

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/3.0/de/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

Andrea Herzog-Kienast ▪ Franz Holzinger

Der TYPO3-Webshop

Installation – Produktangebot – Zahlungsabwicklung

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/3.0/de/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grunde sind die in dem vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor(en), Herausgeber, Übersetzer und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht, auch nicht für die Verletzung von Patentrechten, die daraus resultieren können. Ebenso wenig übernehmen Autor(en) und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. werden ohne Gewährleistung der freien Verwendbarkeit benutzt und können auch ohne besondere Kennzeichnung eingetragene Marken oder Warenzeichen sein und als solche den gesetzlichen Bestimmungen unterliegen.

Dieses Werk ist urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches – oder Teilen daraus – vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlags in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Bibliografische Information Der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Das vorliegende Werk steht unter der Lizenz:

Creative Commons, Namensnennung – Weitergabe unter gleichen Bedingungen 3.0 Deutschland

<http://creativecommons.org/licenses/by-sa/3.0/de/>

Diese Ausgabe ist textidentisch mit der Originalausgabe:

© 2007 Open Source Press, München

Gesamtlektorat: Dr. Markus Wirtz

Satz: Open Source Press (L^AT_EX)

Umschlaggestaltung: www.fritzdesign.de

Gesamtherstellung: Kösel, Krugzell

ISBN 978-3-937514-43-7

<http://www.opensourcepress.de>

Vorwort von Kasper Skårhøj

Als sich abzeichnete, dass TYPO3 eine große Gemeinschaft von Anwendern anspricht, wurde mir klar, dass dieses CMS eine Shop-Extension benötigt, denn neben der Darstellung von Information ist eben auch der Vertrieb von Waren und Dienstleistungen eine Selbstverständlichkeit im Netz. Also begann ich um das Jahr 2000 mit der Entwicklung von tt_products.

Die Anforderungen an TYPO3 wuchsen rasant, und ich musste mich darauf konzentrieren, das Innenleben des Systems innovativ und flexibel zu halten und voranzubringen. Der Grundstein war gelegt, aber der eigentliche Hausbau hatte gerade erst begonnen. Die weitere Entwicklung des Shopsystems mussten darum andere übernehmen, und ich war sehr glücklich, dass sich ebenso interessierte wie fähige Entwickler für diese Aufgabe fanden, an der seitdem kontinuierlich weitergearbeitet wird.

Seit den Anfängen von tt_products, das zunächst die schnelle Reaktion eines Entwicklers auf konkrete praktische Bedürfnisse an das neue CMS war, haben sich die Bedingungen grundlegend verändert. Mittlerweile haben verschiedene Gruppen in unserer großen Community Shopsysteme für unterschiedliche Belange geschaffen. Indem sie die Bedürfnisse von Unternehmen und deren Marktbedingungen beobachteten und darauf reagierten, haben sie TYPO3 auch im Bereich des Online-Handels beständig vorangebracht. Der Code ist nicht mehr nur eine schnelle Lösung, sondern das Ergebnis umfangreicher Analyse mit dem Ziel, sich auch langfristig an sich verändernde Anforderungen anpassen zu können. Diese neue, professionelle Herangehensweise lässt mich darum auch im Bereich der Shopsysteme sehr optimistisch in die Zukunft von TYPO3 sehen.

Doch es genügt nicht, TYPO3 in technischer Hinsicht immer weiter zu verbessern und immer mächtigere Programme zur Verfügung zu stellen – wir müssen sie auch bekannt machen! Die Welt muss wissen, was sie mit TYPO3 und z. B. einem Shopsystem erreichen kann, und dazu benötigen die Menschen verständliche und umfangreiche Dokumentation. Was nützt der beste Code, die beste Extension, wenn niemand – außer einem Spezialisten – in der Lage ist, die Software auch zu installieren und anzupassen? Hinzu kommt, dass diese Spezialisten rar und meist ohnehin überlastet sind; die Nachfrage steigt, die Spezialisten sind ausgebucht, und so bleibt gerade

für kleinere Projekte selten Zeit. Welcher Kunde aber ist begeistert, wenn er Monate auf die Installation seines Webshops warten muss?

In der Vergangenheit war es üblich, dass nur ein „eingeweihter“ Personenkreis TYPO3 und seine Extensions installiert, doch die Zeiten ändern sich: Immer mehr Menschen interessieren sich für unser Projekt, und so wird es immer mehr „Poweruser“ geben, die sich komplexen Aufgaben der Installation, Administration und Implementierung stellen und so auch die Experten bei der Entwicklung entlasten. Präziser Dokumentation kommt darum eine immer wichtigere Bedeutung zu, sei es als technisches Handbuch oder als Entscheidungshilfe. Wir müssen aufpassen, nicht arrogant zu werden, und dürfen all unser Wissen und unsere Erfahrung nicht für uns behalten. Wir alle haben irgendwelche Gaben, und wir sollten sie mit der Welt teilen.

