

# Extensions für **TYPO3**

So entwickeln Sie maßgeschneiderte TYPO3-Erweiterungen

- > Die TYPO3-Programmierschnittstelle beherrschen
- > Extensions sauber programmieren und der Community zur Verfügung stellen
- > Das bringen TYPO3 5.0 und FLOW3

Inklusive einer kompletten Beispiel-Extension  
zum kostenlosen Download

## **Bibliografische Information der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

## **© 2010 Franzis Verlag GmbH, 85586 Poing**

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

**Herausgeber:** Franz Graser

**Satz:** DTP-Satz A. Kugge, München

**art & design:** [www.ideehoch2.de](http://www.ideehoch2.de)

**Druck:** Bercker, 47623 Kevelaer

Printed in Germany

**ISBN 978-3-645-60000-2**

# Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitung</b> .....	<b>9</b>
1.1	Einleitung .....	9
1.1.1	Voraussetzungen .....	9
1.1.2	Einrichten von TYPO3 als Entwicklungsumgebung.....	10
<b>2</b>	<b>Planung der Erweiterung</b> .....	<b>13</b>
2.1	Anforderungen sammeln.....	13
2.1.1	Das Pflichtenheft der Beispiel-Extension .....	13
2.1.2	Technische Anforderungen.....	14
2.1.3	Die Datenbankstruktur .....	15
2.2	Umsetzung des Entwurfs in TYPO3 .....	17
2.2.1	Der Extension Key .....	17
2.2.2	Veröffentlichen von Extensions .....	19
2.2.3	Anpassung der Datenbankstruktur.....	21
<b>3</b>	<b>Struktur einer Extension</b> .....	<b>27</b>
3.1	Kategorien von Extensions .....	27
3.1.1	Backend .....	28
3.1.2	Backend Modules .....	29
3.1.3	Frontend .....	29
3.1.4	Frontend Plugins .....	29
3.1.5	Services .....	29
3.1.6	Documentation .....	29
3.1.7	Examples .....	30
3.1.8	Templates .....	30
3.1.9	Miscellaneous.....	30
3.1.10	Kategorie [ ].....	30
3.2	Dateistruktur einer Extension .....	31
3.2.1	ext_emconf.php .....	31
3.2.2	ext_localconf.php .....	32
3.2.3	ext_tables.php.....	32
3.2.4	ext_tables.sql .....	33

3.2.5	ext_tables_static+adt.sql.....	33
3.2.6	ext_typoscript_constants.txt.....	34
3.2.7	ext_typoscript_setup.txt.....	34
3.2.8	ext_typoscript_editorcfg.txt.....	34
3.2.9	ext_conf_template.txt.....	34
3.2.10	ext_icon.gif.....	35
3.2.11	class.ext_update.php.....	35
3.2.12	Sprachdateien.....	36
3.2.13	Ordnerstrukturen.....	36
3.2.14	Statische Templates.....	37
<b>4</b>	<b>Der Extension Kickstarter.....</b>	<b>39</b>
4.1	Aufbau des Kickstarters.....	39
4.1.1	General info.....	40
4.1.2	Setup languages.....	40
4.1.3	New Database Tables.....	41
4.1.4	Extend existing Tables.....	52
4.1.5	Frontend Plugins.....	54
4.1.6	Backend Modules.....	57
4.1.7	Integrate in existing Modules.....	59
4.1.8	Clickmenu items.....	59
4.1.9	Services.....	60
4.1.10	Static TypoScript code.....	62
4.1.11	TSconfig.....	64
4.1.12	Extension schreiben.....	65
4.2	Dateien des Kickstarters.....	67
4.2.1	ext_emconf.php.....	68
4.2.2	ext_localconf.php.....	75
4.2.3	ext_tables.php.....	78
4.2.4	tca.php.....	81
4.2.5	locallang_db.xml.....	83
4.2.6	ext_tables.sql.....	84
4.2.7	Icons.....	85
4.2.8	Texte.....	85
4.2.9	Konfiguration einer Extension.....	86
<b>5</b>	<b>Das Frontend-Plugin.....</b>	<b>89</b>
5.1	Ausgangsbasis.....	89
5.2	Frontend-Funktionen.....	91

5.2.1	tslib_pibase .....	91
5.2.2	Weitere Frontend-Klassen .....	92
5.3	Plugin-Konfiguration .....	93
5.3.1	Konfiguration via Typoscript.....	93
5.3.2	Konfiguration via Template Editor .....	93
5.3.3	Konfiguration via Flexform.....	95
5.3.4	Auswerten der Konfiguration .....	103
5.3.5	Unterstützung von Templates.....	104
5.4	Lokalisierung .....	106
5.4.1	locallang.xml .....	107
5.4.2	Lokalisierte Texte im Plugin.....	107
5.4.3	Lokalisierte Templates .....	108
5.4.4	Lokalisierte Einträge .....	108
5.4.5	Verwendung von Zeichensätzen.....	109
5.5	Caching in Plugins .....	110
5.5.1	Der TYPO3-Cache .....	110
5.5.2	Caching in Plugins .....	111
5.5.3	Caching abschalten .....	112
5.6	Der Plugin-Code .....	114
5.6.1	Beispielcode säubern .....	114
5.6.2	Sichtbarkeitsattribute (access identifier).....	116
5.6.3	Die Funktion main().....	116
5.6.4	Die Listenansicht .....	118
5.6.5	Die Einzelansicht .....	124
5.6.6	Erweitern der Extension-Funktionen .....	131
5.6.7	Zusammenfassung.....	134
<b>6</b>	<b>Backend-Programmierung .....</b>	<b>135</b>
6.1	Das Backend-Formular .....	135
6.1.1	ext_tables.php / tca.php.....	135
6.2	Das Backend-Modul.....	139
6.2.1	Die Ausgangsbasis.....	139
6.2.2	Backend-Klassen .....	141
6.2.3	Das Objekt \$GLOBALS['BE_USER'].....	144
6.2.4	Verwendung des System-Logs.....	148
6.2.5	index.php .....	151
6.2.6	mod_template.html .....	161

