

1 Die Speicherung und seine 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 eine neue oder ganz anders – SEBLOD™ gibt da einige Möglichkeiten vor und jede hat ihre eigene Auswirkung.

Der Speicherbereich ist sehr wichtig, gleichermaßen bisweilen mysteriös. Fragt man jemanden, der sich mit SEBLOD™ ein wenig auskennt, dann bekommen Sie fast immer die Antwort: "Der Speicherbereich ist ein Kapitel für sich". Wie aber dieses "Kapitel" genau funktioniert, weiß kaum einer. Leider ist auch die Dokumentation gerade in diesem so wichtigen Bereich auf www.Seblod.com sehr dürftig.

Bringen wir also Licht ins Kapitel.

1.1 EinBlick 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. SQL¹ ist nichts weiter als eine Datenbanksprache.

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

Die Datenbank selber 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 einen Text in den Editor beispielsweise eingeben, ist der 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™ selber legt von vornherein bereits einige Datenbanktabellen an, die zur Verwaltung der Erweiterung selber benötigt werden.

¹ SQL = Structured Query Language

2 Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!

Eine Tabelle besteht aus Spalten. Diese Spalten wiederum haben Namen und das sind exakt die Namen, die Sie angeben müssen (neben der Datenbanktabelle, wo diese Spalte steht). Damit Joomla und oder SEBLOD™ das hinterher auch alles einwandfrei zuordnen kann, 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*.

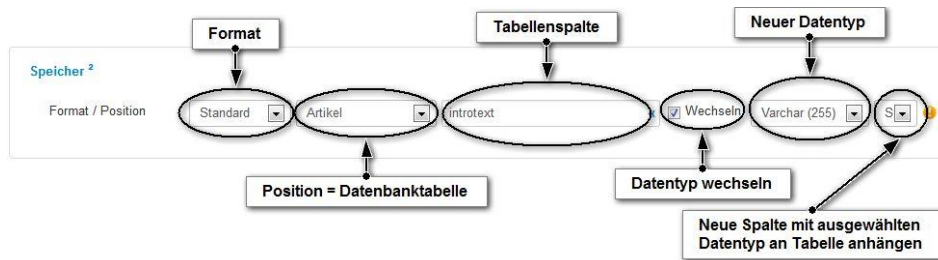
	id	asset_id	title	alias	introtext	fulltext
	13	95	Kultur vom Feinsten	kultur-vom-feinsten	...:19:/cck:; :seitenbilder:2: consectetur a /seitenb...	
	14	97	Kulturnews	kulturnews	...:20:/cck:; :newsverweis: consectetur a /newsverwe...	
	15	99	Neues aus Berlin	neues-aus-berlin	...:21:/cck:; :seitenbilder:2: consectetur a /seitenb...	
	16	102	Finanzen News	finanz-nachrichten	...:22:/cck:; :seitenbilder:2: consectetur a /seitenb...	

1.1: 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 der 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 Gelegenheit bietet eine Menge Möglichkeiten. Vielleicht haben Sie eine eigene Datenbanktabelle erstellt oder 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 hier weitere großartige Möglichkeiten geschaffen worden. Microsoft-Excel-Tabellen lassen sich beispielsweise im CSV-Format abspeichern und dann mit Hilfe von SEBLOD™ in die Datenbank laden. Dort wird eine neue Tabelle angelegt, auf die Sie dann wiederum mit von Ihnen angelegten SEBLOD™-Feldern zugreifen können.

Um den Inhalt eines Feldes abzuspeichern, sind verschiedene Angaben notwendig.



1.2: Der Speicherbereich eines Feldes

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 nach dem, was Sie zuvor ausgewählt haben, erzeugen Sie eigene Spalten, hängen die an bestehende Tabellen dran – erweitern also bestehende Tabellen oder erzeugen lediglich in einer ganz bestimmten SEBLOD™-Tabelle eine neue Spalte. In den nachfolgenden Seiten wird das alles ein wenig entknotet.