Mit diesem ersten Buch zum Thema TYPO3-Webshops, das mit tt_products eine Extension detailliert beschreibt, aber auch einige alternative Lösungen vorstellt, ist ein wichtiger Schritt in die richtige Richtung gemacht. Teilen wir unser Wissen!

- kasper

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/4.0/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de/>)

Inhaltsverzeichnis

| | |
|--|-----------|
| Vorwort von Kasper Skårhøj | 5 |
| Vorwort der Autoren | 13 |
| 1 Grundlagen der Installation in ein bestehendes TYPO3-System | 17 |
| 1.1 Das Shopsystem tt_products | 18 |
| 1.2 Benötigte Extensions | 19 |
| 1.3 TS-Templates | 23 |
| 1.4 Der Produkte-SysOrdner | 24 |
| 1.5 Das tt_products-Plugin | 25 |
| 1.5.1 Plugin einfügen | 26 |
| 1.5.2 Shopseiten erstellen | 27 |
| 2 Grundkonfiguration | 31 |
| 2.1 TS-Template | 31 |
| 2.1.1 Constants | 32 |
| 2.1.2 Setup | 35 |
| 2.2 HTML-Template | 38 |
| 2.2.1 Adresdaten | 41 |
| 2.2.2 AGB | 42 |
| 2.2.3 Bestelldaten zusammenfassen | 44 |
| 2.2.4 Bestellung abschließen | 45 |
| 2.2.5 E-Mail-Benachrichtigung | 46 |
| 3 Benutzerregistrierung | 49 |
| 3.1 New Login Box | 50 |
| 3.1.1 Installation über Marker | 50 |

| | | |
|----------|---|-----------|
| 3.1.2 | Installation als Plugin | 53 |
| 3.2 | Frontendbenutzerregistrierung | 54 |
| 3.2.1 | SysOrdner und Benutzergruppen | 55 |
| 3.2.2 | Seiten erstellen und Plugins einfügen | 56 |
| 3.2.3 | TS-Template zur Benutzerregistrierung | 57 |
| 3.2.4 | HTML-Template | 62 |
| 3.2.5 | Verschlüsselung | 63 |
| 4 | Produktpflege | 65 |
| 4.1 | Registerkarte Allgemein | 66 |
| 4.2 | Registerkarte Varianten | 79 |
| 4.3 | Registerkarte Beziehungen | 83 |
| 4.3.1 | Template Produkte | 84 |
| 4.3.2 | Verwandte Produkte | 84 |
| 4.4 | Registerkarte Versand | 84 |
| 4.5 | Mehrsprachigkeit | 85 |
| 4.5.1 | TS-Template | 86 |
| 4.5.2 | HTML-Template | 87 |
| 5 | Shopseiten | 95 |
| 5.1 | Shopseiten und Plugin-Objekte | 95 |
| 5.1.1 | Listenansicht | 95 |
| 5.1.2 | Einzelansicht | 96 |
| 5.1.3 | Warenkorb | 97 |
| 5.1.4 | Kontrolle und Bezahlung | 97 |
| 5.1.5 | Bestellung abschließen | 98 |
| 5.1.6 | AGB | 98 |
| 5.1.7 | Tracking | 99 |
| 5.1.8 | Gutscheine | 103 |
| 5.1.9 | Freundschaftswerbung | 107 |
| 5.1.10 | Liste Highlights | 108 |
| 5.1.11 | Liste Aktionen | 108 |
| 5.1.12 | Liste Neue Artikel | 108 |
| 5.1.13 | Produkte Suche | 109 |