<b>7</b>	<b>Fertigstellen der Extension.....</b>	<b>163</b>
7.1	Code Review .....	163
7.1.1	Funktionalität prüfen.....	163
7.1.2	Code prüfen .....	163
7.2	Dokumentation .....	166
7.2.1	Handbuch-Struktur .....	167
7.2.2	Verwendung der Formatvorlagen .....	168
7.3	Veröffentlichen der Erweiterung .....	170
<b>8</b>	<b>Project Coding Guidelines.....</b>	<b>173</b>
8.1	Project Coding Guidelines .....	173
8.1.1	Konventionen für Benennung und Formatierung.....	173
8.1.2	Variablen und Konstanten.....	178
8.1.3	Datenbankzugriff .....	180
8.1.4	Sicherheit .....	182
8.1.5	Coding .....	184
<b>9</b>	<b>Die TYPO3-API.....</b>	<b>189</b>
9.1	Die Programmierschnittstelle von TYPO3.....	189
9.1.1	Grundlagen .....	189
9.1.2	Workspaces .....	194
9.1.3	TYPO3 Core Engine (TCE) .....	198
9.1.4	Nützliche Variablen und Konstanten .....	207
9.1.5	AJAX im TYPO3-Backend.....	211
9.1.6	Der Table Configuration Array (TCA).....	216
<b>10</b>	<b>Ausblick.....</b>	<b>263</b>
10.1	Einschätzung .....	263
10.2	MVC als Entwurfsmuster .....	264
10.3	FLOW3 und Extbase .....	265
10.3.1	FLOW3 .....	265
10.3.2	Extbase.....	266
10.3.3	Entwicklungsprinzipien.....	266
10.4	Fluid .....	269
10.5	Kickstarter .....	271
10.6	Zusammenfassung.....	272
	<b>Stichwortverzeichnis .....</b>	<b>275</b>

# 1 Vorbereitung

## 1.1 Einleitung

Dieses Buch ist eine praxisorientierte Hilfe für alle, die den Schritt vom reinen TYPO3-Anwender hin zum Entwickler gehen wollen.

Im Laufe dieses Buches wird der Werdegang einer TYPO3-Extension Schritt für Schritt nachvollzogen – vom Einrichten einer passenden Entwicklungsumgebung bis hin zur Veröffentlichung im TYPO3-Repository.

Basis dieses Buches ist TYPO3 in der Version 4.2.10.

### 1.1.1 Voraussetzungen

Folgende Voraussetzungen und Kenntnisse sollten Sie als Leser dieses Buches mitbringen:

#### **Sicherer Umgang mit TYPO3**

Sie sollten TYPO3 als Administrator kennen und verstehen. Grundwissen zum Thema Typoscript und TYPO3-Extensions sollte vorhanden sein.

#### **Programmierkenntnisse**

Die Programmiersprache für TYPO3 und seine Extensions ist PHP. Sie sollten daher solide Kenntnisse im Umgang mit PHP im Allgemeinen sowie mit Klassen und Objekten im Besonderen haben.

#### **HTML-Kenntnisse**

Die meisten Extensions erzeugen Ausgaben im Frontend. Sicherer Umgang mit HTML und CSS wird daher vorausgesetzt.

### 1.1.2 Einrichten von TYPO3 als Entwicklungsumgebung

Grundsätzlich lässt sich eine TYPO3-Extension mit nicht mehr als einem simplen Texteditor programmieren. Allerdings würde das nicht nur unnötige Zeit kosten, sondern auch viele Fehlerquellen öffnen. Eine sinnvoll eingerichtete TYPO3-Umgebung ist daher für die Extension-Entwicklung äußerst hilfreich. Im Folgenden werden einige Erweiterungen und Dokumentationen beschrieben, die Ihnen helfen, den Weg zur eigenen Extension zu ebnen.

#### Der Extension Kickstarter

Der Extension Kickstarter (*kickstarter*) ist der Ausgangspunkt fast jeder Erweiterung. Er hilft, ein sauberes Grundgerüst für eine Extension aufzubauen – allerdings auch nicht mehr. Es handelt sich nämlich nicht um einen Editor, mit dem sich eine Extension nach dem Erstellen dauerhaft weiter pflegen lässt.

Wir werden den Kickstarter im Verlauf der Entwicklung unserer Erweiterung noch genauer kennenlernen.

#### Der Extension Development Evaluator

Die zweite hilfreiche TYPO3-Extension ist der Extension Development Evaluator (*extdeveval*). Diese Erweiterung stellt eine Reihe von Funktionen zur Verfügung, um Ihre Erweiterung zu säubern und an die TYPO3-Richtlinien anzupassen. Außerdem stellt sie eine Dokumentation der TYPO3-Funktionen zur Verfügung.

Auch diese Erweiterung werden wir beim Aufbau unserer Extension einsetzen.

#### T3Dev

Die Erweiterung T3Dev (*t3dev*) stellt ebenfalls einen Zugang zur Dokumentation zur Verfügung. Darüber hinaus enthält sie ein Tool zum Erstellen der XML-Strukturen für eine Flexform.

#### Dokumentationen

Die wichtigsten Dokumentationen zu TYPO3 stehen im Repository als »Extensions« zur Verfügung, auch wenn sie keinerlei Code enthalten. Stattdessen bringen sie die Dokumentation in Form einer OpenOffice-Datei auf Ihren Rechner, sodass die Informationen auch ohne Internet-Verbindung jederzeit zur Verfügung stehen.

Die Dokumentationen lassen sich mithilfe des Extension Managers problemlos installieren, indem Sie nach »doc\_« suchen. Die wichtigsten Dokumentationen sind:

<i>Dokumentation</i>	<i>Extension Key</i>	<i>Beschreibung</i>
TSconfig	doc_core_tsconfig	Dokumentation der Typoscript-Konfiguration für Page und User TSConfig.
TSref	doc_core_tsref	Die Typoscript-Referenz
TYPO3 Coding Guidelines	doc_core_cg1	Guidelines für die Code-Entwicklung von TYPO3-Extensions
TYPO3 Core API	doc_core_api	Dokumentation der Programmierschnittstellen (APIs) von TYPO3
TypoScript Syntax and In-depth Study	doc_core_ts	Einführung in Typoscript

**Tabelle 1.1:** TYPO3-Dokumentationen



## 2 Planung der Erweiterung

Am Anfang jedes Projekts steht eine sorgfältige Planung, und eine TYPO3-Extension macht hier keine Ausnahme – im Gegenteil. Da jede Erweiterung Bestandteil des umfangreichen und komplexen TYPO3-Systems wird, muss sie über die »normalen« Anforderungen an ein Software-Projekt hinaus auch noch die Regeln erfüllen, die TYPO3 vorgibt. Ohne diese Vorgaben wäre es kaum möglich, die Stabilität und Flexibilität von TYPO3 zu gewährleisten.

### 2.1 Anforderungen sammeln

Am Beginn jeder Planung steht das Sammeln der Anforderungen – noch unabhängig vom Weg der Implementierung. So banal das klingen mag, so häufig wird dieser wesentliche Schritt versäumt oder zumindest zu leichtfertig angegangen. Stattdessen entwickelt sich das endgültige Konzept oft erst bei der Implementierung bzw. der Vorstellung der ersten Funktionen. Allerdings bedeutet das meist unnötige Verzögerungen und führt zu nicht optimalen Ergebnissen, weil an bereits fertigen Funktionen nachträglich Anpassungen erfolgen müssen. Im schlimmsten Fall stellt sich heraus, dass die grundlegende Struktur falsch ist, was dann umfangreiche Umbauarbeit nach sich zieht.

