, *Standard* oder *Erweitert* –, bekommen Sie unterschiedlich viel Post. Grundsätzlich ist immer eine E-Mail mit dem Betreff *Seiten Erstellung auf [Titel Ihrer Multisite]* dabei, die für den *Administrator* der Site gedacht ist. Zudem erhalten Sie meist mehrere E-Mails mit dem Betreff *Details zum neuen Benutzer*.

Alle E-Mails erhalten einen neuen Benutzernamen und ein Passwort, stets in der gleichen Art angelegt:

001 Benutzername: [Benutzergruppe].[Titel der Multisite]

002 Passwort: [Benutzergruppe]

Mit diesen Zugangsdaten können Sie sich auf der entsprechenden Site anmelden und die verschiedenen Ansichten überprüfen.

Sie können aber auch sehen, welcher Benutzergruppe der jeweilige Benutzer zugewiesen ist, falls Sie noch Zweifel über die unterschiedlichen Rollen der Benutzer haben sollten. Sie können aber natürlich jederzeit neue Benutzer anlegen, die entsprechende Rechte auf den Multisites haben.

### **Achtung, Passwörter!**

Jemand, der SEBLOD kennt, kennt auch die standardgenerierten Zugänge von SEBLOD. Ändern Sie deshalb unbedingt die Passwörter!

## 20.4 Eigene Felder

Sie können jeder Multisite eigene Felder mitgeben, die Sie per Ajax<sup>38</sup>-Programmierung verarbeiten können. Wer nicht programmierkundig ist, für den gibt es auf der Produktseite von SEBLOD ein Plug-in<sup>39</sup> für 5 Euro, das diese Aufgabe übernimmt.

<sup>38</sup> Ajax = Aynchronous JavaScript and XML

<sup>39</sup> Plug-in: *SEBLOD Site (Multi-sites)*: <http://www.seblod.com/products/1720>

Rufen Sie zunächst die *Optionen* von SEBLOD auf. Sie finden sie, wenn Sie direkt auf *SEBLOD 3.x* in der *Komponentenauswahl* klicken. In den *Optionen* zunächst den Tab *Komponente* aufrufen und *Entwicklungs-Speicherung* auf *Ja* setzen. Nach dem Speichern wechseln Sie bitte in die *Felder* von SEBLOD.

**Bild 20.6:** Eigene Felder anlegen – in den Optionen die *Entwicklungs-Speicherung* einstellen.

Bei den Feldern habe ich ein Textfeld angelegt, in dem ich festhalten möchte, wann ich die jeweilige Multisite erstellt habe. Bei der Speicherung gilt es zu beachten, dass Sie als Formular *Entwicklung* auswählen und als Ort:

```
json[options][ein_Name_Ihrer_Wahl]
```

eintragen. Als Beispiel:

**Bild 20.7:** Speichern unter *Entwicklung* und *json*.

Sie können später auf dieses Feld, das entsprechend in der Datenbank angelegt wird, mittels `Json`<sup>40</sup> und entsprechender Programmierung zugreifen.

Idealerweise setzen Sie das Feld selber auf *Off* (die Einstellung finden Sie oben rechts), denn dieses Feld wird in keinem Formular oder Listentyp benötigt. SEBLOD erkennt durch den Speicherort *Entwicklung* auch, dass es sich um ein besonderes Feld handelt.

<sup>40</sup> JSON = JavaScript Object Notation, ein kompaktes und einfaches Datenformat

Wechseln Sie wieder in die *Optionen* und rufen Sie dort den Tab *Multisites* auf. Sie können nun das eben angelegte Feld auswählen.

The screenshot shows the Joomla! configuration interface for Multisites. The tabs at the top are: Generell, Komponente, Integration, Sprache, Medien, **Multisites**, Berechtigungen, and Seite. Under the 'Multiseiten System' section, there are two buttons: 'Ja' (selected) and 'Nein'. Below that, under 'Berechtigungen', there is a dropdown menu with 'Im Besitz' selected. Under the 'Optionen' section, there is a text input field containing 'angelgt\_am' and a dropdown menu. The dropdown menu is open, showing 'Angelegt am' (selected), '- Feld zufügen', and 'Angelegt am'.

**Bild 20.8:** Ein Feld für die Multisites hinzufügen.

Wenn Sie wieder in die *Seitenübersicht* für die Multisites gehen und entweder eine bereits angelegte Multisite editieren oder eine neue anlegen, finden Sie unter *Felder* das soeben angelegte Feld, das Sie mit Inhalten füllen können.

The screenshot shows the 'Felder' section of the Joomla! Multisites configuration. It contains a single field with the name 'COM\_CCK\_Angelegt\_am' and an empty text input box next to it.

**Bild 20.9:** Das Feld in der Multisite.

#### **Im Zweifelsfall »Text«**

Eigentlich wollte ich ein Datumsfeld anlegen, was ja bei dem Inhalt auch naheliegend wäre. Aber jedes Mal, wenn ich dann die Multisite aufrief, hatte sich das Formular komplett verabschiedet und war nicht mehr benutzbar.

Ich habe nicht alle Feldtypen ausprobiert, aber falls Sie Ähnliches beobachten, liegt es am Feldtyp des neu angelegten Feldes. Ändern Sie es im Zweifelsfall auf *Text*, dann funktioniert es immer.

## 20.5 Mastersite?

Es gibt verschiedene Ansätze, wie Sie mit Ihren Multisites arbeiten können:

Da ist zum einen eine »große« Domain, in der alles liegt und in der der Benutzer mit einem ausgeklügelten Rechtesystem nur das sieht, was er sehen darf und soll. Das kann schnell zu Unübersichtlichkeiten führen, gerade wenn Sie eine große Site haben, in der es womöglich mehr als die Standardberechtigungen gibt.

Oder Sie arbeiten mit einer Mastersite, die Sie zur Verwaltung benutzen, die aber nicht von Benutzern besucht wird.

Wenn Sie mit einer Mastersite arbeiten möchten, dann können Sie diese Site quasi verstecken, indem Sie sie offline schalten und wirklich nur zu Verwaltungszwecken benutzen. Schalten Sie aber die Site in der Hauptkonfiguration offline, sind auch die Subdomains davon betroffen. Aber Sie können sich eine neue Multisite anlegen – Basis würde hier auch vollkommen reichen – und die *URL* auf sich selbst legen. Diese Seite können Sie dann offline schalten.

The screenshot shows the Joomla! administrator interface for configuring a multisite. The 'Url' field is set to 'multimaster.chalambar.de' and is circled. A tooltip points to it with the text 'Die Mastersite verlinkt quasi auf sich selbst. In dieser Subdomain ist Joomla! installiert'. Another tooltip points to the 'Offline' section with the text 'Über Seblod wird diese Domain Offline geschaltet'. The 'Offline' section has a 'Nein' button selected.

**Bild 20.10:** Die Hauptdomain offline schalten.

SEBLOD behandelt die Hauptdomain, Ihre Mastersite, nun als eine ganz normale Multisite und schaltet folglich nur diese Site offline. Zusätzlich erhalten Sie eine neue Berechtigung und können sich einige interne Beiträge oder Module anlegen. Auf dieser Site können Sie schalten und walten und haben gute Chancen, den Überblick zu behalten, weil alles sauber zugeordnet ist.

# 21

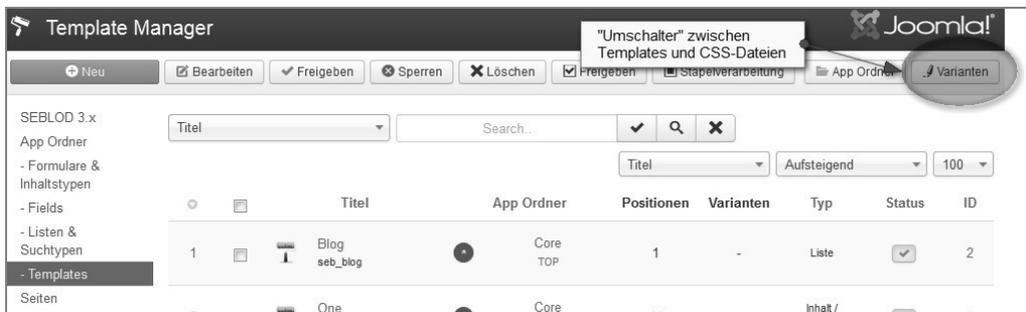
## Templates

Die Templates von SEBLOD unterscheiden sich an einigen Stellen gravierend vom Joomla!-Templating. Dennoch sind ein paar Parallelen vorhanden. Die Templates sind ausschließlich für die Darstellung des Inhaltsbereichs zuständig, also für den Bereich, in dem die Beiträge und andere Komponenten abgebildet werden. Die Templates greifen nur auf Inhaltstypen und/oder (Such-)Listen zu, die mit SEBLOD erstellt wurden. In diesem Kapitel werfen wir zunächst einen Blick auf die vorhandenen Templates. Dazu gehören auch die voreingestellten CSS-Dateien. Danach geht es um die Entwicklung eigener Templates.

### 21.1 Grundsätzlicher Aufbau

Wenn Sie im SEBLOD-Menü *Template* ausgewählt haben, erhalten Sie zunächst einen Überblick über die installierten Templates.

Klicken Sie oben rechts im *Template Manager* auf *Varianten*, wechseln Sie in die CSS-Ansicht. Dort sehen Sie alle installierten CSS-Dateien, die Sie im Inhaltstyp auswählen können.



**Bild 21.1:** Zwischen den installierten Templates und den vorhandenen CSS-Dateien wechseln.

Sie können die CSS-Dateien lediglich downloaden und bei den Templates generell nur die Struktur einsehen.

In der *Template*-Ansicht können Sie einige wenige Einstellungen verändern, die jedoch bei bereits benutzten Templates mit einiger Vorsicht zu genießen sind. Der Versuch, dem Template einen anderen Namen zuzuweisen, schlug bei mir mit der Fehlermeldung, dass der Eintrag bereits vorhanden und ein *Duplicate entry* (doppelter Eintrag) nicht möglich sei, fehl. Somit sind Sie vor einem versehentlichen Überschreiben eines bestehenden Templates geschützt.

Ändern können Sie die Zuordnung *Inhalt/Formular* oder *Liste*. Das Template kann (und muss) gezielt der Ausgabe zugewiesen werden. Nicht nur das generelle Aussehen kann verändert werden, es kann auch gezielt ein Template für die Listenansicht oder für die Suchergebnisse und für den eigentlichen Inhalt angegeben werden. Damit reiht sich SEBLOD auch im Individualdesign in die Joomla!-Tradition der sehr flexiblen Layoutmöglichkeiten ein.

Sie können einem vorhandenen Template jederzeit einen anderen *Titel* geben. Wenn Ihnen beispielsweise das Template namens *Blog* nicht gefällt, geben Sie einen anderen Titel ein. Bestehende Zuweisungen verlieren nicht ihre Gültigkeit, und Sie können das *Blog*-Template in Zukunft an einem für Sie besseren Namen erkennen.

Standardmäßig ist ein Template dem *App Ordner Core* zugewiesen. Sie können das Template grundsätzlich in jeder App auswählen und benutzen. Wenn Sie jedoch ein Template für einen bestimmten Inhaltstyp erstellen, ist es für den besseren Überblick sinnvoll, das Template dieser App entsprechend zuzuweisen. Wenn Sie Ihre App downloaden, wird ein der App zugewiesenes Template automatisch mit heruntergeladen und steht nach dem Installieren in einer anderen Umgebung wieder zur Verfügung.

Mit dem *On/Off-Umschalter* können Sie ein Template auch deaktivieren bzw. aktivieren. Interessant kann das sein, wenn Sie an einem Template noch arbeiten möchten, aber verhindern wollen, dass etwaige Redaktionsmitglieder dieses Template bereits auswählen können.

Unter *Beschreibung* können Sie einen eigenen Kommentar schreiben, vielleicht einige Besonderheiten für andere Administratoren oder wofür Sie das Template gemacht haben.