| | | |
|----------|---|------------|
| 5.1.14 | Währung Auswahl | 110 |
| 5.1.15 | Warenkorb Mini | 110 |
| 5.1.16 | Warenkorb Eingabe Kundendaten | 112 |
| 5.1.17 | Bestellung Rechnung | 113 |
| 5.1.18 | Bestellung Lieferschein | 114 |
| 5.1.19 | Bestellung Liste | 115 |
| 5.1.20 | Allgemein Memo | 115 |
| 5.1.21 | Allgemein Hilfe | 116 |
| 5.1.22 | Adressen Auswahl | 116 |
| 5.2 | SysOrdner | 117 |
| 5.3 | Shops auf unterschiedlichen Seiten | 117 |
| 5.4 | Kategorien | 117 |
| 5.4.1 | mbi_products_categories | 118 |
| 5.4.2 | Seiten als Kategorien | 121 |
| 5.5 | Benutzergruppen und Template-Suffixe | 122 |
| 5.5.1 | Postidentverfahren | 128 |
| 5.6 | Listenansicht in unterschiedlichen Spalten | 128 |
| 6 | Versand- und Zahlungsmethoden | 131 |
| 6.1 | Versandkosten nach Stückzahl | 132 |
| 6.1.1 | Versand innerhalb Deutschlands | 132 |
| 6.1.2 | Versand ins Ausland | 134 |
| 6.1.3 | Ländergruppen definieren mit static_info_tables | 138 |
| 6.2 | Versandkosten nach Gewicht | 142 |
| 6.3 | Versandkosten nach Gewicht und Stückzahl | 143 |
| 6.4 | Preisnachlässe | 144 |
| 6.4.1 | Benutzer | 144 |
| 6.4.2 | Skonto | 144 |
| 6.4.3 | Bestellmenge und Produktpreis | 145 |
| 6.4.4 | Händler-Rabatte mit Preis2 | 147 |
| 6.5 | Mindestbestellwert | 148 |
| 6.6 | Kreditkartenzahlung | 149 |
| 6.6.1 | PayPal über Formular | 151 |
| 6.6.2 | PayPal über Extension | 152 |

| | | |
|----------|---|------------|
| 6.6.3 | ipayment über Formular | 161 |
| 6.6.4 | ipayment über Extension | 164 |
| 7 | Calendar Base | 167 |
| 7.1 | Installation | 168 |
| 7.1.1 | Jahresansicht | 168 |
| 7.1.2 | TS-Template | 172 |
| 7.2 | Saisonartikel | 176 |
| 8 | Mehrere Shops in einer TYPO3-Instanz | 179 |
| 8.1 | Katalog | 180 |
| 8.2 | Autorenverzeichnis | 182 |
| 9 | Alternative TYPO3-Shopsysteme | 187 |
| 9.1 | Commerce | 188 |
| 9.1.1 | Installation | 188 |
| 9.1.2 | TS-Template | 189 |
| 9.1.3 | Seiten und Plugins | 189 |
| 9.1.4 | PIDs zuweisen | 192 |
| 9.1.5 | Webmodul Commerce | 193 |
| 9.1.6 | Produkte und Artikel | 198 |
| 9.1.7 | Artikel erzeugen | 198 |
| 9.1.8 | Warenkorb | 203 |
| 9.1.9 | Bestellungen | 203 |
| 9.1.10 | Statistiken | 206 |
| 9.1.11 | HTML-Template | 208 |
| 9.2 | GSA Shop | 209 |
| 9.2.1 | Registrierung | 211 |
| 9.2.2 | Installation | 212 |
| 9.2.3 | Shopseiten und Plugins | 213 |
| 9.2.4 | TS-Template | 223 |
| 9.3 | PDF Shop | 223 |
| 9.4 | Trade | 226 |
| 9.4.1 | Plugin einfügen | 227 |
| 9.4.2 | Einzelansicht | 229 |

| | | |
|--------------------------------------|--|------------|
| 9.4.3 | Produkte und Kategorien | 229 |
| 9.4.4 | Shop mit registrierten Benutzern | 231 |
| 9.4.5 | TS-Template | 231 |
| 9.4.6 | Der Shop im Frontend | 235 |
| 9.4.7 | Order Status | 238 |
| 9.5 | Webformat Shop System | 241 |
| 9.5.1 | Shopseiten erstellen und Plugin einfügen | 242 |
| 9.5.2 | SysOrdner | 244 |
| 9.5.3 | Produkte | 244 |
| 9.5.4 | Shop mit registrierten Benutzern | 247 |
| 9.5.5 | TS-Template | 247 |
| 9.5.6 | Der Shop im Frontend | 250 |
| 9.5.7 | Bestellungen | 253 |
| Anhang | | 257 |
| A Übersicht TYPO3-Shopsysteme | | 259 |
| B Der Extension Manager | | 263 |
| B.1 | Sichtbarkeit von Extensions | 263 |
| B.2 | Sprachauswahl | 264 |
| B.3 | Editieren von Extension-Dateien | 265 |
| B.4 | Backup und Update | 265 |
| B.4.1 | Versionen von Extensions | 265 |
| B.4.2 | MD5-Fehler | 267 |
| B.4.3 | Installation als .t3x-Datei | 267 |
| B.5 | Fehlermeldung | 268 |
| B.6 | static_info_tables | 268 |
| B.7 | Erweiterung des Seiten-Moduls | 269 |
| C phpMyAdmin | | 271 |
| C.1 | Daten sichern | 274 |
| C.2 | Tabellen leeren und löschen | 276 |
| C.3 | Tabellen wiederherstellen | 277 |
| C.4 | Weitere Export-Formate | 282 |

| | | |
|----------|---------------------------------------|------------|
| C.5 | Daten suchen und ändern | 283 |
| D | tt_products im Constant Editor | 285 |
| D.1 | Referenz | 286 |
| D.2 | clickIntoBasket | 299 |
| D.3 | showNotinStock | 300 |