Eine gute erste Planungsphase hilft auch, unterschiedliche Denkweisen auszugleichen. Einfaches Beispiel: Denkt ein Bankangestellter an eine Transaktion, dann meint er meistens den Transfer von Geld. Ein Programmierer hat dagegen bei diesem Begriff eher Datenbanken im Sinn. Es lohnt sich daher, solche Missverständnisse von vornherein auszuschließen und eine gemeinsame Sicht der Dinge zu schaffen.

#### 2.1.1 Das Pflichtenheft der Beispiel-Extension

Hier kommt nun das erste Mal unsere Beispiel-Extension ins Spiel. Im Laufe des Buches wollen wir einen Event Manager entwickeln, eine Erweiterung also, mit der sich Veranstaltungen verwalten lassen. Die Ausgangssituation soll folgendermaßen aussehen:

- Auf der Website sollen Veranstaltungen angekündigt werden, zu denen sich Besucher registrieren können.

- Eine Veranstaltung wird beschrieben mit Titel, Ort und Zeit. Außerdem soll es eine ausführliche Beschreibung geben sowie eine Agenda und einen Anreißtext, beispielsweise für die Homepage.
- Zu einer Veranstaltung gibt es Sponsoren. Auf der Seite einer Veranstaltung soll eine Liste der Sponsoren mit Name und Logo zu sehen sein, die jeweils mit einem kurzen Firmenprofil verlinkt werden.
- Es gibt Haupt- und Nebensponsoren.
- Die Registrierung der Besucher erfolgt per Online-Formular. Dabei ist zu berücksichtigen, dass eine Veranstaltung nur eine begrenzte Zahl von Plätzen hat.
- Jede Registrierung soll eine Mail an einen Mitarbeiter der Firma auslösen. Darüber hinaus soll in der Website-Verwaltung zu jeder Veranstaltung eine Teilnehmerliste abrufbar sein.

Als ergänzende Faktoren kommen dazu:

- Ein Sponsor kann mehrere Veranstaltungen bezuschussen. Das Firmenprofil soll in diesem Fall nicht jedes Mal neu erstellt werden.
- Besucher, die sich bereits einmal registriert haben, sollen ihre Daten nicht neu eingeben müssen.

Es gibt natürlich noch weitere naheliegende Funktionen, die zu unserer Erweiterung passen würden. Wir werden sie hier aber nicht umsetzen, um den Rahmen nicht zu sprengen. Denkbare zusätzliche Funktionen wären:

- Veranstaltungsreihen, also sich wiederholende Termine
- Online-Bezahlungsfunktionen
- Registrierte Besucher können sich nach der Veranstaltung online anmelden, um zusätzliche Daten abzurufen, etwa die Präsentationen.

### 2.1.2 Technische Anforderungen

Über das funktionelle Pflichtenheft hinaus gibt es einige technische Details, die häufig nicht zur Sprache kommen, aber auf jeden Fall berücksichtigt werden sollten. Dazu gehören:

- Einfache Anpassbarkeit des Layouts
- Einfache Konfiguration im Backend

- Unterstützung für mehrsprachige Websites
- Dokumentation

Insbesondere der letzte Punkt gilt als undankbare Aufgabe und wird oft nur kurz oder lieblos erfüllt. Erfahrene Auftraggeber bestehen auf einer guten Dokumentation, doch auch aus Sicht des Programmierers ist sie wichtig. Denn nahezu jedes Projekt holt die Entwickler irgendwann wieder ein, beispielsweise weil neue Funktionen implementiert werden sollen. Und auch wenn man den Code selbst geschrieben hat – ein Jahr später wirkt er so fremd, als hätte man ihn nie gesehen.

### 2.1.3 Die Datenbankstruktur

Die obigen Anforderungen müssen nun in eine Datenbankstruktur umgesetzt werden – zunächst noch unabhängig von TYPO3. Folgende Tabellen sollen die Grundlage bilden:

#### Veranstaltungen

<i>Feld</i>	<i>Typ</i>	<i>Beschreibung</i>
title	varchar(100)	Titel der Veranstaltung
location	varchar(100)	Ort der Veranstaltung
date	date	Termin
teaser	tinytext	Anreißertext
description	text	Beschreibung
agenda	text	Agenda
seats	int	Anzahl Plätze

**Tabelle 2.1:** Datenbanktabelle für die Veranstaltungen

#### Sponsoren

<i>Feld</i>	<i>Typ</i>	<i>Beschreibung</i>
name	varchar(100)	Name des Sponsors
logo	tinytext	Dateiname des Logos
url	tinytext	Link zur Website
profile	text	Firmenprofil

**Tabelle 2.2:** Datenbanktabelle für die Sponsoren

### Teilnehmer

Feld	Typ	Beschreibung
name	tinytext	Name des Teilnehmers
email	varchar(200)	E-Mail-Adresse
street	varchar(150)	Straße / Hausnummer
zipcode	varchar(10)	Postleitzahl
city	varchar(100)	Ort
country	varchar(100)	Land

Tabelle 2.3: Datenbanktabelle für die Teilnehmer

Zu den reinen Datentabellen kommen noch zwei Kreuztabellen, in denen gespeichert wird, welcher Sponsor zu welcher Veranstaltung gehört und welcher Teilnehmer sich wo registriert hat. Kreuztabellen wählen wir deshalb, weil zwischen Veranstaltungen und Sponsoren eine m-zu-n-Beziehung besteht, d. h. eine Veranstaltung kann mehrere Sponsoren haben, ein Sponsor mehrere Veranstaltungen fördern. Das Gleiche gilt für Veranstaltungen und Teilnehmer. Gemäß den Normalisierungsregeln für Datenbanken sind in solchen Fällen Kreuztabellen das Mittel der Wahl.