Unter *Quellen* sehen Sie, wie bereits erwähnt, lediglich die Struktur des Templates, die installierten Dateien und wo sie liegen. Mit den kleinen Pfeilen neben den Verzeichnissen können Sie die Unterordner einsehen, sofern vorhanden. Sie können jedoch kein Verzeichnis anklicken, um gezielt zu sehen, ob und, wenn ja, welche Dateien dort liegen. Auch können Sie an dieser Stelle leider keine Dateien editieren. Allerdings können Sie schnell überblicken, ob es Overrides in diesem Template gibt.

## 21.2 Ein neues Template zuweisen

Ob Sie ein eigenes Template erstellt oder ein fertiges aus dem Internet heruntergeladen<sup>41</sup> haben – Sie müssen es explizit in SEBLOD zuweisen. Der erste Schritt für ein neues Template ist, wie bei Joomla! üblich, der Installer bei *Erweiterungen/Erweiterungen*. Nach der Installation kehren Sie in die *Template*-Ansicht von SEBLOD zurück und klicken oben links auf *Neu*. Sie kommen nun in die Detailansicht des Templates. Da Sie an dieser Stelle noch keines ausgewählt haben, ist die Strukturansicht, die *Quelle*, zunächst leer.

Sie müssen zunächst einen *Titel* vergeben und darunter, unter *Name*, das dazugehörige *Template* auswählen. Ihnen werden hier alle Templates, die Sie installiert haben, angezeigt, also auch die Joomla!-Templates, die mit SEBLOD nichts zu tun haben und dort auch nicht funktionieren. Zur Erinnerung: Joomla! benutzt ein Template, das die gesamte Webseite abbildet – SEBLOD das Template lediglich zur Anzeige des Inhalts, wozu die Ein- und Ausgabe sowie das Backend wie auch das Frontend gehören.

Nachdem Sie das frisch installierte Template ausgewählt haben, entscheiden Sie, ob das Template in allen Inhaltstypen, Listen und Suchformularen benutzbar oder auf bestimmte *App Ordner* eingeschränkt ist. Wie im vorigen Kapitel bereits erwähnt, können Sie ein Template stets in jedem Inhaltstyp verwenden, egal, welchem *App Ordner* Sie es zugewiesen haben. Es ist also zunächst lediglich eine optische Zuweisung. Wenn das Template überall gelten soll, dann ordnen Sie es dem *Schnellen Ordner* oder *Core* zu.

Der *Schnelle Ordner* ist eine Art Sammelbecken für alles, was nicht einwandfrei zugeordnet werden kann bzw. wird. Das ist in Ordnung, solange Sie allein damit arbeiten oder der *Schnelle Ordner* übersichtlich gehalten wird. Um auch noch in einigen Monaten oder gar Jahren einen schnellen Überblick über das System zu gewährleisten, ist es jedoch ratsamer, konsequent übersichtlich und transparent zu arbeiten.

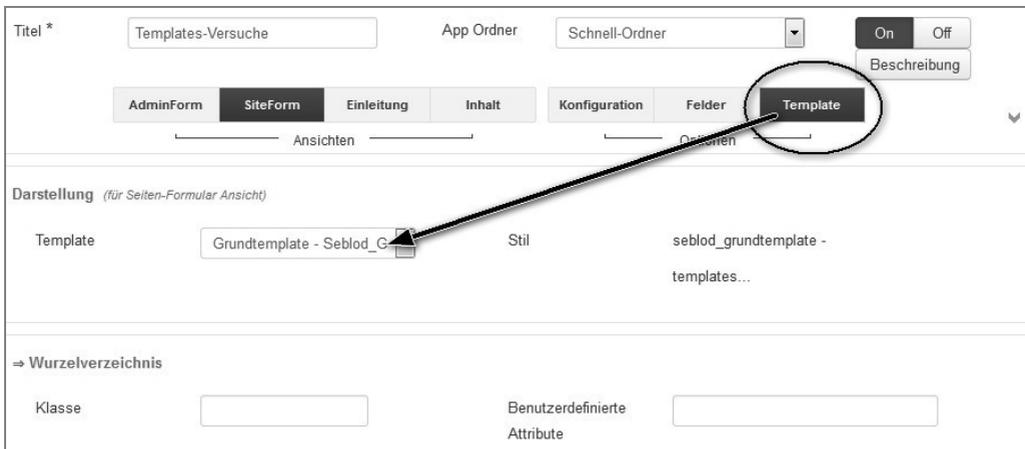
---

<sup>41</sup> Unter <http://www.seblod.com> finden Sie auch fertige Templates zum Downloaden.

Nun müssen Sie noch auswählen, um was für einen *Typ* es sich bei dem *Template* handelt. Soll es für *Inhalte und Formulare* gültig sein oder für *Listenansichten*?

Nach dem ersten Speichern wird Ihnen auch die *Quellen*-Ansicht angezeigt und somit die Struktur des Templates.

Wechseln Sie zu den Inhaltstypen oder Suchlisten – je nachdem, wem das neue Template zugeordnet wurde – und weisen Sie dort bei den Templates das neue Template zu.



**Bild 21.2:** Template im Inhaltstyp zuweisen.

Beachten Sie, dass Sie das Template jedem Formular zuweisen müssen, wenn Sie es in allen Formularen benutzen möchten, also in *AdminForm*, *SiteForm*, *Einleitung* und *Inhalt*.

### 21.3 Ein eigenes Template

Es gibt generell zwei Möglichkeiten für ein eigenes Template. Die eine ist ein komplett eigenes Template, bei dem aber ein wenig Programmieraufwand vonnöten ist, die andere Möglichkeit besteht aus Overrides in einem bestehenden Template wie beispielsweise dem Standard-Template *seb\_one*. Zusätzlich kann eine Mischung aus beidem gemacht werden, also ein eigenes Template mit individuellen Overrides. Damit lassen sich verschiedene Varianten mit nur einem Template darstellen und unterschiedlichen Bereichen gezielt zuweisen.

In fast allen Verzeichnissen finden Sie stets eine leere *index.html*. Wenn Sie Overrides erstellen oder ein eigenes Template bauen, schreiben Sie diese Datei in die Verzeichnisse. Damit verhindern Sie, dass jemand den Inhalt eines Verzeichnisses auslesen kann. Denn wird das Verzeichnis im Browser aufgerufen, sieht man nur eine leere Seite – die leere *index.html*.

Fangen wir mit dem Standard-Template von SEBLOD an: mit dem *seb\_one*.

## 21.4 Overrides im seb\_one

Es gibt einige Möglichkeiten, die Ausgaben seiner Formulare mit dem *seb\_one* zu beeinflussen. Ausgaben für die Positionen oder für einzelne Felder zur manipulieren und eigenes CSS sind natürlich auch möglich.

### 21.4.1 Positionen

Beginnen wir mit der Ausgabe in den Positionen. Ich erstelle zum Üben ein kleines Formular, um Buchbesprechungen zu veröffentlichen. Dazu erzeuge ich drei Felder:

- Buchtitel, Speicherung: Standard - article - title
- Article Category id - ich verwende das vorgegebene Kategoriefeld
- Buchtext als Wysiwyg-Editor - Benutzerdefiniert - article - introtext

Der Buchtext wird in der Position *#MAINBODY* veröffentlicht, die anderen beiden Felder in der Position *#TOPBODY*, davon jeweils eine *AdminForm* und das Gleiche als *Inhalt*.



**Bild 21.3:** Eine einfache Form zur Buchbesprechung.

Ich möchte nun unter dem Titel direkt einen Werbebanner einblenden. Eine Möglichkeit wäre, eine Modulposition einzufügen und dieser Position ein Bannermodul zuzuweisen. In diesem Beispiel lege ich den Werbebanner aber mit einem Override für die Position *# TOPBODY* an, denn dort ist der Buchtitel veröffentlicht, und danach möchte ich den Banner abbilden.

Dafür müssen Sie im Template *seb\_one* das Verzeichnis *positions* mit dem Aliasnamen Ihres Inhaltstyps erweitern und ein weiteres Verzeichnis für den Ausgabebereich erstellen. In unserem Fall sieht der komplette erweiterte Pfad dann so aus:

```
001 /positions/buchbesprechung/content
```

Den *Alias* Ihres Inhaltstyps finden Sie nach dem ersten Speichern in der Übersicht. Sie haben im oberen Bereich des Inhaltstyps, im Ausklappbereich, zwar die Möglichkeit, einen Alias einzutragen, jedoch hat dieser bei mir nicht funktioniert.

Da ich den Werbebanner nur im *Inhalt* ausgeben möchte, reicht hier als Zuweisung *content*, der englische Begriff für »Inhalt«.

Dort erzeuge ich nun eine neue PHP-Datei, die den Namen der Position trägt, die ich überschreiben möchte. In meinem Beispiel soll der Werbebanner unter dem Titel ausgegeben werden, der wiederum an der Position # *TOPBODY* steht. Also erzeuge ich eine Datei namens *topbody.php*. Da ich einen kompletten Override mache, muss ich hier zunächst kundtun, dass meine Felder auch gerendert werden, also der Buchtitel zur Ausgabe gebracht wird. Darunter schreibe ich die Ausgabe für den Werbebanner.

```
001 <?php
002 echo $cck->renderField('buchtitel');
003 echo "<img src=\"images/werbung/\".$cck->getValue('art_catid').
    \".jpg\">";
004 ?>
```

Ich habe der Einfachheit halber meinen Werbebanner mit der ID der Kategorie abgespeichert. Dazu habe ich zuvor zwei Kategorien erstellt

- Belletristik (ID = 10)
- Sachbuch (ID = 11)

und dazu einen Banner *10.jpg* in das Verzeichnis *images/werbung* geladen. Mit

```
001 $cck->getValue('art_catid')
```

frage ich über die Kategorie-ID den Valuewert des Feldes ab und hänge diesen Wert an den Bildpfad. Dieser wird anschließend um den Dateisuffix ergänzt.



**Bild 21.4:** Override, um den Werbebanner an der richtigen Stelle einzufügen.

Wenn Sie mehrere Felder an einer Position haben, müssen Sie die anderen Felder auch rendern. Dadurch können Sie beliebigen Code vor, nach oder zwischen den einzelnen Feldern einfügen.

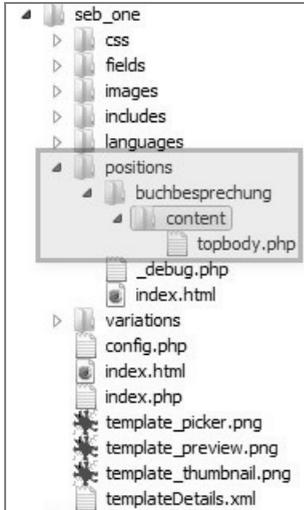
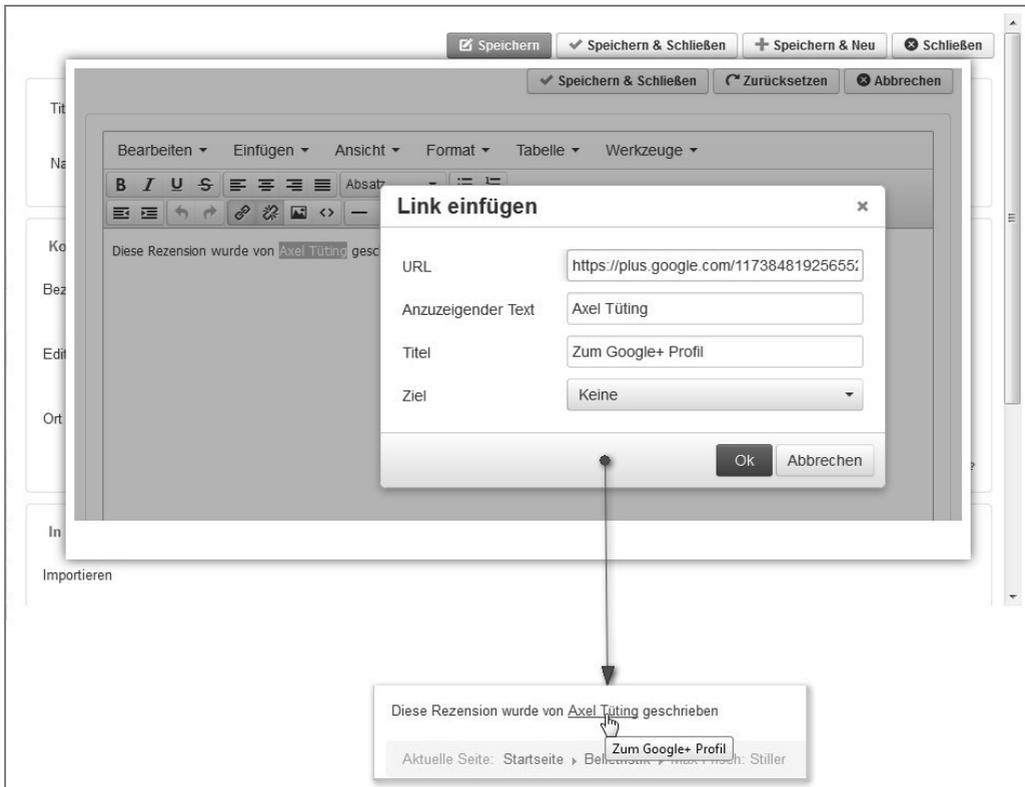


Bild 21.5: Der Override im *positions*-Verzeichnis.