1.2 Speicherformat: Standard

Als Beispiel erstelle ich ein neues Formular zu unserer App: News und zunächst einem einzigen Feld, welches ich im Admin Formular anzeigen lasse.

Titel *	<input type="text" value="Ein Feld zum testen"/>	App Ordner	<input type="text" value="News"/>	Be
Name	<input type="text" value="ein_feld_zum_testen"/>	Typ	<input type="text" value="Text"/>	

Konstruktion 1

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"/>		

Speicher 2

Format / Position

1.3: Das Feld...

4 **Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!**

Titel * App Ordner

Admin Formular | Seitenformular | Intro | Inhalt | Konfiguration | **Felder** | Templates

Ansichten | Optionen

Konstruktion (für Admin Formularansicht)

	Bezeichnung	Variante
# MAINBODY	<input type="text"/>	- Standard -
Ein Feld zum testen (Text)	<input type="text" value="Ein Feld zum testen"/>	Standard
# TOPBODY	<input type="text"/>	- Standard -

1.4: ...eingebaut im Admin Formular

Erzeuge ich nun einen neuen Beitrag in unserem Datenbanktest, welches ja nur das eine Felde hat und schreibe dort "Hochwichtigen Text" hinein, dann wird dieser Text auch genauso in der Datenbank abgespeichert.

fulltext mediumtext

1.5: 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 auch genau an dieser Stelle gespeichert.

Für das Format *Standard* gelten vier Bedingungen:

1. Es muss eine der *vorgegebenen Tabellen* benutzt werden
2. Es handelt sich nicht um ein Sammelfeld/Sammelspalte (siehe dazu das nächste Kapitel: *Speicherformat: Benutzerdefiniert*)
3. Es ist eine vorhandene Tabellenspalte in den *vorgegebenen Tabellen*
4. Oder es wird ein *Wechsel* des Datentyps vorgenommen und eine *freie* Tabelle ausgewählt (näheres weiter unten)

Unter *vorgegebenen Tabellen* werden die bezeichnet, die Sie bei den *Positionen* im Pulldwonbutton unter dem Bereich *Joomla!* auswählen können

Format / Position

Frei

Joomla!

Artikel

Benutzer

Benutzergruppe

Kategorie

1.6: Eine der vorgegebenen Tabellen

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

1.3 Speicherformat: Benutzerdefiniert

Wenn Sie unter *benutzerdefiniert* abspeichern, dann ändert sich das Speicherformat dahingehend, dass Ihre Informationen vom Aliasnamen Ihres Feldes eingefasst wird:

```
<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 Alias-Namen des Feldes eingefasst. Wobei der Aliasname sowohl am Anfang, als auch mit einem Ende-Slash, wie es bei HTML-Elementen auch benutzt wird, am Ende steht. Statt der spitzen Klammern, wie Sie es aus HTML her 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 Alias-Feld gehören, anzeigt.

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

Speicher ²

Format / Position Wechseln

1.7: Beide Felder werden gleich abgespeichert

Konstruktion (für Admin Formularansicht)

	Bezeichnung	Variante
# MAINBODY	<input type="text"/>	- Standard -
Ein Feld zum testen (Text)	<input type="text" value="Ein Feld zum testen"/>	Standard
Ein zweites Feld zum testen (Text)	<input type="text" value="Ein zweites Feld zum testen"/>	Standard

1.8: Und in das Admin Formular eingebaut

6 Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!

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



The image shows a horizontal row of three buttons: a green 'Speichern' button with a floppy disk icon, a 'Speichern & Schließen' button with a checkmark icon, and a '+ Speiche' button with a plus icon. Below the buttons are two input fields. The first is labeled 'Ein Feld zum testen' and contains the text 'Hochwichtiger Text'. The second is labeled 'Ein zweites Feld zum testen' and contains the text 'Noch viel wichtigerer Text'.

1.9: Beitrag eingeben



The image shows a form with a dropdown menu on the left and a code preview on the right. The dropdown menu has 'fulltext' and 'text' as options. The code preview shows the following HTML structure:

```
<br
/>::ein_feld_zum_testen::Hochwichtiger
Text::/ein_feld_zum_testen::<br
/>::ein_zweites_feld_zum_testen::Noch
viel wichtigerer
Text::/ein_zweites_feld_zum_testen::
```

1.10: So wird es dann in der Datenbank gespeichert

Der große Vorteil an dieser Art der Speicherung ist es, dass Sie mehrere unterschiedliche Datenfelder in eine Tabellenspalte speichern können und es SEBLOD™ überlassen, alles anschließend wieder der richtigen Stelle zuzuordnen. Diese Art der Speicherung macht besonders auch dann Sinn, 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™ selber benutzt für diese "Sammelspeicherungen" sehr gern die Tabellenspalte *introtext*.

Wenn Sie nun ein Feld auf den *fulltext* in unserem Beispiel auf *Standard* setzen würden, würde Ihnen alles, was dort jetzt drin steht angezeigt. Also der gesamte Text mit den SEBLOD™-Formatierungen und so, wie er im Bild 1.10 abgebildet ist. Denn der *Standard* unterscheidet nicht die einzelnen Aliasnamen, sondern zeigt das so an, wie die Daten dort drin stehen.

1.4 Speicherformat: Json

Das Speicherformat *Json*² ist fast beinahe identisch mit dem Speicherformat *benutzerdefiniert*. Der eine Unterschied ist das etwas andere Format:

```
{"ein_feld_zum_testen": "Hochwichtiger  
Text", "ein_zweites_feld_zum_testen": "Noch viel wichtigerer  
Text"}
```

Also keine doppelten Doppelpunkte und kein Anfang und Ende. Dieses ist mehr eine Auflistung, die mit Kommata getrennt werden und die einzelnen Inhalte in Anführungsstrichen eingefasst sind. Der ganze Block ist mit geschweiften Klammern eingefasst.

Rein Inhaltlich ändert sich nichts im Vergleich zur *benutzerdefinierten* Speicherung. Auch hier gilt all das, was ich im Kapitel vorher geschrieben habe. Der eigentliche 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, dann macht dieses Speicherungsformat vermutlich nicht allzuviel Sinn. Denkbar wäre vielleicht noch, dass Sie ein Plugin oder eine andere Erweiterung benutzen, die dieses Speicherungsformat voraussetzt. Allerdings sind mir derzeit keine solche Erweiterungen im Zusammenhang mit SEBLOD™ bekannt.

1.5 Speicherformat: Kein

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

1.6 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*.

² JSON steht für: JavaScript Object Notation

8 Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!

Zu jedem dieser Bereiche ist eine bestimmte Datenbanktabelle zugeordnet:

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

Wählen Sie also beispielsweise die *Artikel* unter Position aus, werden Ihre Daten automatisch in die Datenbanktabelle #__content gespeichert. Das "#_" steht hier auch wieder für eine beliebige Prefix. Siehe dazu auch Kapitel 3, Seite 6

Wenn Sie eine andere Tabelle benutzen möchten, dann wählen Sie *Frei* aus. Danach erhalten Sie ein zusätzliches DropDown-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 ist als einzige hier vorgegeben. Wenn Sie die News angelegt haben, dann 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, dann legt SEBLOD™ eine neue Tabelle mit dem Namen #__cck_store_item_IHR-FORMULARNAME an und erzeugt dort eine Spalte mit dem "unbekannten" Namen.

Als Beispiel habe ich hier ein kleines Formular mit dem Namen *Datenbankspeicherung* erstellt. Es gibt in diesem Formular nur ein Textfeld, welches ich der *Artikel-Tabelle* mit dem Spaltennamen *unbekannte_spalte* zugeordnet habe.



1.11: 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.



1.12: \$\$Unsere Daten in der Datenbank

Am Rande sei angemerkt, dass es natürlich egal ist, welches *Format* Sie dafür auswählen. Sowohl *Standard*, als auch *Benutzerdefiniert* ist dafür benutzbar. Das *Format* hat Einfluß auf das *WIE* – und nicht auf das *WO*.

Der Vorteil ist, dass Sie damit die Joomlaeigenen Datenbanktabellen in Ruhe lassen können und ausweichen auf eigene Tabellen, 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".

1.7 Datentyp wechseln

Standardmäßig werden neue Daten so abgespeichert, wie sie vom *Feldtyp* vorgegeben sind. Ein *Link-Feld* speichert eine *URL*, ein *eMail-Feld* eine *eMail*, ein *Bild-Feld* einen *image-Pfad*, ein *Text-Feld* einen *Varchar (255)*, ...

Soweit 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 Sinn machen, 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 Sinn machen, dass auch gleich als solches abzuspeichern, in dem Sie *Datetime* auswählen. Oder Sie geben über das Textfeld

Zahlen ein und können dann dieses auch als Zahlen 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 zur Verfügung gestellt wird pro Datensatz – auch wenn gar nichts drin steht. Wer also nun auf Nummer sicher gehen will und überall das höchste auswählt, erzeugt so eine große Datenbank, mit hoher Speicheranforderung, die auch erst mal geladen werden muss, wenn entsprechende Anfragen kommen.

Darum rate ich zu übernehmen, was SEBLOD™ anbietet. Wenn Sie es genau wissen möchten, was dort passiert, empfehle ich einen Blick in die wirklich sehr gute MySQL-Dokumentation, die in deutscher Sprache vorliegt.³