Eine grafische Darstellung des Datenbankmodells sieht dann etwa so aus:

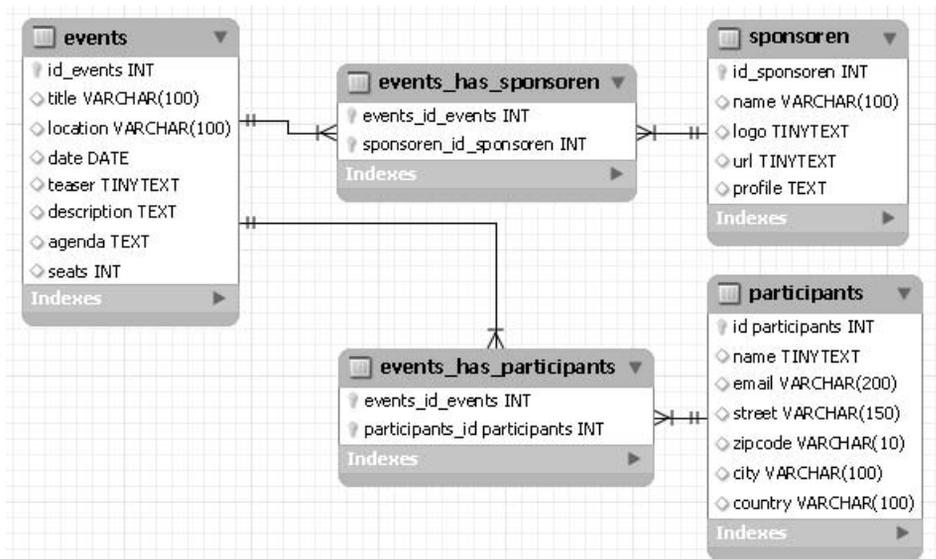


Bild 2.1: Der Entwurf für das Datenbankmodell

## 2.2 Umsetzung des Entwurfs in TYPO3

Nachdem die Datenstruktur feststeht, soll nun die Anpassung an die TYPO3-Umgebung erfolgen. Das beginnt mit der Suche nach dem richtigen Namen.

### 2.2.1 Der Extension Key

Erweiterungen werden von TYPO3 nicht zuletzt anhand des Namens verwaltet. Daher muss jede TYPO3-Extension eine eindeutige Bezeichnung erhalten, den *Extension Key*. Bei der Benennung von Extensions sind daher einige Regeln einzuhalten.

Für Extensions, die nur für eine Installation verwendet werden sollen, muss die Eindeutigkeit des Namens nur innerhalb des Systems hergestellt sein. Wenn Sie sicher sind, dass eine Erweiterung nur für ein bestimmtes Projekt von Interesse ist und nicht allgemein verfügbar sein soll, nehmen Sie daher einfach einen Namen Ihrer Wahl und setzen das Präfix `user_` davor. Damit ist klar, dass diese Extension nicht aus dem öffentlichen Repository für Erweiterungen stammen kann. Vom Extension Key abgeleitete Bezeichner für Tabellen usw. ergeben sich dann wie in der folgenden Tabelle angegeben:

	<i>Regel</i>	<i>Beispiel</i>
Extension Key	Frei wählbar, aber mit Präfix <code>user_</code>	<code>user_myextension</code>
Datenbank-Tabellen und -Felder	mit dem Präfix der Erweiterung	<code>user_myextension_images</code> <code>user_myextension_categories</code>
Backend-Modul	enthält keine Unterstriche, beginnt mit <code>u</code>	<code>umyextension</code>
PHP-Klassen	Klassennamen werden gebildet wie Datenbank-Tabellen und Felder, die Dateien dazu erhalten den Vorsatz <code>class.</code>	<code>class.myextension_pi1.php</code>

**Tabelle 2.4:** User Extension Keys und abgeleitete Bezeichner

Soll eine Erweiterung dagegen der TYPO3-Gemeinde zur Verfügung gestellt, sprich ins TYPO3 Extension Repository (TER) geladen werden, muss der Extension Key weltweit eindeutig sein.

Um dies sicherzustellen, werden Extension Keys im TER registriert. Dabei werden neben der Eindeutigkeit auch noch einige formale Kriterien für den Aufbau des Keys geprüft.

Diese sind:

- Der Key darf nur aus Buchstaben, Ziffern und dem Unterstrich (`_`) bestehen.
- Der Key darf nicht mit einem reservierten Präfix wie `tx_`, `u_`, `tt_` oder `sys_` beginnen – auch wenn im Repository aus historischen Gründen noch Extensions wie `tt_news` oder `tt_address` zu finden sind.
- Da aus dem Key Bezeichner für Module abgeleitet werden, indem die Unterstriche entfernt werden, muss der Key auch in diesem Fall eindeutig bleiben. Unter anderem deswegen wird für neue Extensions empfohlen, von vornherein keine Unterstriche im Extension Key zu verwenden.
- Da aus dem Extension Key auch der Verzeichnisname in TYPO3 abgeleitet wird, sollten keine Großbuchstaben verwendet werden. Dies würde je nach Betriebssystem zu Komplikationen führen.
- Damit ergeben sich folgende Bezeichner in der Extension:

	<i>Regel</i>	<i>Beispiel</i>
Extension Key	Vom TER zugewiesen bzw. akzeptiert	<code>myextension</code>
Datenbank-Tabellen und -Felder	Name der Erweiterung, versehen mit dem Präfix <code>tx_</code>	<code>tx_myextension_images</code> <code>tx_myextension_categories</code>
Backend-Modul	enthält keine Unterstriche, beginnt mit <code>tx</code>	<code>txmyextension</code>
PHP-Klassen	Klassennamen werden gebildet wie Datenbank-Tabellen und Felder, die Dateien dazu erhalten den Vorsatz <code>class.</code>	<code>class.tx_myextension_pi1.php</code>

**Tabelle 2.5:** TER Extension Keys und abgeleitete Bezeichner

Neben den formalen Kriterien gibt es auch einige »weiche« Anforderungen. So soll der Key den Zweck der Extension beschreiben, gleichzeitig aber nicht zu lang sein. Im Repository finden Sie viele Extensions, die mit einem Kürzel für den Autor beginnen. Das wird nicht empfohlen, lässt sich aber manchmal nicht vermeiden, wenn man eine Erweiterung in einer Kategorie schreibt, in der es schon einige andere gibt. Besser ist es aber, wenn Ihre Extension eine Eigenart aufweist, die im Extension Key benutzt werden kann.

Für unseren Event Manager wären also gute Extension Keys:

- eventmanager
- eventmgr
- eventmanagement

Nicht zu empfehlen wären dagegen:

- emgr (zu kurz)
- my\_manager (keine Beschreibung, Unterstrich)
- event\_and\_sponsor\_management (lang, viele Unterstriche)