Ich möchte einen Link zu meinen Google+-Profil bei den Buchbesprechungen angeben. Dafür gibt es verschiedene Lösungsansätze. Einer besteht darin, im *Template* des Inhaltstyps (Form: *Inhalt*) einzustellen, dass der Beschreibungstext angezeigt werden soll. Dazu stehen Ihnen verschiedene Methoden zur Verfügung. Im Beschreibungsfeld des Feldes *buchtext* habe ich mir einen Link auf mein Google+-Profil gesetzt, wie in der folgenden Abbildung zu sehen ist.

Nun möchte ich das Ganze ein wenig aufhübschen und umgestalten. Eine Möglichkeit ist, direkt in der CSS-Datei von SEBLOD entsprechende Formatierungen vorzunehmen, eine andere, einen Override für die Position *#MAINBODY* zu machen. Dazu habe ich zunächst eine gleichnamige Datei angelegt – *mainbody.php* –, die ich im gleichen Verzeichnis wie die *topbody.php* gespeichert habe.



**Bild 21.6:** Beschreibungstext mit Link zum Google+-Profil.

Ich möchte meinen Beschreibungstext über dem eigentlichen Text ausgeben, weshalb ich zunächst den Beschreibungstext aufrufe und mit ein wenig CSS formatiere und erst dann das eigentliche Feld rendere:

```
001 <?php
002 echo "<div style='border-bottom:1px solid #000000;margin-
003         bottom:20px;'>".$cck->getDescription('buchtext')."</div>";
004 echo $cck->renderField('buchtext');
```

Eine Linie unter meinem Beschreibungstext und ein wenig Abstand nach unten reichen mir hier. Mit `$cck->getDescription('buchtext')` rufe ich den Beschreibungstext auf.



**Bild 21.7:**  
Die Ausgabe im Frontend.

## 21.4.2 Variationen

Wenn Sie viele Felder haben, könnte der Weg mit den Positionen natürlich etwas mühselig sein, da Sie jedes einzelne Feld rendern müssten. Wollen Sie hingegen einzelne Blöcke, einzelne Positionen insgesamt ausgeben und lediglich davor oder danach den Code ändern, dann haben Sie die Möglichkeit, im *variations*-Verzeichnis ein Override zu machen. Ich habe dort ein neues Verzeichnis namens *werbung* erstellt und eine gleichnamige PHP-Datei gespeichert. Der Code ist abermals recht simpel und wirkt fast identisch mit dem obigen Code bei den Positionen.

```
001 <?php
002 echo $content;
003 echo "<img src=\"images/werbung/\".$cck->getValue('art_catid').
    \".jpg\">";
004 ?>
```

Einziger Unterschied ist die erste Zeile:

```
001 echo $content;
```

Mit dieser Zeile sprechen Sie alle Felder an. Allerdings müssen Sie diese Datei explizit zuweisen. Dazu wechseln Sie in Ihre Inhaltsform und wählen unter *Variation* die PHP-Datei aus. In meinem Beispiel heißt diese *werbung.php*. Wichtig ist, dass das Unterverzeichnis genau so heißt wie die PHP-Datei.

Sie erhalten nach der Auswahl ein kleines Pluszeichen. Wenn Sie es anklicken, werden Sie nichts sehen. Das können wir natürlich ändern und uns ein paar zusätzliche Einstellungen generieren. Dazu legen Sie im gleichen Verzeichnis eine Datei namens *option.xml* an. In dieser erstellen Sie ein `fieldset` mit Parameterangaben.



Bild 21.8: Auswählen des Overrides unter der *Variation* im Inhaltstyp.

Als Beispiel habe ich mir drei Felder gemacht. Zum einen kann ausgewählt werden, wie der Banner ausgerichtet werden soll (links, rechts oder zentriert), und die Bannergröße kann auch verändert werden. Da ich Ausrichtung und Größe optisch trennen möchte, verwende ich zwei `fieldset`s.

```

001 <?xml version="1.0" encoding="utf-8"?>
002 <form>
003   <fieldset name="banner_ausrichtung" label="Ausrichtung">
004     <field name="ausrichtung" type="list" default="center"
005       description="Ausrichtung des Banners" label="Ausrichtung">
006       <option value="left">links</option>
007       <option value="right">rechts</option>
008       <option value="center">mittig</option>
009     </field>
010   </fieldset>
011   <fieldset name="banner_groesse" label="Größe">
012     <field name="breite" type="text" default="468"
013       description="Breite des Banners angeben" label="Breite"
014       size="50"></field>
015     <field name="hoehe" type="text" default="60" description="Höhe
016       des Banners angeben" label="Höhe" size="50"></field>
017   </fieldset>
018 </form>

```

Wenn Sie nun auf das kleine Pluszeichen klicken, erhalten Sie nachfolgendes Formular angezeigt:

Speichern & Schließen Zurücksetzen Abbrechen

⇒ Ausrichtung

Ausrichtung

⇒ Größe

Breite  Höhe

**Bild 21.9:** Individuelle HTML- und CSS-Angaben.

Um die Eingaben im eigenen Code verwenden zu können, brauchen Sie nur

```
$option->get('FIELD_NAME')
```

einzugeben. `FIELD_NAME` müssen Sie natürlich mit dem Namen im *field* ersetzen. Um den Banner auszurichten und mit der gewählten Bildgröße anzuzeigen, erweitern Sie die Datei *werbung.php* mit nachfolgendem Code:

```
001 <?php
002 echo $content;
003 echo '<div style="clear:both; text-align:'.$options->get
    ('ausrichtung').'">';
004 echo '<img src='.$this->baseurl.'/images/werbung/'.$$cck->getValue
    ('art_catid'). '.jpg width='.$options->get('breite').' height=
    '.$options->get('hoehe').' />';
005 echo '</div>';
006 ?>
```

`clear:both` ist notwendig, weil SEBLOD im vorherigen Rendern den *Buchtitel* mit `float:left` ausgibt.

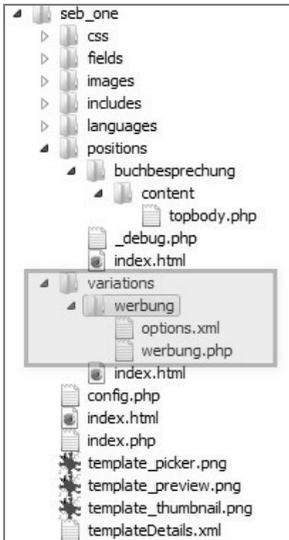


Bild 21.10: Der Override im *variations*-Verzeichnis.

### 21.4.3 Aufgeräumter Quellcode mit eigenen Markups

Nach wie vor haben wir das Problem, dass der Quellcode, den SEBLOD erzeugt, mit vielen DIV-Verschachtelungen daherkommt. Des Rätsels Lösung, um das zu verhindern, liegt in einem Override der Felder. SEBLOD hat uns hierfür bereits eine Datei mitgegeben, die allerdings standardmäßig deaktiviert ist. Wir finden sie im Verzeichnis *fields:\_markup.php*. Damit wir mit ihr arbeiten können, müssen Sie den Unterstrich am Anfang der Datei entfernen. Noch besser ist es, Sie kopieren die Datei, belassen die Originaldatei im Ordner und entfernen bei der Kopie den Unterstrich.

#### Datei mit Unterstrich

Eine Datei, die mit einem Unterstrich beginnt, ist deaktiviert. Sie kann wegen dieses vorangestellten Unterstrichs nicht ausgeführt werden.

Zunächst unterscheidet sich die Ausgabe mit dieser aktivierten Datei nicht von dem, was SEBLOD normalerweise ausgibt. Wir können jedoch diese Datei nach eigenem Gusto anpassen. Die Funktion *ccMarkup\_seb\_one(...)* kommt mit fünf Blöcken daher:

- *Computation* - Bezieht sich auf Berechnungen, die Sie im Inhaltstyp bei den Nummern eingeben können.
- *Conditional* - Bezieht sich auf Abhängigkeiten, die Sie unter der jeweils gleichen Nummer eingeben können.
- *\$desc* - Formatiert Feldbeschreibungen, wenn Sie welche angegeben haben.
- *\$label* - Bezieht sich auf den Titel des Feldes.
- *\$html* - Ist für die Ausgabe des Inhalts des Feldes zuständig.

Im einfachsten Fall belassen Sie *Computation* und *Conditional* für die Berechnungen und Abhängigkeiten und ändern die drei unteren Blöcke. Wenn Sie komplett auf DIV-Umschließungen verzichten möchten, reicht dieser Code:

```
001 $desc = '';
002 if ( $cck->getStyleParam( 'field_description', 0 ) ) {
003     $desc = ( $field->description != '' ) ? $field->description :
004     '';
005 }
006 $label = '';
007 if ( $options->get( 'field_label', $cck->getStyleParam(
008     'field_label', 1 ) ) ) {
009     $label = $cck->getLabel( $field->name, true, (
010     $field->required ? '*' : '' ) );
011     $label = ( $label != '' ) ? $label : '';
012 }
013 $html = $label.$html.$desc;
014 return $html;
```

In `$desc` und `$label` fragen Sie zunächst ab, ob für diese Typen überhaupt Werte existieren. Wenn ja, werden sie ausgegeben, andernfalls wird eine leere Zeichenkette erzeugt. Da Sie Inhalte ausgeben, werden Sie vermutlich immer einen Wert für `$html` haben. In der vorletzten Zeile setzen Sie diese einzelnen Werte einfach zusammen und geben sie mit `return` zurück.