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 aber nicht tun. Wenn Sie aber eigene Tabellen erzeugen oder benutzen, dann ist das vielleicht eine Option. Dann haben Sie Ihre Daten immer zusammen in einer Tabelle, die Sie nach Belieben erweitern können. Allerdings nicht verkleinern. Das geht dann nur in der Datenbank selber.

1.8 Eigene Tabellen erstellen

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

In die Datenbank kommen 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 Tabellenauflistung einen Button mit der Aufschrift "Erzeuge Tabelle". Dort einfach mutig draufgeklickt.

Oben geben Sie einen Tabellennamen ein. Hier ist es sehr wichtig, dass Sie die Prefix 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.

³ <http://dev.mysql.com/doc/refman/5.1/de/data-types.html>

Tabellenname:

Struktur

Spalte	<input type="text" value="id"/>	<input type="text" value="irgendeine_spalte"/>
Typ	<input type="text" value="INT"/>	<input type="text" value="TEXT"/>
Länge/Set	<input type="text" value="10"/>	<input type="text"/>
Standard	<input type="text" value="Kein(e)"/>	<input type="text" value="Kein(e)"/>
Kollation	<input type="text" value="utf8_general_ci"/>	<input type="text" value="utf8_general_ci"/>
Attribute	<input type="text"/>	<input type="text"/>
Null	<input type="checkbox"/>	<input type="checkbox"/>
Index	<input type="text" value="UNIQUE"/>	<input type="text" value="---"/>
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kommentare	<input type="text"/>	<input type="text"/>
MIME-Typ	<input type="text"/>	<input type="text"/>
Darstellungsumwandlung	<input type="text"/>	<input type="text"/>
Umwandlungsoptionen	<input type="text"/>	<input type="text"/>

Tabellen-Kommentar:

Tabellenformat:

Kollation:

PARTITION Definition:

1.13: Eine neue Tabelle in der Datenbank erzeugen

In der Struktur haben Sie Platz, um zwei Spalten vorzugeben. Wobei die erste wichtigerweise 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*. Die finden Sie im Auswahl-Feld ziemlich weit unten im vorletzten Block. Im *Index* tragen Sie noch *UNIQUE* ein. Und aktivieren das Kästchen für den *AUTO_INCREMENT*. Damit haben Sie eine Spalte definiert, die einen Wert nur einmal enthalten kann (*UNIQUE*) und 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

12 Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!

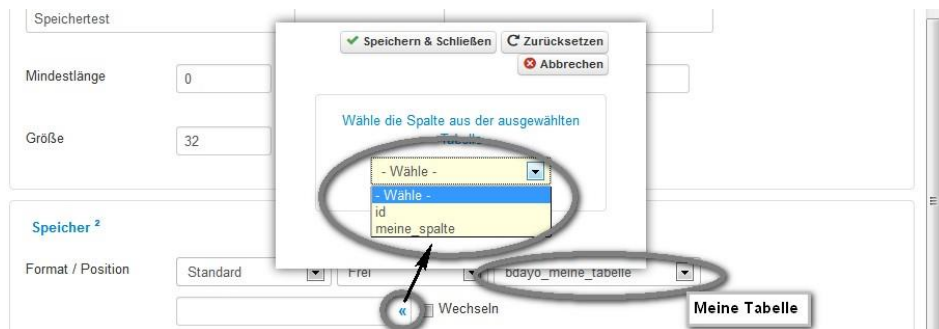
einen fantasiereicheren Namen ersetzen können. Dabei beachten Sie bitte: Keine Sonderzeichen, keine Umlaute und keine Leerzeichen. Statt Leerzeichen benutzen Sie die Unterstriche. Als *Typ* wählen Sie den *TEXT* aus, *Standard* bleibt auf *Kein(e)*, in die *Kollation* wieder unser *utf8_general_ci*.

Wenn Sie möchten geben Sie einen Kommentar zu Ihrer Tabelle ein. Das Tabellenformat lassen Sie bitte auch auf *InnoDB*⁴ stehen.

Das wars auch schon. Jetzt noch beherzt ganz unten links auf *Speichern* drücken und schon haben Sie eine neue Datenbanktabelle angelegt.

Sie kommen zu dieser Eingabemaske auch, wenn Sie sich in der Datenbank-Übersicht 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 das zwei und klicken dann "OK" an.

Ihre neu erstellte Tabelle können Sie wie im Kapitel *Speicherposition* 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 Ihrer bestehenden Tabelle anhängen.



1.14: Die eigene Tabelle in SEBLOD™ benutzen

1.9 CSV-Import

In SEBLOD™2 gab es lange Zeit eine Betaversion eines CSV-Importers. In SEBLOD™3 ist der verschwunden, findet sich aber auf der Extensionsseite direkt auf der SEBLOD™seite wieder. Dort kommen Sie hin, wenn Sie auf der

⁴ <http://dev.mysql.com/doc/refman/5.1/de/innodb.html>

SEBLOD™-Startseite "Download" anklicken und dann im Menü "Products" auswählen. Dort finden Sie den *SEBLOD™ Importer*⁵.