## 2.2.2 Veröffentlichen von Extensions

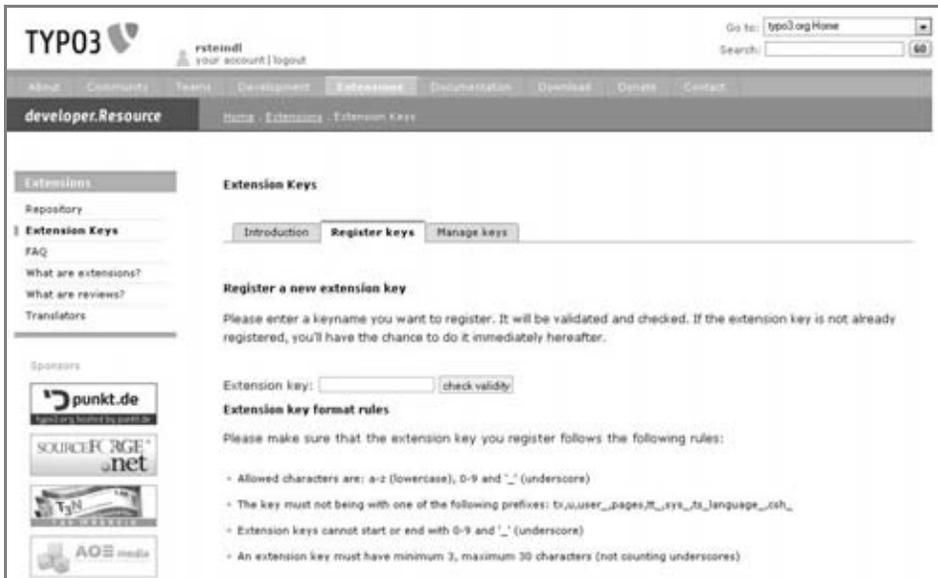
Selbst wenn beim Schreiben einer Erweiterung zunächst nicht geplant ist, die Extension im TER zu veröffentlichen, sollten Sie diesen Schritt auf jeden Fall aus zwei Gründen mit einplanen:

Zum einen ist es mit viel Aufwand verbunden, den Namen einer Extension nachträglich zu ändern. Der Extension Key wird für Verzeichnis- und Dateinamen ebenso benutzt wie für Datenbanktabellen. Und selbst wenn die Erweiterung zu speziell ist, um für andere von Nutzen zu sein: Sie können sie als privat oder als Test markieren und sie so im TER ablegen, ohne den Zugriff für andere freizuschalten.

Zum anderen lebt TYPO3 nicht zuletzt von der Vielzahl an Erweiterungen. Wenn eine Extension also von allgemeinem Interesse sein könnte, sollte sie auch im TER vorhanden sein.

Alle Erweiterungen, die Sie schreiben, unterliegen wie TYPO3 selbst automatisch der GPL (GNU General Public License). Das zwingt Sie zwar nicht zur Veröffentlichung, doch sollten Sie sich im Klaren sein, dass der Quellcode Ihrer Extension von jedem benutzt und weiterbearbeitet werden darf, der Zugriff darauf hat.

Um eine Extension im Repository veröffentlichen zu können, benötigen Sie einen Login auf der Website *TYPO3.org*. Hier können Sie ein persönliches Profil hinterlegen, sind aber nicht dazu gezwungen. Nach der Anmeldung können Sie auf der Seite <https://TYPO3.org/extensions/extension-keys/> einen Extension Key registrieren.



**Bild 2.2:** Das Registrieren eines Extension Keys geht schnell und einfach

Das Repository prüft, ob der Key die Vorgaben erfüllt und ob er frei ist. Falls ja, wird er eingetragen, andernfalls erhalten Sie eine Fehlermeldung.

**Tipp:** Wenn Sie einen Extension Key reservieren wollen, recherchieren Sie vorher im Repository, welche Namen bereits benutzt werden. Allerdings kann es passieren, dass ein Extension Key reserviert ist, obwohl keine Extension dazu im Repository angeboten wird.

Einmal reservierte Extension Keys können Sie über die oben genannte Website im Abschnitt »Manage keys« auch verwalten. Sie sehen dort, wann der letzte Upload der Extension erfolgt ist. Auch eine Löschfunktion für Ihre Extension Keys ist hier zu finden. Allerdings lässt sich ein Key nur dann komplett löschen, wenn er noch nie benutzt wurde, d. h. wenn noch keine Extension mit diesem Key ins Repository eingestellt wurde.

Wenn Sie eine Extension nicht mehr weiter verwalten wollen, aber ein anderer User sich dazu bereit erklärt hat, können Sie die Rechte an einem Extension Key auch einem anderen User übertragen.

Von den oben genannten brauchbaren Keys waren bereits einige vergeben, für die hier vorgestellte Erweiterung wurde daher der Key `eventmanagement` reserviert. Nach den Empfehlungen von TYPO3 ist er mit 15 Zeichen etwas zu lang, doch abgesehen davon erfüllt er alle Anforderungen.

### 2.2.3 Anpassung der Datenbankstruktur

Wir wissen bereits, dass unsere Extension eine Datenbank benötigt, und haben die gewünschte Struktur bereits bestimmt. Da aber diese Struktur in die bereits bestehende von TYPO3 integriert werden soll, müssen einige Anforderungen erfüllt werden.

#### TYPO3-eigene Felder

Jede TYPO3-Tabelle hat einige Felder, die für die Verwaltung durch TYPO3 wichtig sind. Zwingend sind hierbei zunächst nur `uid` und `pid`.

`uid` (Unique ID) ist die eindeutige Kennzeichnung jedes Eintrags. Dabei handelt es sich um ein sogenanntes Autoincrement-Feld, das automatisch hochzählt.

`pid` (Page ID) bestimmt, auf welcher Seite des TYPO3-Baums der Inhalt platziert wird. Eine 0 steht für den Beginn des Baums, den `root level`.

Wie wir später sehen werden, lassen sich beim Anlegen der Tabelle im Kickstarter weitere Optionen aktivieren, die in folgenden Feldern verwaltet werden:

<i>Feld</i>	<i>Typ</i>	<i>Beschreibung</i>
<code>tstamp</code>	Integer	Datum und Uhrzeit der letzten Änderung des Datensatzes als Unix-Timestamp.
<code>crdate</code>	Integer	Datum und Uhrzeit der Erstellung des Datensatzes als Unix-Timestamp.
<code>cruser_id</code>	Integer	Die ID des Backend-Users (entspricht dem Feld <code>uid</code> in der Tabelle <code>beuser</code> ), der den Eintrag erstellt hat. Wurde ein Eintrag über das Frontend angelegt, steht hier meist eine 0.
<code>sys_language_uid</code>	Integer	Die <code>uid</code> der Sprache, in der der Eintrag angelegt wurde. Dieses Feld ist zusammen mit <code>l10_parent</code> und <code>l10_diffsource</code> wichtig für die Lokalisierung von Datensätzen.

Feld	Typ	Beschreibung
l10n_parent	Integer	Die uid des Datensatzes, der durch diesen Eintrag lokalisiert wird.
l10n_diffsource	Medium Text	Die Originaldaten, die die Grundlage der Lokalisierung bilden.
deleted	Tiny Integer	Flag, das zeigt, ob der Datensatz gelöscht ist.
hidden	Tiny Integer	Flag, das zeigt, ob der Datensatz auf »verborgen« gesetzt ist.
starttime	Integer	Zeitpunkt, ab dem der Datensatz online gezeigt wird, als Unix-Timestamp.
endtime	Integer	Zeitpunkt, ab dem der Datensatz nicht mehr gezeigt wird, als Unix-Timestamp.
fe_group	Integer	uid bzw. Liste von uids der Frontend-Usergruppen, die auf diesen Eintrag zugreifen dürfen.