Damit haben Sie einen aufgeräumten Quellcode. Sie können das natürlich auch ein wenig anpassen, damit Sie entweder eine individuelle Ausgabe formatieren oder die einzelnen Feldausgaben mit CSS-Klassen und -IDs belegen können. Um beispielsweise einfache DIVs um die Feldausgaben zu legen, können Sie nachfolgenden Code einfügen:

```
001 $label = '';
002 if ( $options->get( 'field_label', $cck->getStyleParam( 'field_label', 1
003 ) ) ) {
004     $label = $cck->getLabel( $field->name, true, ( $field->required ? '*' :
005     '' ) );
006     $label = ( $label != '' ) ? '<div
007     class="meine_labelklasse">'.$label.'</div>' : '';
008 }
009 $html = '<div class="meine_feldklasse">'.$html.'</div>';
010 $html = $label.$html.$desc;
011 Return $html;
```

Wenn Sie sich den Quellcode Ihrer Ausgabe nach einem Browser-Refresh anschauen, dann haben Sie um Ihre Felder Ihren HTML-Code:

```

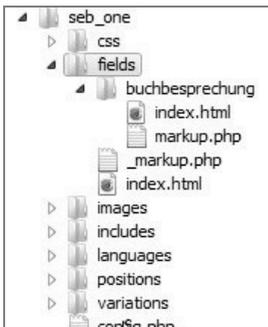
<div class="meine_labelklasse">
  <label>Buchtext</label>
</div>
<div class="meine_feldklasse">
  <p>
    In eget velit eget sem mollis scelerisque nec id ipsum! Donec dignissim
    tellus consectetur, ut pellentesque erat suscipit. Nulla bibendum ac l
    feugiat nulla. Donec eget ipsum mattis, consectetur lectus ut, finibus
    purus nulla, tincidunt imperdiet pulvinar feugiat, fermentum et odio.
  </p>

```

**Bild 21.11:** Eigene CSS-Klassen.

#### 21.4.4 Verschiedene Markup-Overrides

Im vorangegangenen Kapitel haben Sie gelernt, wie Sie auf einfache Art und Weise die Ausgabe »aufräumen« können. Sie können zusätzlich auch Markup-Dateien für bestimmte Inhaltstypen erstellen. Dazu brauchen Sie nur das *fields*-Verzeichnis mit dem Namen des Inhaltstyps zu erweitern und die *markup.php* hineinzukopieren. Wenn ich beispielsweise meinen Inhaltstyp *Buchbesprechung* mit einer eigenen Markup-Datei ausstatten möchte, lege ich ein entsprechendes Unterverzeichnis an:



**Bild 21.12:** Die HTML-Ausgabe für einen bestimmten Inhaltstyp anpassen.

Zusätzlich müssen Sie den Funktionsnamen verändern, indem Sie an das Ende den Namen des Inhaltstyps anfügen, also letztlich den Verzeichnisnamen übernehmen. In meinem Beispiel muss der Funktionsname nun so aussehen:

```

001 function cckMarkup_seb_one_buchbesprechung( $cck, $html, $field,
    $options )
002 { ... }

```

Die genaue Syntax lautet: `cckMarkup_[Templatenname]_[Inhaltstyp]`

Statt *Inhaltstyp* können Sie natürlich auch einen *Suchtyp* angeben. Diese Markup-Datei wird nur ausgeführt, wenn Sie den Inhaltstyp *Buchbesprechung* ausgeben.

Haben Sie mehrere Inhaltstypen, die alle das Standard-Template *seb\_one* benutzen, können Sie so dennoch auf sehr einfache Art und Weise individuelle Ausgaben formatieren, ohne ein eigenes Template erstellen zu müssen. Da *seb\_one* sehr viele Positionen mitbringt, ist dieser Weg vielleicht einer der besten, um SEBLOD zu individualisieren.

### 21.4.5 Zusammenfassung und Fazit

Der Unterschied zwischen den beiden Methoden des *Overrides* ist nicht nur, dass Sie mit den *Variationen* ganze Blöcke und bei den *Positionen* einzelne Felder in den jeweiligen Positionen ansprechen, sondern auch, dass Sie die *Overrides* in den *Variationen* explizit zuweisen müssen, wohingegen die *Overrides* in den *Positionen* immer ausgeführt werden. Mit den Zuweisungen der *Variationen* haben Sie damit zusätzliche Möglichkeiten: Einzelne Positionen in den Formularen können eine eigene Formatierung erhalten.

Das, gekoppelt mit der *markup.php* für das ganze Template oder bestimmte Inhaltstypen, bietet vielfältige Möglichkeiten der eigenen Template-Gestaltung mit den bereits mitgelieferten Standard-Templates.

## 21.5 Einstieg mit einem einfachen Template

Auf den ersten Blick sind die Templates von SEBLOD recht verwirrend und scheinen rein gar nichts mit den Joomla!-Templates gemeinsam zu haben. Bei genauer Betrachtung finden sich aber durchaus Gemeinsamkeiten. Da wir hier aber ein Template haben, das sich zum einen ausschließlich auf den Content-Bereich bezieht und es sich zudem um die Darstellung speziell von SEBLOD-Formularen handelt, sind einige Dateien und teilweise die Ordnerstruktur gravierend anders, als es von den Joomla!-Templates bekannt ist.

Um das Templating von SEBLOD-Templates zu verstehen, erstelle ich ein sehr einfaches Template.

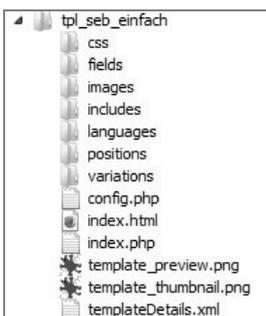


Bild 21.13: Struktur für ein ganz einfaches Template.

Wie die Abbildung zeigt, unterscheidet sich das SEBLOD-Template in der Struktur zunächst nicht so stark von der Joomla!-Template-Struktur. Lediglich ein paar Verzeichnisse sind hinzugekommen. Allerdings brauchen Sie in diese Verzeichnisse nichts hineinzuschreiben bis auf die obligatorische leere *index.html*. Die benötigen Sie zum einen, um auch eigentlich leere Verzeichnisse aus der ZIP-Datei installieren zu können, aber auch als einfachen Verzeichnisschutz, damit die Dateien eines Verzeichnisses nicht einfach aufgelistet werden und somit für jeden sichtbar sind.

Wirklich wichtig sind zunächst nur die drei Dateien im Template-Root-Verzeichnis. Also:

- *index.php*
- *config.php*
- *templateDetails.xml*

Die beiden Grafikdateien dienen als Template-Miniansicht. Die *template\_preview.png* wird angezeigt, wenn Sie im Inhaltstyp auf den Button *Positionen* klicken, und *template\_thumbnail.png* wird Ihnen angezeigt, wenn Sie im Joomla!-Backend in die *Template-Ansicht* wechseln (*Erweiterungen/Templates/Templates*).

### 21.5.1 Festlegen der Positionen

Wie Sie es aus den Joomla!-Templates kennen, werden auch im SEBLOD-Formular die Positionen in der *templateDetails.xml* bestimmt.

```

001 <?xml version="1.0" encoding="utf-8"?>
002 <extension type="template" version="3.3" method="upgrade">
003 <name>SEBLOD_Grundtemplate</name>
004 <creationDate>28.10.2014</creationDate>
005 <author>Axel Tüting</author>
006 <copyright>Axel Tüting</copyright>
007 <authorEmail>tueting@time4mambo.de</authorEmail>
008 <authorUrl>www.time4mambo.de</authorUrl>
009 <version>1.0</version>
010 <description>Ein ganz einfaches SEBLOD-Template - Ein Grundgerüst mit
dem Sie weiterarbeiten können</description>
011
012 <files>
013 <folder>css</folder>
014 <folder>images</folder>
015 <folder>positions</folder>
016 <folder>variations</folder>
017 <folder>fields</folder>
018 <folder>includes</folder>
019 <folder>languages</folder>
020 <filename>config.php</filename>

```

```

021     <filename>index.php</filename>
022     <filename>templateDetails.xml</filename>
023     <filename>index.html</filename>
024 </files>
025
026 <languages folder="languages">
027     <language tag="en-GB">en-GB/en-GB.tpl_seb_one.ini</language>
028     <language tag="de-DE">de-DE/de-DE.tpl_seb_one.ini</language>
029 </languages>
030
031 <positions>
032     <position>links</position>
033     <position>rechts</position>
034 </positions>
035
036 </extension>

```

Anders als im Joomla!-Template legen Sie nicht den `client` fest, sondern die `method=upgrade`. Das SEBLOD-Template greift sowohl im Front- als auch im Backend. Im oberen Abschnitt geben Sie Informationen zum Ersteller des Templates, der vermutlich Sie sind, ein. Sie finden diese Angaben im Backend bei den Templates wieder.

Im Abschnitt `files` binden Sie die Verzeichnisse und deren Dateien und einzelne Dateien ein. Sie können auch Sprachdateien definieren, die Sie entsprechend einbinden. In diesem ersten Beispiel habe ich zwei Positionen definiert. Eine für links und eine für rechts. Wenn Sie dieses Template auswählen, erhalten Sie im Inhaltstyp diese beiden Positionen.

### 21.5.2 Damit überhaupt etwas ausgegeben wird: config.php

Die `config.php` können Sie aus dem bestehenden `tpl_seb_one` kopieren. Das Template finden Sie im Übrigen auch im Template-Verzeichnis von Joomla!.

```

001 <?php
002 defined( '_JEXEC' ) or die;
003 global $user;
004 $app      =      JFactory::getApplication();
005 $path_lib=      JPATH_SITE.'/libraries/cck/rendering/rendering.php';
006 $user     =      JCck::getUser();
007 if ( ! file_exists( $path_lib ) ) {
008     print( '/libraries/cck/rendering/rendering.php file is missing.' );
009     die;
010 }
011 require_once $path_lib;
012 ?>

```

Zwecks besseren Lesens des Codes spare ich am Anfang die @package-Hinweise etc.

Neben dem Setzen einiger globaler Variablen sind hier die Zuweisung und die Abfrage auf das Vorhandensein der *rendering.php* entscheidend.

Die *rendering.php* – *rendering* bedeutet frei übersetzt *Ausführung* – sorgt dafür, dass das gesamte Back- und Frontend überhaupt angezeigt und ausgegeben werden kann. Dazu gehört so ziemlich alles, was SEBLOD ausmacht. Wie Sie gleich sehen werden, nutzen wir die Methoden der *rendering.php*, weshalb es auch sehr wichtig ist, dass diese Datei eingebunden ist.

Sie sollten allerdings Änderungen nicht in dieser SEBLOD-Core-Datei vornehmen. Das können Sie an anderer Stelle besser machen. Außerdem wird beim nächsten Update die *rendering.php* vielleicht wieder überschrieben. Die *rendering.php* lädt denn auch Ihre Einstellungen und Ihre App mit allem, was dazugehört.

Sie könnten auch auf die *config.php* verzichten und den Code direkt an den Anfang der *index.php* setzen. Da aber die Möglichkeit besteht, seine eigene Rendering-Klasse zu erstellen, ist es zwecks besseren Überblicks besser, wenn Sie solcherlei in der *config.php* belassen. Sie folgen damit der SEBLOD-Struktur. Und da es ja auch sein kann, dass Sie Ihre Applikationen weitergeben, kann ein Nachfolger Ihre Dateien und Änderungen sauber nachvollziehen.

Ein Beispiel für eine eigene Rendering-Klasse können Sie dem kostenlosen *SD Simple Simon Template*<sup>42</sup> entnehmen. Sie benötigen hierfür allerdings vertiefte Programmierkenntnisse, was den Rahmen dieses Buches sprengen würde.

### 21.5.3 Die eigentliche Ausgabe: index.php

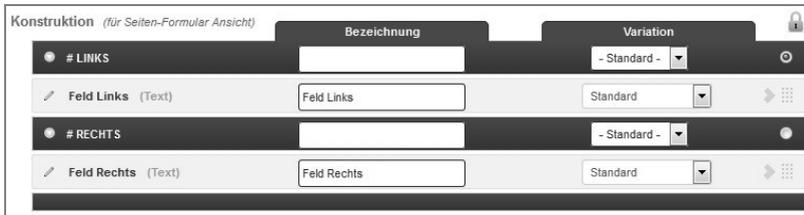
Die *index.php* ist die Datei, die dafür sorgt, dass unsere Felder ausgegeben werden und dass wir mit ihnen arbeiten können. Nachdem Sie die *config.php* eingebunden haben und somit die *rendering.php* ansprechen können, benutzen wir deren Methoden, um unsere Felder zur Anzeige zu bringen.

```
001 <?php
002 defined( '_JEXEC' ) or die;
003 require_once dirname(__FILE__).'/config.php';
004 $cck = CCK_Rendering::getInstance( $this->template );
005 if ( $cck->initialize() === false ) { return; }
006 echo $cck->renderPosition( 'links' );
007 echo $cck->renderPosition( 'rechts' );
008 ?>
```

In *\$cck* steht quasi unser Template als Objekt. Mit *\$cck->renderPosition* sprechen Sie unser Template an und rufen die Methode *renderPosition* in der *rendering.php*

<sup>42</sup> <http://www.seblod.com/products/2315>

auf. Diese Methode sorgt dafür, dass unsere Felder in unseren Positionen auch angezeigt werden.