Sie müssen sich registrieren/anmelden zum Download. Nachdem Sie das Add-on wie üblich über den Installer in Joomla installiert haben, finden Sie das Add-on direkt im *Komponentenmenü* als Untermenüeintrag bei *SEBLOD 3.x*.

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

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

1.15: Eine Exceltabelle, 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.

⁵ <http://www.SELOD™.com/v3/products/7>

14 Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!

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 *	Artikel ▼	Standard ▼	
Inhaltstyp *	Article ▼		
File to Import *	D:\DATEN\TEXTE\Buchprojekte\J		Durchsuchen...
Trenner	; ▼		
Force UTF-8	<input type="checkbox"/> Ja <input checked="" type="checkbox"/> Nein		

Standardwerte

Autor	- Nutze Allgemein - ▼
Kategorie	- Nutze Allgemein - ▼
Status	- Nutze Allgemein - ▼

WIE STELLT MAN DIESE ERWEITERUNG EIN?

Import :: from Third-party Component

1.16: Die CSV-Datei importieren

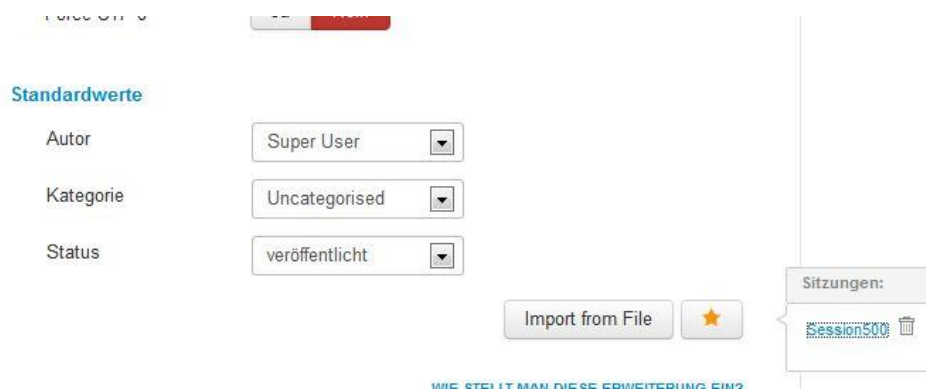
Mit *File to Import* wähle ich meine CSV-Datei auf meiner Festplatte aus.

Wichtig ist der *Trenner*. MS-Excel nimmt hier 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. In meinem Fall stelle ich den *Trenner* auf das Semikolon um.

Unsere Datenbank ist ohnehin als *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.

Unter *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 diese öfter benutzen möchten, können Sie selbige über den orangenen Stern unten rechts speichern. Sie können dann beim nächsten Aufruf des Importers durch Überfahren des Sternes die Voreinstellung erneut abrufen oder eine andere durch anklicken speichern. Allerdings wird die ausgewählte Datei nicht mitgespeichert. Die müssen Sie mittels "Durchsuchen..." jedesmal erneut auswählen.



1.17: 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 diese noch nicht existiert und hängt dort die drei neuen Felder nebst Inhalt an:

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

1.18: 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

16 Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!

neuen Tabelle verknüpft sind und 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
...			2013-05-09-13-52-59 Kategorie: Uncategorized	Öffentlich	Super User	Alle	09.05.2013	1
...			2013-05-09-13-52-59 Kategorie: Uncategorized	Öffentlich	Super User	Alle	09.05.2013	2
...			2013-05-09-13-52-59 Kategorie: Uncategorized	Öffentlich	Super User	Alle	09.05.2013	3

1.19: Drei neue Beiträge

Die drei neuen Felder erscheinen am Ende des Beitrages:

Introtext

Pfad: p

Fulltext

Buchtitel

Autor

Verlag

1.20: 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 und somit Ihre CSV-Daten angezeigt werden. Beziehungsweise: Sie können natürlich auch vor dem Import einen passenden *Inhaltstyp* erstellen, den Sie dann dem Importer zuweisen. Was Sie dabei beachten müs-

18 **Kapitel Fehler! Verweisquelle konnte nicht gefunden werden. Fehler!**

sen, 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.

#__categories 6
 #__cck_store_form_ 6
 #__cck_store_form_article 11
 #__cck_store_item_ 6
 #__cck_store_item_users 6
 #__content 6
 #__usergroups 6
 #__users 6
 :: 4
 Add-on 10
 Alias 4
 AUTO_INCREMENT 9
 Autor 10
 benutzerdefiniert 4
 Buttons 5
 Core 7, 8
 CSV 2
 Daten 1
 Datenbank 1
 Datenbanktabelle 2
 Datenfelder 4
 Datensatz 2
 Datentyp 7
 Datenzelle 2
 Datetime 7
 doppelte Doppelpunkte 4
 Erzeuge Tabelle 8
 Excel 2, 10
 Feldtyp 7
 Format 2, 7
 Frei 6, 9
 fulltext 3
 HTML 4
 ID 2, 9
 Informationen 1
 Inhaltstyp 11
 Javascript 5
 Joomla! 3
 Json 5
 Kategorie 10
 Kein 5
 Kollation 9
 localhost 8
 Microsoft 2
 Microsoft Server 1
 MySQL 1
 MySQL-Dokumentation 8
 PHP 5
 phpmyadmin 8
 Positionen 3
 Spalte anhängen 8
 Spalten 2
 Spaltennamen 2
 Speicherposition 6
 SQL 1
 Standard 3
 Status 10
 Tabelle 2
 Trenner 10
 UNIQUE 9
 utf8_general_ci 9
 Wechseln 9
 Zeichencodierung 9
 Zeilenumbruch 4