**Tabelle 2.6:** Zusätzliche Felder in TYPO3

**Tipp:** In einigen (älteren) Extensions finden Sie statt der Felder `l10n_parent` und `l10n_diffsource` die Felder `l18n_parent` und `l18n_diffsource`. Das hat historische Gründe – und ist eigentlich ein Bug. Denn `l10n` ist die übliche Abkürzung für das Wort *Localisation* (weil zwischen »l« und »n« 10 weitere Buchstaben kommen). Das Gegenstück dazu ist *Internationalisation*, was abgekürzt `i18n` ergibt.

Der Prozess der Übersetzung, der mithilfe der Datenbankfelder realisiert wird, gehört allgemein zur Lokalisierung, darum ist `l10n...` die korrekte Bezeichnung. Als Internationalisierung wird eher die Vorbereitung und grundlegende Struktur bezeichnet, auf der die Lokalisierung aufsetzt.

Wenn eine Extension das Workspace-Konzept bzw. die Versionierung von Datensätzen unterstützt, kommen das Feld `t3_origuid` sowie eine Reihe weiterer Felder hinzu, die alle mit `t3ver_` beginnen.

Die Felder unserer Beispieldatenbank kommen mit den genannten Feldern nicht in einen Namenskonflikt. Wäre das der Fall, sollten Sie die Benennung der Felder entsprechend ändern. Theoretisch lassen sich auch die Namen der TYPO3-Felder ändern, doch das erfordert zum einen Handarbeit und erschwert zum anderen das Verständnis der Extension.

### Datenbankrelationen in TYPO3

TYPO3 kennt unterschiedliche Methoden, um Beziehungen zwischen zwei Tabellen herzustellen.

Die einfachste Methode (und die erste, die in TYPO3 vorhanden war) besteht darin, in einem Feld einer Tabelle eine kommaseparierte Liste von `uid`-Werten der zweiten Tabelle zu speichern. In manchen Fällen wird statt der `uid` als Zahl ein String aus Tabellename und `uid` in der Form `tabelle_uid` gespeichert. Dies sorgt dafür, dass wirklich jedes Inhaltselement ansprechbar wird.

Der Vorteil dieser Methode ist die relativ einfache Verständlichkeit und die Verarbeitungsgeschwindigkeit.

Aus Sicht der Datenbanktheorie ist eine solche Kopplung allerdings für die m-zu-n-Relationen, die wir in unserer Extension verwenden wollen, nicht korrekt. Stattdessen fordern die Normalisierungsregeln in solchen Fällen eine Kreuztabelle. Eine solche Tabelle nimmt Paarungen von ID-Werten auf.

Das folgende Beispiel zeigt den prinzipiellen Aufbau einer Datenbankstruktur mit Kreuztabelle:

Unsere Event-Tabelle soll folgende Einträge haben:

<i>uid</i>	<i>name</i>
1	TYPO3-Seminar
2	PHP-Kurs
3	Datenbankentwicklung mit MySQL

**Tabelle 2.7:** Beispieldaten Events

Die Liste der Sponsoren sieht so aus:

<i>uid</i>	<i>name</i>
1	IBM
2	MySQL
3	Oracle
4	Franzis-Verlag

**Tabelle 2.8:** Beispieldaten Sponsoren

Wenn nun (nur als Beispiel) IBM das PHP-Seminar sponsort, MySQL das Datenbank-Seminar und der Franzis-Verlag bei TYPO3 und PHP als Sponsor auftritt, sieht die Kreuztabelle wie folgt aus:

<i>event_uid</i>	<i>sponsor_uid</i>
2	1
2	4
3	2
1	4

**Tabelle 2.9:** Beispiel Kreuztabelle

Eine solche Konstruktion ist auf den ersten Blick komplexer und erfordert tatsächlich auch einen sorgfältigeren Umgang mit den Daten. Im Gegenzug erhält man damit aber eine robustere Struktur mit weniger Redundanzen. Und für Datenbank-Abfragen kann sozusagen jede der beiden Spalten die Führung übernehmen, d. h. wenn man die Sponsoren eines Events wissen will, sucht man alle Einträge mit einer bestimmten `event_uid`. Für die Frage, welche Events ein Sponsor bezuschusst, wird die `sponsor_uid` festgelegt.

In TYPO3 ist dieses Konzept ebenfalls implementiert und kann im Kickstarter beim Anlegen einer Relation aktiviert werden. Die Kreuztabelle wird dann automatisch angelegt. Die Besonderheiten der M-M-Relation (Many-to-Many) in TYPO3 werden wir beim Aufbau der Extension genauer behandeln.

TYPO3 kennt auch noch eine dritte Methode, Beziehungen zwischen zwei Datenbank-Tabellen herzustellen – sie lautet kurz IRRE. Die Abkürzung steht für Inline Relational Record Editing und wurde mit der Absicht eingeführt, das Editieren verknüpfter Datensätze zu erleichtern. Wenn beispielsweise eine Person mehrere Adressen hat, lässt sich mit IRRE ein Backend-Formular gestalten, das sowohl die Personendaten als auch die Adressen bearbeitbar macht, ohne auf neue Seiten zu wechseln. Das Anlegen und Bearbeiten neuer Adressen wird dabei mittels AJAX gesteuert. Für IRRE gibt es im TYPO3 Extension Repository das *Tutorial for Inline Relational Record Editing IRRE* (*irre\_tutorial*), das mit Beispieldaten das Konzept näher erläutert.

Für unseren Event-Manager werden wir die Event- und die Sponsoren-Tabelle mit einer Kreuztabelle realisieren, die Kopplung der Teilnehmer zu den Events dagegen mit der klassischen Methode, um beides vorzustellen.

## Verwendung vorhandener Tabellen

Die Anpassung der Datenbankstruktur an TYPO3 erfordert aber noch eine weitere Überlegung: Die geplante Extension soll die Adressen der Teilnehmer speichern, wofür in unserem Entwurf eine entsprechende Tabelle definiert wurde. Allerdings gibt es in TYPO3 bereits Möglichkeiten, Teilnehmer bzw. Adressen zu speichern – zum einen die Extension `tt_address`, zum anderen die Tabelle für die Frontend-User (`fe_users`).

**Tipp:** Lassen sich Funktionen einer Erweiterung unter Zuhilfenahme der vorhandenen Tabellen durchführen, so ist das meist der bessere Weg, weil sich ähnliche Daten nicht verteilt über das ganze System finden. Im schlimmsten Fall könnte ein angemeldeter User über `fe_users` verwaltet werden, seine Adresse in `tt_address` zu finden sein und seine Reservierungen in einer dritten Tabelle.