**Bild 21.14:** Zwei Positionen und jeweils ein einfaches Textfeld im Inhaltstyp.

Alles, was Sie nun noch tun müssen, ist, aus den Verzeichnissen und Dateien eine ZIP-Datei zu erstellen und sie Ihren Formularen zuzuweisen.

## 21.5.4 Rendering-Variationen

Statt alle Positionen Ihres Templates einzeln anzusprechen, so wie ich es in dem überschaubaren Beispiel oben gemacht habe, können Sie auch eine Schleife über Ihre Positionen legen, was besonders dann sinnvoll ist, wenn Sie viele Positionen in Ihrem Template haben.

```
001 foreach($this->positions as $position => $fields){
002     if($cck->countFields($position)){
003         echo $cck->renderPosition($position);
004     };
005 };
```

Sie können hier auch sehr gut HTML-Code mit übergeben. Um eine DIV-Klasse, um die Felder zuzuweisen, können Sie die Schleife folgendermaßen erweitern:

```
001 foreach($this->positions as $position => $fields){
002     if($cck->countFields($position)){
003         echo "<div class=".trim(strtolower($position)).">";
004         echo $cck->renderPosition($position);
005         echo "</div>";
006     };
007 };
```

Sie erhalten nun ein `div` mit einer Klasse gleichen Namens wie die Position:

```
001 <div class=links>
002 <div id="cck1r_links" class="cck-pos-links seb_css3 vertical cck1r-
003     deepest">
003 <div id="cck1r_feld_links" class="cck_forms cck_admin cck_text
```

```

cck_feld_links"><div id="cck1r_label_feld_links" class="cck_label
cck_label_text">
004 <label for="feld_links">Feld Links</label>
005 </div>
006 <div id="cck1r_form_feld_links" class="cck_form cck_form_text">
007 <input type="text" id="feld_links" name="feld_links" value=""
class="inputbox text" size="32" maxlength="255" />
008 </div>
009 </div>

```

### 21.5.5 Der Ordnung halber: die eigene CSS-Datei

In der Ordnerstruktur des SEBLOD-Templates gibt es auch ein Verzeichnis namens `css`. Es liegt nahe, dort eine CSS-Datei unterzubringen.

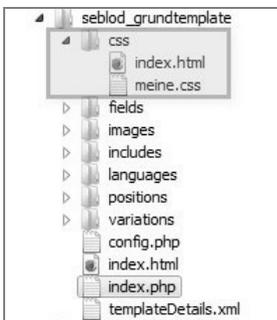


Bild 21.15: Eigene CSS-Dateien einbinden.

Es reicht allerdings nicht, dort einfach eine CSS-Datei hineinzuschreiben – sie muss auch eingebunden werden. Das machen Sie in Ihrer `index.php`. Als Beispiel habe ich eine kleine CSS-Datei mit dem Namen `meine.css` erstellt. Ich möchte den Beitragstext links um 150 Pixel einrücken und ihn in roter Schrift ausgeben. Der CSS-Code dafür:

```

001 #cck5_value_buchtext {
002     padding:0 0 0 150px;
003     color:#E8143C;
004     width:500px;
005 }

```

In die `index.php` meines Templates muss ich diese Datei noch einbinden. Dazu schreibe ich folgende Zeilen an den Anfang meiner Datei:

```

001 $doc = JFactory::getDocument();
002 $doc->addStyleSheet( JURI::root( true ).'/templates/'. $cck->template.
'/css/'. 'meine.css' );

```

Mit `getDocument` ist die Datei gemeint, die im Browser ausgegeben wird. Diese wird quasi in `$doc` gespeichert. Mit `addStyleSheet` weisen Sie dieser Ausgabe Ihre CSS-Datei zu.

Wenn Sie zur Laufzeit den Quelltext der Ausgabe anschauen, dann finden Sie im Header diese Zeile:

```
001 <link rel="stylesheet"href="/j336/templates/seb_one/css/meine.css"
    type="text/css" />
```

Übrigens gründet sich letztlich auf die Möglichkeit der eigenen CSS-Datei auch das Verzeichnis `images`. Wie von den Joomla!-Templates bekannt, liegen hier meistens Grafiken, die zu Layoutzwecken genutzt werden, die einfach mit relativen Pfaden von der CSS-Datei aufrufbar sind.

Wer nicht so bewandert mit den relativen Pfaden ist:

```
001 #beispiel {
002     Background:url(..images/meinBild.jpg);
003 }
```

Da die CSS-Datei im Verzeichnis `css` liegt, müssen Sie zuerst mit `../` eine Verzeichnisebene zurück- und dann direkt in das `images`-Verzeichnis hineingehen. Um eine eigene HTML-Ausgabe des Templates zu erstellen, passen Sie die `markup.php` im Verzeichnis `fields` an.

## 21.6 Globale Methoden und Eigenschaften

Es gibt einige fertige Methoden und Eigenschaften, die Sie in Ihren Individualanpassungen übernehmen können. Hier eine kleine Liste mit diesen Befehlen. `FELDNAME` müssen Sie jeweils mit dem *Alias* Ihres Feldes ersetzen.

<code>\$cck-&gt;getLabel('FELDNAME')</code>	Gibt Ihnen den Inhalt des Labels für das Feld aus.
<code>\$cck-&gt;getValue('FELDNAME')</code>	Gibt den Inhalt des Feldes aus.
<code>\$cck-&gt;getHTML('FELDNAME')</code>	Gibt Ihnen den kompletten Code aus, der erzeugt wird, wenn Sie bei den Nummern <i>Link und Typografie</i> auswählen und einen <i>Link</i> zuweisen.  Beispiel einer Ausgabe mit <i>Typografie Fett</i> :  <strong>Max Frisch: Stiller</strong>

<pre>\$cck-&gt; getTypo('FELDDNAME')</pre>	<p>Gibt Ihnen den kompletten Code aus, der erzeugt wird, wenn Sie bei den Nummern <i>Link und Typografie</i> auswählen und eine <i>Typografie</i> zuweisen.</p> <p>Beispiel einer Ausgabe:</p> <pre>&lt;a href=" /index.php/belletristik?id=3:max- fritsch-stiller"&gt;Max Frisch: Stiller&lt;/a&gt;</pre>
<pre>\$cck-&gt; getText('FELDDNAME')</pre>	<p>Wenn Sie ein Feld haben, das Ihnen als Value eine ID zurückgibt, können Sie mit diesem Methodenaufwurf stattdessen den zugehörigen Text ermitteln. In einer Checkbox wird Ihnen der Text der aktivierten Checkboxen ausgegeben, genau so bei den Radiobuttons.</p> <p>Beispiel:</p> <pre>\$cck-&gt;getValue('art_catid'); //Liefert das Ergebnis: 10  \$cck-&gt;getText('art_catid'); //Ergibt als Rückgabe den Kategorienamen »Belletristik«</pre>
<pre>\$cck-&gt; renderField('FELDDNAME')</pre>	<p>Sowohl das Label als auch der Inhalt des Feldes (Value) werden ausgegeben. Das Label kann mit einem <i>clear</i> im Inhaltstyp jedoch explizit gesperrt werden. Genau so wie ein Deaktivieren der Labelausgabe in den Template-Optionen des Inhaltstyps.</p>
<pre>\$cck-&gt;getDescription ('FELDDNAME')</pre>	<p>Gibt den Beschreibungstext eines Feldes aus.</p>
<pre>\$cck-&gt;id</pre>	<p>Liefert Ihnen die ID der aktuell verwendeten Formularausgabe zurück.</p> <p>Beispiel:</p> <pre>cck5</pre> <p>Diese ID wird als Präfix vor verschiedenen HTML-IDs und -Klassen benutzt:</p> <pre>&lt;div id="cck5_topbody" class="cck5-deepest"&gt;</pre>

# 22

## Plug-ins

In der Plug-in-Liste, die Sie über das Joomla!-Pull-down-Menü erreichen (*Erweiterungen/Plug-ins*), finden Sie sehr viele Plug-ins mit CCK am Anfang des Namens. Alle diese Plug-ins gehören zu SEBLOD.

### 22.1 Plug-in-Übersicht

Hier finden Sie die verschiedenen Bereiche, die mit Plug-ins in SEBLOD abgedeckt werden. Dazu gehören die verschiedenen *Felder*, die sich wiederum in *Typografie*, *Bestätigung*, *Validation* und die verschiedenen *Feldtypen* aufteilen. Dazu gehören auch Plug-ins, die sich um den Bereich der *Speicherung* kümmern.

Wenn Sie eines der Plug-ins editieren, haben Sie in jedem Plug-in den Tab *Erweiterte Optionen*.

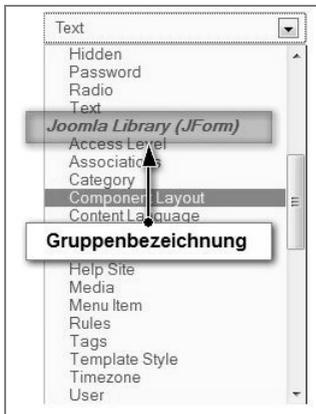


The screenshot shows a Joomla! configuration interface with three tabs: 'Details', 'Basisoptionen', and 'Erweiterte Optionen'. The 'Erweiterte Optionen' tab is active. It contains two rows of options:

Option	Value
Gruppe	PLG_CCK_FIELD_GROUP_JFORM
Export	<a href="#">Download</a>

**Bild 22.1:** *Erweiterte Optionen* in den Plug-ins.

In *Gruppe* können Sie einen Sprachschlüssel eintragen, der die Gruppe bezeichnet, unter der Sie das jeweilige Plug-in in den SEBLOD-Listen wiederfinden. Beispiel: Wenn Sie Ihrem Formular einen neuen Feldtyp hinzufügen, dann sind in der Liste alle Feldtypen verschiedenen Gruppen zugeordnet:



**Bild 22.2:** Gruppenbezeichnung in den Listen.

Diese Gruppenzuordnung können Sie in den Plug-ins ändern, indem Sie einen anderen Sprachschlüssel eintragen. Sprachschlüssel deswegen, weil ja die Gruppenzuordnung in allen Sprachen passen muss. Dummerweise gibt es aber keine zentrale Sprachdatei, in der diese ganzen Schlüssel im Überblick verfasst sind. Alle Gruppen fangen mit `PLG_CCK_FIELD_GROUP_` an und haben am Ende die eigentliche Gruppenbezeichnung. Eine schnelle Suche im Windows Explorer brachte 46 Treffer:

PLG_CCK_FIELD_GROUP_			
	de-DE.plg_cck_field_upload_file.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 868 Bytes
	de-DE.plg_cck_field_upload_image.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 2,50 KB
	de-DE.plg_cck_field_video_youtube.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 689 Bytes
	de-DE.plg_cck_field_wysiwyg_editor.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 1,21 KB
	de-DE.plg_cck_field_42.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 630 Bytes
	de-DE.plg_cck_field_author.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 575 Bytes
	de-DE.plg_cck_field_button_submit.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 697 Bytes
	de-DE.plg_cck_field_calendar.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 1,39 KB
	de-DE.plg_cck_field_captcha_math.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 739 Bytes
	de-DE.plg_cck_field_checkbox.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 630 Bytes
	de-DE.plg_cck_field_colorpicker.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 619 Bytes
	de-DE.plg_cck_field_div_clear.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 610 Bytes
	de-DE.plg_cck_field_email.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 1,46 KB
	de-DE.plg_cck_field_field_x.ini C:\wamp\www\j31\administrator\language\de-DE	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13 Größe: 588 Bytes
	de-DE.plg_cck_field_freertext.ini	Typ: INI-Datei	Änderungsdatum: 28.06.2013 23:13

Bild 22.3: Gruppenbezeichnungen sind Fleißarbeit.