`tt_address` wird von einigen weiteren Extensions unterstützt, sodass man damit beispielsweise eine gemeinsame Adresdatenbank für Mailings und Events aufbauen kann. Da sich Adressen in hierarchisch gegliederte Gruppen einteilen lassen, ist eine saubere Trennung nach Funktion problemlos möglich.

Die Teilnehmer als Frontend-User zu registrieren, ist vor allem dann sinnvoll, wenn mit der Registrierung auch Funktionen der Website freigeschaltet werden. So könnte ein Teilnehmer Zugang zu den Präsentationen einer Veranstaltung erhalten oder sich bei der Buchung weiterer Seminare mit seinem Login anmelden und sich so die erneute Eingabe aller Daten ersparen.

Allerdings ist die Struktur der `fe_users`-Tabelle für unsere Zwecke nicht optimal, vor allem für deutschsprachige Webseiten. Es fehlen Felder für die Anrede und akademische Titel, Name und Adresse sind nicht aufgeteilt in Vor- und Nachname bzw. Straße und Hausnummer. Außerdem wollen wir speichern, zu welchen Veranstaltungen sich jemand angemeldet hat.

Trotzdem lässt sich `fe_users` für unsere Zwecke verwenden. In TYPO3 ist es kein Problem für eine Extension, bestehende Tabellen zu erweitern. Unser Event Manager wird daher einfach die `fe_users`-Tabelle um die nötigen Felder ergänzen.



## 3 Struktur einer Extension

Nachdem die Planung abgeschlossen ist und die grundsätzliche Datenstruktur feststeht, müssen wir im nächsten Schritt festlegen, wie die gewünschten Funktionen der Erweiterung in TYPO3 realisiert werden können. Um diese Entscheidung zu treffen, sollen hier die verschiedenen möglichen Bestandteile einer Extension vorgestellt werden.

### 3.1 Kategorien von Extensions

Im Extension Manager von TYPO3 finden wir 10 verschiedene Kategorien:

- Backend
- Backend Modules
- Frontend
- Frontend Plugins
- Services
- Documentation
- Examples
- Templates
- Miscellaneous
- Kategorie []

<b>Backend</b>							
	Database Abstraction Layer	<i>dbal</i>	0.9.20			System	Beta
	Editor with syntax highlighting	<i>t3editor</i>	0.0.10			System	Alpha
	Extension Kickstarter	<i>kickstarter</i>	0.4.0			Local	Stable
	htmlArea RTE	<i>rtehtmlarea</i>	1.7.12			System	Stable
	T3dev	<i>t3dev</i>	1.1.0			Local	Beta
	Versioning Management	<i>version</i>	1.2.0			System	Beta
<b>Backend Modules</b>							
	Extension Development Evaluator	<i>extdeveval</i>	3.1.0			Local	Stable
	User>Task Center, Actions	<i>sys_action</i>	1.2.0			System	Stable
<b>Frontend</b>							
	CSS styled content	<i>css_styled_content</i>	0.4.0			System	Beta
<b>Frontend Plugins</b>							
	CSV Table	<i>csv_table</i>	0.1.0			Local	Beta
	Event Manager	<i>eventmanagement</i>	0.0.0			Local	Alpha
	Frank Naegler libs	<i>fn_lib</i>	0.2.0			Local	Alpha
	Frontend Login for Website Users	<i>felogin</i>	1.0.0			System	Beta
	Indexed Search Engine	<i>indexed_search</i>	2.11.1			System	Stable
	MailformPlus	<i>th_mailformplus</i>	4.0.15			Local	Beta
	News	<i>tt_news</i>	2.5.2			Local	Beta
	Unternehmensliste	<i>hyline_companylist</i>	0.0.0			Local	Beta
<b>Miscellaneous</b>							
	ADODB	<i>adodb</i>	4.94.0			System	Stable
<b>Documentation</b>							
	TSconfig	<i>doc_core_tsconfig</i>	4.2.0			Local	Stable
	TSref	<i>doc_core_tsref</i>	4.2.1			Local	Stable
	TYPO3 Coding Guidelines	<i>doc_core_cgl</i>	4.3.2			Local	Stable
	TYPO3 Core API	<i>doc_core_api</i>	4.2.0			Local	Stable
	TypoScript Syntax and In-depth Study	<i>doc_core_ts</i>	4.2.0			Local	Stable
	Upgrade-Tutorial Typo3 3.8.0 to 4.2.0	<i>doc_tut_upgrade_380_to_420</i>	1.0.0			Local	Stable
<b>//</b>							
	TYPO3 skin	<i>t3skin</i>	4.2.0			System	Stable

Bild 3.1: Die Kategorien im Extension Manager

### 3.1.1 Backend

Die Kategorie Backend umfasst Erweiterungen, die die Funktion des TYPO3-Backends erweitern, ohne einen eigenen Eintrag in der Modulliste zu benötigen. Beispiele hierfür sind der Rich Text Editor *rtehtmlarea* oder *cms*, das eigentliche Content Management von TYPO3.

### 3.1.2 Backend Modules

In der Kategorie `Backend Modules` finden sich Module, also Extensions, die die Backend-Funktionalität von TYPO3 erweitern. Sie sind typischerweise links im Modulbaum mit Einträgen gelistet. Beispiele hierfür sind die Grundmodule von TYPO3 oder Erweiterungen wie Full Backup (`w4x_backup`).

### 3.1.3 Frontend

Unter `Frontend` finden sich Erweiterungen, die Funktionalität für das Frontend bereitstellen, aber nicht notwendigerweise selbst HTML-Ausgaben erzeugen. Vertreter dieser Kategorie sind etwa `css_styled_content` oder `RealUrl`.

### 3.1.4 Frontend Plugins

`Frontend Plugins` erzeugen Ausgaben auf der Website. Sie stellen neue Inhaltsobjekte zur Verfügung oder erweitern bestehende. Beispiele hierfür sind Erweiterungen wie `tt_news` oder `fe_login`. Dieser Erweiterungstyp ist vermutlich derjenige, der am häufigsten für selbstgeschriebene Extensions benutzt wird.

### 3.1.5 Services

`Services` sind spezielle Extensions, die sozusagen Dienstleistungen für andere Erweiterungen bereitstellen, beispielsweise die Authentifizierung von Usern. In einem TYPO3-System kann eine Service-Funktion mehrfach angeboten werden, etwa einmal durch Aufruf eines externen Programms und einmal durch ein PHP-Skript. Welche Funktion genutzt wird, kann beispielsweise vom Betriebssystem abhängen.

**Tipp:** Services sollten nicht mit Frontend- oder Backend-Funktionen in eine Extension gepackt werden. Lediglich für die Funktion notwendige Tabellen oder Tabellenfelder werden Bestandteil der Service-Erweiterung.