Sie können allerdings eigene Gruppennamen auch einfach in das Feld schreiben und finden anschließend diese Zuordnung in Ihren Listen wieder. Allerdings sollten Sie bestehende Plug-ins nur unter Vorbehalt ändern. Zum einen könnten Sie Schwierigkeiten haben, in diesem Buch alles wiederzufinden, da ich natürlich strikt dem Standard folge, zum anderen sind Supportanfragen eventuell erschwert, weil beim Supporten jeder vom Standard ausgeht.

Wirklich spannend wird das, wenn Sie eigene Plug-ins entwickeln und sie nach Ihrem Gusto zuordnen möchten. Wie auch immer, Sie können geänderte Plug-ins mittels des Download-Buttons exportieren und sie somit sowohl anderen leicht zur Verfügung stellen als auch in eigenen Installation wiederverwenden. Sie müssen so nicht jede Einstellung eines Plug-ins, die Sie in mehreren Installationen verwenden möchten, jedes Mal neu machen.

## 22.2 Wo sind die Plug-ins geblieben?

Die Plug-ins, die Sie zusätzlich installieren können, haben jeweils unterschiedliche Aufgaben, und Sie sollten vor der Installation darauf achten, wofür das Plug-in ist. So finden Sie beispielsweise alle Plug-ins, die etwas mit Typografie zu tun haben, unter der *Nummer 2* in den jeweiligen Ausgabefeldern; Plug-ins, die mit der Speicherung zu tun haben, finden Sie im Speicherbereich eines Feldes ...

## 22.3 Verbindung zwischen Benutzer und Beitrag

Es gibt zwei Plug-ins, die eine Option namens *Brücke aktivieren* haben und standardmäßig deaktiviert sind. Sie finden die beiden Plug-ins, wenn Sie in der Plug-in-Übersicht von Joomla! den *Typ* auf *cck\_storage\_location* filtern. Die beiden Plug-ins sind:

- CCK Speicherort – Joomla!-Benutzer
- CCK Speicherort – Joomla!-Benutzergruppe

Wie der *Typ* bereits vermuten lässt, handelt es sich um das Speichern von Benutzergruppen oder Benutzern. Sie können in diesen Plug-ins noch ein paar Dinge mehr einstellen, die Sie bei den anderen Plug-ins nicht haben.

Die *Brücke* ist ein Verbindungsglied zwischen Joomla! und SEBLOD und hat die Aufgabe, Beiträge und Benutzer miteinander zu verbinden.

Wenn Sie *Brücke aktivieren* auf *Ja* stellen, wird in Zukunft bei jedem neuen Benutzer, den Sie anlegen oder der sich in Joomla! registriert, ein Dummy-Beitrag erstellt. In diesem Beitrag mit dem Titel »Joomla User – [ID]« wird auch der Inhalt von SEBLOD geschrieben. Im Beitragsfeld steht dann:

```
001 ::cck::5::/cck::
```

Die zugeordnete Kategorie ist *User*, und der Beitrag ist veröffentlicht. SEBLOD benötigt diese Beiträge, um die angelegten Benutzer einwandfrei ihren Beiträgen zuzuordnen, damit bei Anklicken des Autorennamens das Benutzerprofil angezeigt wird. Wenn Sie viele Benutzer haben, haben Sie auch eine sehr große Anzahl eigentlich unnützer Beiträge. Da diese stets in der Beitragsübersicht zu finden sind, entsteht sehr viel unnötiger Ballast, und insbesondere wenn Sie Kundenseiten erstellen, kann das eine Zumutung für Ihren Kunden darstellen. Im Anhang finden Sie einen Link zu einem Mitgliedertutorial. Dort finden Sie eine Programmierlösung, die dieses Problem anders handhabt und die ganzen Beiträge verhindert.



## CSS-Klassen

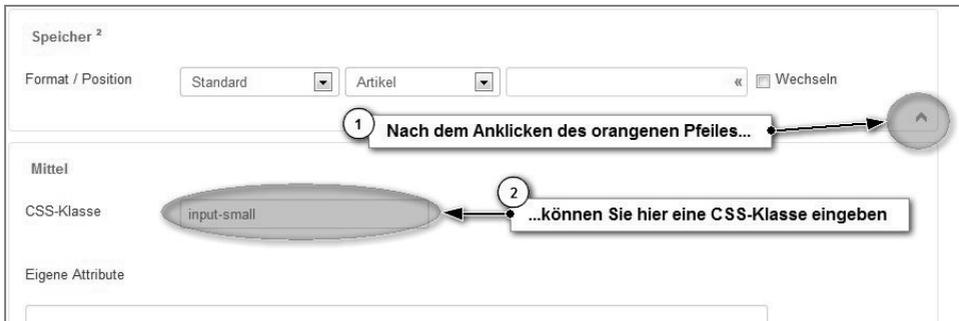
Es gibt die Möglichkeit, einige CSS-Klassen aus dem Bootstrap<sup>43</sup> in SEBLOD zu übernehmen. Diese Klassen können an verschiedenen Stellen in SEBLOD eingebunden werden und bewirken teilweise sehr unterschiedliche Formatierungen. Dieses Kapitel soll ein wenig Überblick geben.

### A.1 Eingabe direkt im Feld

Wenn Sie direkt im Feld arbeiten, können Sie am unteren Rand im Speicherbereich die orangefarbenen Pfeile anklicken. Es klappt dann der Bereich *Mittel* auf, in dem Sie CSS-Klassen direkt eingeben können.

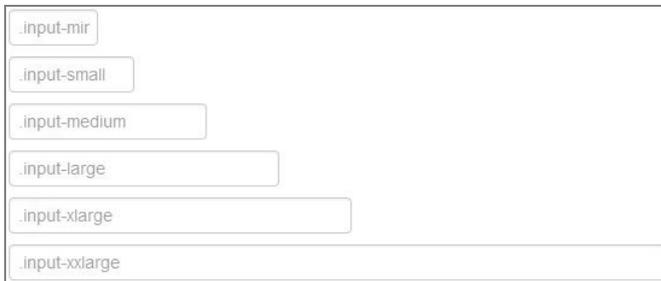
---

<sup>43</sup> Bootstrap ist ein Framework und enthält eine Sammlung von Hilfsmitteln für die Gestaltung von Websites. Dazu gehören auch fertige CSS-Formatierungen für Typografie, Formulare, Buttons und andere.



**Bild A.1:** CSS-Klassen direkt beim Feld angeben.

Legen Sie Textfelder an, können Sie die Länge der Felder in der Ansicht mit CSS-Klassen festlegen:



**Bild A.2:** CSS-Klassen für die Länge der Eingabefelder.

Die Klassen von oben nach unten:

- `input-mir`
- `input-small`
- `input-medium`
- `input-large`
- `input-xlarge`
- `input-xxlarge`

Wenn Sie Buttons benutzen möchten, bieten sich einige Klassen zur Gestaltung an, darunter auch eine, die aus dem Button einen optischen Link macht:

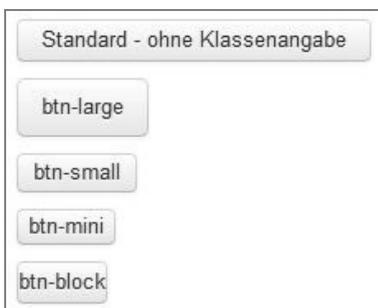


**Bild A.3:** Buttongestaltung mit CSS-Klassen aus dem Bootstrap.

Ganz oben ist ein Button ohne Klassenangabe, und darunter:

- btn-primary
- btn-info
- btn-success
- btn-warning
- btn-danger
- btn-inverse
- btn-link

Unterschiedliche Größen sind ebenfalls möglich:



**Bild A.4:** Buttons unterschiedlich groß.

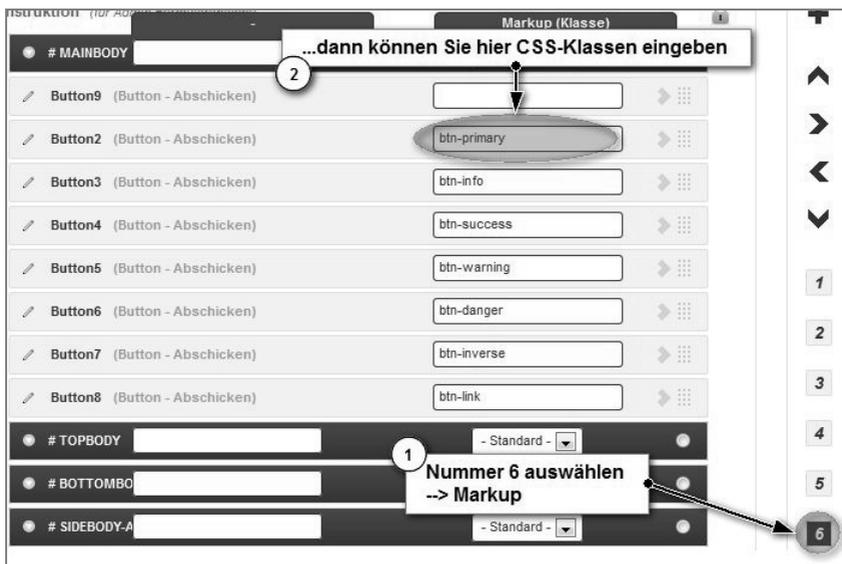
Die Klassennamen unter dem Standard von oben nach unten:

- btn-large
- btn-small
- btn-mini
- btn-block

**btn-block** macht den Button exakt so lang, wie die Beschriftung breit ist. Alle anderen Klassen fügen rechts und links ein paar Pixel bis zum Buttonrand hinzu.

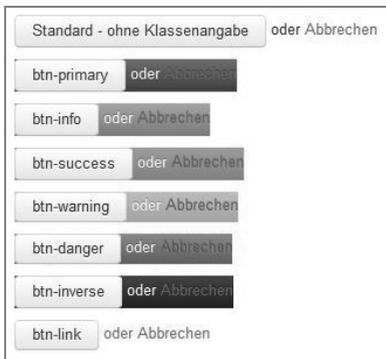
## A.2 Eingabe im Markup

Sie können Klassen auch über die *Nummer 6 - Markup* einfügen. Der Effekt ist dann etwas anders, als wenn Sie die Klassen direkt in die Felder eingeben.



**Bild A.5:** Klassen direkt in der Übersicht eingeben.

Für die Buttons gelten die gleichen Klassen wie oben, sie haben jedoch Auswirkung auf den Hintergrund und lassen den Button selber im Defaultmodus:



**Bild A.6:** Nicht der Button wird formatiert, sondern der Hintergrund.

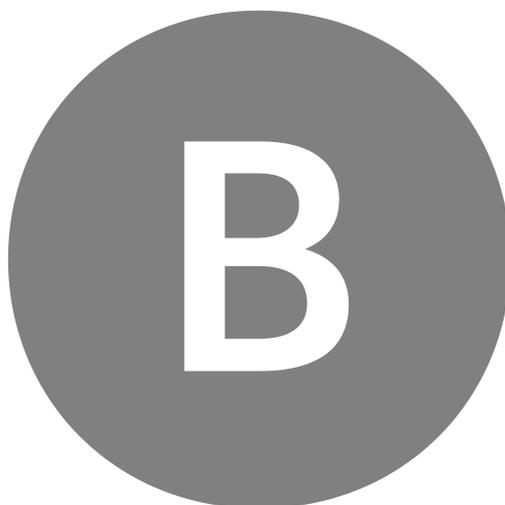
Die CSS-Klassen für die Größenangaben können auch benutzt werden, lassen den Button aber stets auf Defaultgröße und verändern den Schriftzug *oder Abbrechen*. Die CSS-Klasse `btn-block` hat allerdings die gleichen Formatierungen wie der Defaultbutton.



**Bild A.7:** Der Schriftzug wird größer und kleiner.

Sie können die einzelnen Klassen natürlich nach Belieben kombinieren und so alle Effekte nutzen.





## Troubleshooting

### B.1 WYSIWYG-Editor im Frontend

Wenn Sie einen Formulartyp anlegen, der für Gäste einen WYSIWYG-Editor vorsieht, dieser aber nicht angezeigt wird, dann finden Sie in der Plug-in-Übersicht von Joomla! das Plug-in *CCK Feld - Wysiwyg Editor*. Hier müssen Sie den Editor für Gäste explizit erlauben.



Bild B.1: WYSIWYG-Editor für das Frontend erlauben.

## B.2 Merkwürdige Zeichen im Frontend

Wenn Sie im Frontend irgendetwas mit zwei und mehr Doppelpunkten sehen, dann haben Sie einen Fehler in der Speicherung Ihrer Daten gemacht.

Generell können zwei Zustände eingetreten sein:

- ① Sie haben einen benutzerdefinierten Inhalt einer Datenbankspalte zugewiesen, die dafür nicht geeignet ist, weil beispielsweise Joomla! einen »Vollzugriff« auf diese Spalte machen muss und demzufolge den Inhalt so ausgibt, wie es in der Spalte eingetragen ist
- ② Sie greifen mit einem anderen Feld auf den Inhalt einer Datenbankspalte zu, die zuvor als *Benutzerdefiniert* definiert wurde.

Weitere Details entnehmen Sie bitte dem Kapitel über die *Speicherung* (Kapitel 4) und dort insbesondere Kapitel 4.2 und 4.3.

## B.3 SQL-Fehlerbeschreibungen im Frontend

Das kann sehr viele Ursachen haben. Meistens liegt ein Speicherfehler vor. Schauen Sie dafür auch das vorige Kapitel an.

Weitere Möglichkeiten können falsch zugeordnete/ausgefüllte Felder in den einzelnen Feldtypen sein. Da werden Sie vermutlich selber mit detektivischem Einsatz dem Ganzen auf den Grund gehen müssen.

## B.4 Fehlermeldung: Übergeordnete Kategorie ...

Vermutlich haben Sie statt des Feldes *Article Category Id* das Feld *Category Parent Id* eingebunden. *Parent* kümmert sich um die übergeordneten Kategorien und bietet somit die Möglichkeit, Kategorieblöcke einzubinden. Wollen Sie hingegen eine Kategorie zuweisen oder auswählen, müssen Sie den Feldtyp für die *Article Category* einbinden.

## B.5 Das Bild im Frontend ist nicht anklickbar

Sie haben einen Image-Feldtyp eingefügt und können nicht auf das hochgeladene Bild im Frontend klicken? Dann gehen Sie zu Ihrem Formulartyp und wählen die *Nummer 3* im *Intro* oder *Inhalt* aus. Sie müssen dem Feldtyp *Images* oder *Feld X*, wenn sich dieser Feldtyp auf Bilder bezieht, explizit die *Typografie Bild* zuweisen. Wenn Sie auf das dann erscheinende kleine Pluszeichen klicken, können Sie auswählen, welches Bild in welcher Größe konkret angezeigt werden soll.

## B.6 Template existiert nicht

Die Originalfehlermeldung im Frontend lautet:

*Template Style does not exist. Open the Content Type & save it again. (Intro + Content views)*

Manchmal taucht der Fehler auf, weil etwas an den Feldern geändert wurde, diese vielleicht nicht mehr existent sind oder aber in den Eingabefeldern (also Admin- oder Seitenformular) nicht mehr auftauchen und ergo nicht im Frontend angezeigt werden können. Allerdings hatte ich diesen Fehler auch schon ohne ersichtlichen Grund. Er lässt sich einfach beseitigen, indem der Inhaltstyp editiert wird und sowohl das Intro als auch der Inhalt noch einmal explizit gespeichert werden.

## B.7 Anzahl der sichtbaren Felder beschränkt?

In der deutschsprachigen SEBLOD-Community<sup>44</sup> wurde berichtet, dass nicht mehr als 29 Felder in einem Inhaltstyp angezeigt werden können. Bei anderen werden bis zu 60 und mehr Felder problemlos angezeigt.

Eine mögliche Erklärung kann sein, dass die *php.ini* nicht ausreichend eingestellt ist. Dort gibt es einige Angaben zu Maximalwerten:

```
001 max_file_uploads=50
```

```
002 max_input_vars=1000
```

Offenbar funktioniert es problemlos, wenn Sie `max_input_vars` auf **3000** setzen:

```
001 max_input_vars=3000
```

Das Problem ist allerdings, dass Sie an diese Datei normalerweise nicht herankommen, wenn Sie keinen eigenen Server betreiben. Dann können Sie nur Ihren Hoster bitten, ob er diesen Wert für Sie entsprechend ändert. Eventuell hilft aber auch ein Blick in die Domainverwaltung. Manche Hosters bieten Möglichkeiten, dass Sie hier individuelle Einstellungen vornehmen können.

## B.8 Nach dem Joomla!- und SEBLOD-Update

Es ist mir ein paarmal passiert, dass nach einem Joomla!- und dem darauffolgenden SEBLOD-Update irgendetwas in SEBLOD nicht mehr sauber funktionierte. Das muss nicht so sein, und es gibt auch keine Erklärung, warum das zumindest bei mir einige Male so war. Sollten Sie aber ein ähnliches Phänomen feststellen: Bei mir war stets die Lösung, dass ich meine Apps per Download sicherte, SEBLOD deinstallierte (über

---

<sup>44</sup> <http://seblod.time4mambo.de>

*Erweiterungen/Erweiterungen/Verwalten*) und danach komplett neu installierte. Danach dann die Apps installieren, und alles lief wieder einwandfrei.

## B.9 Mehrere Startseitenmenüs bei Multisites

Wenn Sie unter *Homepage* bei den Multisites einen Menüverweis auswählen, haben Sie diesen Verweis manchmal zweimal in Ihrem Menü angezeigt. Warum das so ist und wie man Abhilfe schafft, ist in Kapitel 20.2 »Die Multi-Berechtigungen« ausführlich erklärt.

## B.10 Inhaltstyp wird nicht angezeigt

Sie haben ein neues Formular und einen Inhaltstyp angelegt, aber dieser wird nicht als Auswahl angezeigt, wenn Sie einen neuen Beitrag schreiben möchten? Dann kann es sein, dass Sie den vermeintlichen Beitrag bei den Inhaltstypen nicht als *Artikel* abgespeichert haben.



**Bild B.2:** Nur die Objekte »Artikel« werden in den Beiträgen zur Auswahl angezeigt.

Wenn Sie hier nicht explizit *Artikel* auswählen, wird Ihr Formular auch nicht bei den Beiträgen zur Auswahl angezeigt.

Editieren Sie also Ihren Inhaltstyp, klicken Sie oben rechts auf den orangefarbenen Pfeil und überprüfen Sie das *Objekt*.

Ein zweiter Grund kann sein, dass Sie vielleicht während der Erstellung des Inhaltstyps die Beitragsübersicht offen hatten. Falls ja, laden Sie einfach die Übersicht neu. Ich gebe zu, dass mir das ein paarmal passiert ist ...



## Weiterführendes

### C.1 Deutschsprachige SEBLOD-Community

Während der Entstehung dieses Buches ist die deutschsprachige SEBLOD-Community entstanden. Ursprünglich war es eine Idee auf dem Joomladay 2013 in Nürnberg, geriet dann zunächst ein wenig in Vergessenheit und wurde durch einen Post von Paul im Joomlaportal<sup>45</sup> wieder aufgegriffen. Zu Paul und mir gesellte sich Niels, der die holländische Übersetzung von SEBLOD machte, und im November 2013 gründeten wir die deutschsprachige SEBLOD-Community.

Auf diesen Seiten gibt es nicht nur ein rege besuchtes Supportforum zu SEBLOD, sondern auch eine deutschsprachige Ressourcenliste und Beispielseiten, in denen SEBLOD eingesetzt wird. Diese Seiten werden stetig ergänzt.

Wir freuen uns über jeden Besuch. Sei es, dass Sie Fragen haben, sich über weitere Tutorials oder Videos informieren möchten, sich SEBLOD in der Praxis anschauen wollen oder sich einfach mit netten Menschen austauschen möchten: Sie sind herzlich willkommen!

*<http://seblod.time4mambo.de/>*

---

<sup>45</sup> <http://www.joomlaportal.de>

## C.2 Tutorials

Paul Brand hat auf seinen Seiten ein sehr umfangreiches und wirklich tolles Tutorial zum Thema Userprofile und Mitgliederverwaltung geschrieben und stellt dieses kostenlos zur Verfügung:

<http://seblod.inetserv.net/de/>

Eine noch sehr neue Seite mit einigen SEBLOD-Tutorials stammt von Christian Hamberger und seinem *Joomclub*:

<http://joom.club/de/joomla/seblod>

## C.3 Videos

Es gibt mittlerweile einige wirklich gute Videos zu SEBLOD. Besonders erwähnen muss man die Videos von Kiki, der nicht ohne Grund hier ein Gastkapitel erhalten hat!

<http://www.youtube.com/user/ScreendesignWhykiki>

Mit die Ersten, die im deutschsprachigen Raum über SEBLOD berichteten, waren David Jardin und Niels Nübel. SEBLOD für Fortgeschrittene:

<https://www.youtube.com/watch?v=tucmw002nX0>

## C.4 SEBLOD-Erweiterungen

Das SEBLOD-Team hat uns zum CCK noch einige Erweiterungen gegeben, die teilweise kostenpflichtig sind und teilweise kostenlos. Die kostenpflichtigen Erweiterungen bewegen sich in der Regel zwischen 5 und 30 Euro. Wirklich viel verdienen werden die Jungs aus Frankreich damit bestimmt nicht – dafür, dass sie uns so eine mächtige Komponente kostenlos zur Verfügung stellen:

<http://www.seblod.com/products>

Des Weiteren finden sich ein paar Erweiterungen bei den Joomla!-Extensions:

<http://extensions.joomla.org/search?q=seblod>

Eine Seite, auf der es fertige Apps zum Download gibt, ist mir nicht bekannt. Aber es ist da etwas in Planung in der deutschsprachigen Community ...