### 3.1.6 Documentation

In dieser Kategorie befinden sich Extensions, die lediglich Dokumentationen bereitstellen, meist in Form von OpenOffice-Dokumenten. Selbstgeschriebene Extensions sollten ihre Dokumentation innerhalb der Extension bereitstellen. Eine eigene Dokumentationserweiterung hat nur Sinn, wenn es sich beispielsweise um grundlegende Dinge (wie die TYPO3-Dokumentation) oder um eine übergreifende Dokumentation handelt, die

das Zusammenspiel mehrerer Extensions beschreibt. Typischerweise beginnen Extensions in dieser Kategorie mit `doc_`, doch zwingend vorgeschrieben ist das nicht.

### 3.1.7 Examples

Beispiele für Funktionalitäten von Erweiterungen oder TYPO3 selbst, etwa als Teil von Tutorials, werden in diese Kategorie einsortiert.

### 3.1.8 Templates

Hier finden Sie fertige TYPO3-Templates, entweder als HTML/CSS-Struktur für den Autoparser oder TemplaVoilà oder als Typoscript-Templates. Auch wenn es relativ einfach ist, HTML-Templates für TYPO3 anzupassen, gehen die ersten Schritte mit fertigen Templates leichter. Eine Liste mit Links zu Anbietern kostenfreier TYPO3-Templates finden Sie beispielsweise im TYPO3-Wiki<sup>1</sup>.

### 3.1.9 Miscellaneous

Der Abschnitt `Miscellaneous` dient als Sammelbecken für alles, was sonst nicht recht passt, also etwa Bibliotheken, die von anderen Erweiterungen benutzt werden, oder statische Tabellen. Auch Skins sind häufig hier zu finden.

**Tipp:** Wenn Ihre Extension nicht exakt in eine der normalen Kategorien passt, etwa weil sie Front- und Backend-Funktionen enthält, heißt dies nicht, dass sie in `Miscellaneous` landen sollte. Orientieren Sie sich besser an der Hauptfunktion Ihrer Erweiterung und wählen Sie eine der Standard-Kategorien.

### 3.1.10 Kategorie []

Die Kategorie `[]` ist eigentlich keine Kategorie, sondern enthält die Erweiterungen, bei denen das `category`-Feld leer ist. In TYPO3 4.2 findet sich hier beispielsweise der Skin für das Backend. Für eigene Erweiterungen sollten Sie auf jeden Fall eine Kategorie vergeben.

---

<sup>1</sup> [http://wiki.TYPO3.org/index.php/TYPO3\\_templates\\_repository](http://wiki.TYPO3.org/index.php/TYPO3_templates_repository)

# Extensions für TYPO3

*Extensions für TYPO3 gibt es viele – aber sind darunter auch wirklich die, die Sie für Ihre Website brauchen? Am Beispiel einer konkreten Anwendung demonstriert der Webexperte Robert Steindl in diesem Buch, wie Sie selbst Erweiterungen für TYPO3 programmieren. Der Autor zeigt, wie Sie die Funktionen der TYPO3-API richtig verwenden und wie Sie das Kickstarter-Werkzeug einsetzen, mit dem Sie den Kern der Erweiterung erzeugen. Darüber hinaus führt er in die Frontend- und Backend-Programmierung von TYPO3 ein. Ein Ausblick widmet sich zudem der Zukunft von TYPO3 und den Änderungen, die das neue Framework FLOW3 für Entwickler bringen wird.*

## ► Eine eigene Beispiel-Extension entwickeln

Anhand einer Event-Verwaltung als Beispielanwendung sehen Sie konkret, wie Erweiterungen für TYPO3 entwickelt werden. Schritt für Schritt erläutert Robert Steindl die Definition der Anforderungen, die Generierung des Codegerüsts im Extension Kickstarter sowie die Frontend- und Backend-Programmierung. Gemeinsam mit dem Autor legen Sie die Datenbankstruktur fest: Sie erfahren, wie Sie vorhandene Datenbanktabellen in TYPO3 anpassen und die Erweiterung strukturieren.

## ► Die TYPO3-API

Das Verstehen der komplexen Programmierschnittstelle von TYPO3 ist für alle Extension-Entwickler der entscheidende Erfolgsfaktor. Robert Steindl stellt die zentralen Funktionen und Variablen für das Front- und Backend von TYPO3 vor und zeigt, wie Sie diese Komponenten richtig einsetzen. Breiten Raum nimmt der Table Configuration Array (TCA) ein, der alle Einstellungen der Datenbanktabellen von TYPO3 enthält und die Beziehungen zwischen ihnen beschreibt. Zudem erfahren Sie, wie Sie Ajax im Backend einsetzen, das Erscheinungsbild Ihrer Extension bestimmen und Wizards konfigurieren.

## ► Der richtige Codierstil

Spätestens dann, wenn Sie für einen breiten Anwenderkreis entwickeln, müssen Sie die Coding-Guidelines für TYPO3 einhalten. Robert Steindl erläutert die wichtigsten Regeln für die Benennung von Dateinamen, Klassen, Funktionen und Datenbankbefehlen und erklärt, wie Sie Ihre Erweiterungen korrekt dokumentieren. Darüber hinaus zeigt er, wie Sie das Werkzeug **Extdeveval** einsetzen, um Ihren Code von Flüchtigkeitsfehlern und Unsauberkeiten zu befreien. Last, but not least sehen Sie, wie Sie Ihre Erweiterung über das TYPO3 Extension Repository (TER) der weltweiten User-Community zugänglich machen.

## Aus dem Inhalt:

- Eine Erweiterung planen
- Der Extension Key
- Datenbankrelationen in TYPO3
- Extension-Kategorien
- Der Extension Kickstarter
- Das Codegerüst der Erweiterung erzeugen
- Die TYPO3-Programmierschnittstelle
- Das Frontend-Plug-in entwickeln
- Konfiguration des Plug-ins
- Unterstützung von Frontend-Templates
- Backend-Programmierung
- Die Extension fertigstellen
- Bereinigung des Codes
- Coding-Richtlinien
- Lokalisierung der Erweiterung
- Dokumentationstipps
- Ausblick auf FLOW3 und TYPO3 5.0

## Über den Autor:

Robert Steindl ist Diplom-Physiker und Geschäftsführer des Web-Beratungsunternehmens Agilo GmbH. Gleichzeitig ist er Manager des Online-Services des Consulting-Hauses Experton Group. Zuvor war er als leitender Redakteur für den Onlineauftritt der Zeitschrift PC WELT verantwortlich. Robert Steindl lebt in München.



## Auf [www.buch.cd](http://www.buch.cd)

- Die komplette Beispiel-Extension
- Alle Beispieldateien

Der Franzis-Blog für Webentwickler  
[webdeveloper.franzis.de](http://webdeveloper.franzis.de)



30,- EUR [D]

ISBN 978-3-645-60000-2

Besuchen Sie unsere Website

[www.franzis.de](http://www.franzis.de)