# Stichwortverzeichnis

## Symbole

#\_ 34  
#\_\_categories 68  
#\_\_cck\_store\_form\_ 68  
#\_\_cck\_store\_form\_article 75  
#\_\_cck\_store\_item\_ 68  
#\_\_cck\_store\_item\_users 68  
#\_\_content 34, 68  
#\_\_usergroups 68  
#\_\_users 68  
#name\_des\_feldes# 220  
#SIDEBODY-A 47  
\$cck 276, 279  
\$cck->id 280  
\$option->get 269  
:: 65  
:cck:: 284  
\_blank 40

## A

Abgerundete Ecken 126  
Abhängigkeit 176  
    Checkboxes 238  
ACL 81, 83, 96  
Add Button 91  
Add-ons 73, 184  
addStyleSheet 278  
Admin Formular 43, 46, 54  
Administrationsmenü 184  
Adminmenü 183  
Adminmodule 184

Ajax 122, 256  
    Availability 171  
Aktion 160  
Alias 33, 65, 213  
Aliasnamen 99  
alt-Attribut 38  
Alternativtext 38  
App 23, 192  
App Optionen 79  
App Ordner 210  
App Ordner Core 260  
App Prozeß 80  
App Root (1. Ebene) 30  
App-Folder 23, 99  
Apps installieren 20  
Archiviert 83  
Article Category Id 233, 292  
Article-App 54  
Article-Title 234  
Artikel 294  
Artikel (Ähnliche) 234  
Artikelnummer 158  
Associations 93  
Ausgabe der Formate 85  
Auslöser 59  
AUTO\_INCREMENT 72  
Automatische Speicherung 85  
Automatische Weiterleitung 157  
Autor 74

## B

Barrierefreiheit 38

Bearbeiten 83  
Bearbeitungssymbol 97  
Beez 3 29  
Benutzer 25  
Benutzerdefiniert 65  
Benutzergruppen 25, 119  
Benutzericon 251  
Berechtigungen 81, 96, 122, 145, 223, 251  
Beschreibung 99  
Beschreibungstexte 48  
Bezeichnung 35, 99  
Bild 174  
blank 109  
Blog 152  
Bootstrap 285  
BOTTOM 48  
Braillegeräte 38  
Brand, Paul 6, 21, 296  
Breadcrumbs 241, 244  
Brotkrumen (s. Breadcrumbs)  
Brücke 284  
Buttons 68

## C

Cache 155  
Category 53  
Category Parent Id 292  
category zebra table 154  
catid 242  
CCK 15, 160  
CCK 5  
CCK Feld - Wysiwyg Editor 291

- cck\_storage\_location 284
- change mode 136
- chmod 136
- clear 48
- com\_cck\_ecommerce 186
- config.php 276
- Content Construction Kid
  - 5, 15
- Core 69, 70
- Core-Elemente 205
- Corell, Pierre 21
- CSS 285
- CSS-Dateien 259
- CSS-Klassen 40, 100, 181
- CSV-Datei 73
- CSV-Daten 63
- CSV-Format 63
  
- D**
- Daten 62
- Datenbank 62, 192
- Datenbankabfragen 62
- Datenbanktabelle 20, 62, 193
- Datenfelder 67
- Datensätze 62
- Datentyp 70
- Datenzelle 62
- Datetime 70
- de-DE.com\_cck.ini 95
- Default Inhaltstyp 91
- dfContact 208
- Disable List Template 155
- Dokumentation 28
- Doppelte Doppelpunkte 66
- Downloaden 79
- Duplicate entry 260
- Dynamisch 193
  
- E**
- Easteregg 244
- E-Commerce 186
- Eigene Attribute 100
- Eigene Inhalte bearbeiten 83
- Eigene Tabelle 192
- Eigentümer 96
- Einfach (Simple) 212
- Einleitungen 152
- Elemente 31, 152, 163, 208
  - Beschriftung 158
- Email 213
- E-Mails 207
- embed 138
- Endgültige Sortierung 243
- Entwicklungs-Präfix 88
- Entwicklungsspeicher 89
- Entwicklungs-Speicherung 256
- Erlaubt 82
- Erlaubte Erweiterungen 36
- Erstellen 82
- Erweiterter Modus 38
- Erweiterung 15
- Erzeuge Tabelle 71
- Excel-Tabelle 63, 73
- Exporter 107
  
- F**
- Featured 52
- Fehlermeldungen 159
- Fehlersuche 156
- Feld (Formular) 57
- Feld CCK 160
- Feld X 127
- Felder 44, 153
- Feldtyp 70
  - Bild 35
  - Feld X 57
- Kategorie 52
- Link 39
- Medien 54
- Text 33
- Textbereich 39
- WYSIWYG-Editor 39
  
- FlexiContact 208
- Forciere 92
- Format 63, 69
- Format/Präcision 180
- Formulare 15, 23
- Formulare & Inhaltstypen 43
- Framesets 109
- Frei 68
- Führender 152
- fulltext 65
- Fußzeile 48
  
- G**
- GeolP 103
- Gesperrt 83
- getDescription 280
- getDocument 278
- getHTML 279
- getLabel 279
- getText 280
- getTypo 279
- getValue 279
- Größe 36
- Größenangaben 56
- Grundeinstellungen 144
- Grundlegende Optionen 164
- Gruppe X 128
  
- H**
- h1 55, 157
- h2 55, 158

- Hamberger, Christian 296
- Hauptbeitrag 52
- Haupteintrag 31
- HEADER 48, 56
- Homepage 294
- HTML 65, 211
- I**
- id 235
- ID 62, 72
- Iframe 138
- Image 36, 174
- Image-Feldtyp 292
- images 38
- index.html 262
- Informationen 62
- Inhalt 15, 43, 46
  - erstellen 156
  - vorbereiten 88
- Inhaltserstellung 224
- Inhaltssprache 93
- Inhaltstyp 76, 151
- Inkludiere Seiten-Name 250
- Integration 91
- Intro 43, 46
- introtext 41
- ItemID 241
- J**
- Jardin, David 296
- JavaScript 67, 98, 100
- JForm 118
- JForms 54
- Joomla User - [ID] 284
- Joomla! 65
  - multidomainfähig 17
- Joomla!-Bibliothek 52
- jSEBLOD 17
- Json 67, 257
- JText 85, 94
- K**
- Kanonische Seite 97
- Kategorie 74
- Kategorieblog 26, 152, 201, 235
- Kategorieliste 151, 152, 153
- Kategorien 25
- Kein 68
- Kein Ergebnis 159
- Kiki 207
- Kollation 72
- Komponenten 15
- konfiguriere 237
- Konstruktion 35
- Kopfzeile 48, 56
- L**
- Label 99, 210
- Landing Page 113
- lang 237
- Leere ACL 83
- LEFT 48
- Limit 155
- Linke Spalte (Breite) 47
- Links 152
- Linsner, Christian 6
- Liste 151, 153, 158, 163
- Listen 23
- Listen & Suchtypen 151, 187
- Live Wert 234
- loadposition 242
- localhost 71
- Löschen 83
- M**
- MAINBODY 43, 46
- mainbody.php 265
- markup 269, 279
- Markup 288
- markup.php 269
- max\_input\_vars 293
- Maximalgröße 35
- Media 95
- Mehrfachkategorieauswahl 120
- Mehrsprachigkeit 118, 237
- Menü Konstruktion 28
- Meta 251
- Microsoft-Server 61
- MindMap 226
- Mindmeister 226
- Miniaturbild 38, 55
- Mitteilungen 159
- Mittel 100, 285
- Modalbox 175
- Modul 125, 234, 235
- moduletable 125
- Modulposition 235, 263
- More Link 242
- Multidomain 96
- Multidomainfähig 17
- Multilanguage 93, 118
- Multisites 89, 294
- MySQL 61
- MySQL-Datenbank 61
- MySQL-Dokumentation 70
- N**
- Nachrichten 159
- Nachrichtenstil 141, 159
- Navigation 22
- Neco Papakci 6
- Nested 123
- News-App 23
- Nübel, Niels 296
- Null 162
- Nummer 45, 160
- Nummer 2 237
- Nummer 5 201
- Nutze Allgemein 156

**O**

Objekt 294  
 Octopos 6  
 Offline 258  
 Onlineshop 139  
 option.xml 268  
 Optionale Variante 156  
 Optionen 47, 85, 256  
   Felder 256  
   Multisites 257  
 Ordering 163  
 Override 121  
   (Live/Variation) 164

**P**

pagination 159  
 Panel 114  
 Paniermehl 244  
 Papierkorb 83  
 parent 40, 109  
 parent\_id 119  
 Pflichtfeld 33  
 PHP 67  
 php.ini 293  
 phpMyAdmin 71  
 Pixelkultur 6  
 Placebo 160  
 Plesk 248  
 PLG\_CCK\_FIELD\_GROUP  
   282  
 Plug-in  
   Bestätigung 281  
   Download 283  
   Erweiterte Optionen  
     281  
   Export 283  
   Gruppe 282  
   Site (Multi-sites): 256  
   Speicherung 281  
   System - Sprachenfilter  
     93  
   Typografie 281

Validation 281  
 Plus 55  
 Portfolio 16  
 Positionen 24, 46, 65, 153  
 PrepareContent 139  
 PrepareForm 138  
 PrepareStore 139  
 Prozesse 35, 36  
 Pull-down-Felder 242

**Q**

Query 193

**R**

Random 243  
 Raw Rendering 164  
 Rechte Spalte (Breite) 47  
 Rechtesystem 248  
 Rechteverwaltung 22, 81  
 Reihenfolge 163  
 Relativer Pfad 279  
 renderField 280  
 rendering.php 275  
 Rendering-Klasse 276  
 Responsives Webdesign  
   97  
 Revision 90

**S**

Save \$ Redirect 107  
 Save & Skip 107  
 Schlagwörter 122  
 Schnelle Ordner 261  
 Schnellstartmenü 183  
 Schülling, Christian 6, 296  
 Screendesign Whykiki 6  
 SD Databaser 195  
 Sebastien Lapoux 6  
 SEBLOD 5  
 SEBLOD-Version 17  
 SEF 156

Seitenanzahl 156  
 Seitenformular 43, 46  
 Seitennummerierung 159  
 Seitenteil-a 47  
 Seitenzahl 156  
 Selbst 156  
 self 40, 109  
 SEO 156, 235  
 Shopsystem 186  
 Shortcut 185  
 Spalte anhängen 70  
 Spalten 62, 152  
 Spaltennamen 63  
 Speicherort Entwicklung  
   257  
 Speicherposition 68  
 Speicherung 292  
 Sprachen 86  
 Sprachschlüssel 94, 282  
 Sprachversionen 21  
 SQL 61, 102, 193  
 SQL-Fehler 292  
 Standard Autor 86  
 Standard Reihenfolge 157  
 Standardeditor 135  
 Standardsortierung 243  
 Standard-Wert 33  
 Statische Optionen 103  
 Status 30, 59, 74  
 Status bearbeiten 83  
 Struktur 23  
 Suche 24, 151  
 Suchergebnisse 153  
 Suchformular 153, 155  
 Suchmaschinen-  
   optimierung 158  
 Symbol 31  
 Systemmeldungen 159

**T**  
 Tabelle 62, 154

- Tabellenstruktur 69
  - Table 152, 153
  - Table Layout 154
  - target 40
  - Task 107
  - Template 24, 153, 154
    - Blog 260
    - client 274
    - Computation 270
    - Conditional 270
    - content 264
    - desc 270
    - fieldset 268
    - files 275
    - html 270
    - index.php 276
    - Inhalt 264
    - Inhaltstyp 272
    - Kein 152
    - label 270
    - method 274
    - Override 262
    - Positionen 263
    - Suchtyp 272
    - tbody.php 264
    - Varianten 259
    - Variation 267
  - Template Style does not exist 293
  - template\_preview.png 273
  - template\_thumbnail.png 273
  - templateDetails.xml 274
  - Textbereich 210
  - Textfeld 134
  - Time-out 20
  - title 33
  - Toolbox-Add-on 107
  - top 40, 109
  - Tr (Gerade) 154
  - Trenner 74
  - Trigger 229
  - Trigger Event 180
  - Typografie 55, 174
  - Typografie Bild 292
- U**
- Überschrift 157
  - UNIQUE 72
  - Unter Bezeichnung 153
  - Unter dem Feld 153
  - Unter Formular/Wert 153
  - Update 69, 293
  - URL Assignment 241
  - URL Query (Vars) 150, 164
  - URL zusammensetzen 100
  - utf8\_general\_ci 72
- V**
- Variable 237
  - Verborgen 233
  - Vererbt 82
  - Veröffentlicht 83
  - Version vor der Wiederherstellung 90
  - Versionen 89
  - Verweigert 82
  - Verzeichnisrechte 136
  - Voreinstellungen 85
- W**
- Weiterlesen 94
  - Whykiki 207, 296
  - Wiederherstellung 90
  - WYSIWYG-Editor 30, 291
- X**
- XML 268
- Y**
- YouTube 137, 207
- Z**
- Zeichen 31
  - Zeichencodierung 72
  - Zeige Beschreibung 158
  - Zeige Link 40
  - Zeige Löschen 35
  - Zeige Seitenzahl 159
  - Zeige spezifischen Pfad 37
  - Zeige Text 40
  - Zeige Titel 157
  - Zeige Vorschau 36
  - Zeige Ziel 40
  - Zeilenumbruch 65
  - Zugriffsebenen 120