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/3.0/de/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

Vorwort der Autoren

Gut 80% aller Online-Kunden haben ihren Einkauf schon einmal wegen fehlender Funktionalität oder mangelnder Benutzerfreundlichkeit des Shops abgebrochen. Und: Fast die Hälfte aller Online-Shops in Deutschland basiert auf Open-Source-Lösungen.¹ Damit ist die Motivation zu diesem Buch auch schon deutlich umrissen: TYPO3 ist als Content Management System international etabliert, und zwar im Bereich privater Webseiten ebenso wie bei konzernweiten Internetlösungen – Grund genug also, den wichtigen Bereich des Online-Shoppings für dieses System erstmals in Buchform unter die Lupe zu nehmen.

Dieses Buch wendet sich somit an all jene, die einen Webshop auf TYPO3-Basis aufsetzen und betreiben möchten. Eine vorhandene TYPO3-Installation – sei es auch auf der Grundlage eines statischen Standard-Templates – ist hierfür Voraussetzung, denn es geht *nicht* um eine Einführung in TYPO3 selbst. Es genügt jedoch solides TYPO3-Grundlagenwissen, um Ihren Internetauftritt um einen Webshop zu erweitern. Auch HTML- und CSS-Kenntnisse sollten Sie mitbringen, um die notwendigen Anpassungen an das jeweilige Layout vornehmen zu können. Zudem verlangen Installation und Konfiguration des Shops Administratorrechte; ein Redakteur-Account reicht dazu nicht aus.

Dass wir den im Buch beschriebenen Beispielshop mit der TYPO3-Extension `tt_products` realisieren hat zwei naheliegende Gründe: Zum einen ist `tt_products` die mit Abstand am häufigsten heruntergeladene Shop-Erweiterung für TYPO3 überhaupt, und zum anderen kennen wir diese Extension als Hauptentwickler bzw. Dienstleister recht gut... ;)

Darüber hinaus bietet `tt_products` die Möglichkeit eines recht sanften Einstiegs in die Materie, so dass eben auch weniger versierte Anwender die Beispiele nachvollziehen und an ihre Bedürfnisse anpassen können; bei Bedarf können Sie mit den hier gewonnenen Einblicken Ihren Webshop um von uns nicht behandelte Features erweitern oder auch alternative TYPO3-Shop-Extensions in Betracht ziehen.

¹ Quelle: *Internetworld Business*, Ausgabe 9/07.

Wir verfolgen hier einen konsequent praktischen Ansatz, indem zunächst der Aufbau eines ganz einfachen Shops Schritt für Schritt beschrieben wird, der nach einigen wenigen Anpassungen bereits alle Kernaufgaben erfüllt: Listen- und Einzelansicht der Produkte, Warenkorb und Bestellung, E-Mail-Benachrichtigungen an Shop-Administrator und Kunden.

Ein HTML-Template und eine zugehörige CSS-Datei erleichtern Ihnen dabei die Arbeit und bieten zugleich einen Ausgangspunkt für Ihre eigenen Arbeiten am Design. Diese Beispieldateien zitieren und erläutern wir im Buch nur auszugsweise, entsprechend der jeweils behandelten Thematik. Sie finden sie jedoch zum Download unter anderem auf den Seiten des Verlags unter:

<http://www.opensourcepress.de/fileadmin/osp/download/typo3webshop.zip>

Mit diesen Template- und CSS-Dateien können Sie einen Shop relativ barrierearm gestalten, denn es wird in allen Bereichen auf Tabellen verzichtet. Die Dateien sind nach W3C-Validierung valide. Allerdings müssen Sie die Dateien an Ihre eigene Shop-Umgebung anpassen!

Im weiteren Verlauf behandelt dieses Buch zunehmend komplexere Anforderungen; hier nur einige Stichworte:

- Artikel und Varianten
- Bezahlmethoden
- verschiedene MwSt.-Sätze
- Gutscheine und Rabatte
- Verfügbarkeit von Produkten
- Versandkostenberechnung
- Kreditkartenzahlung über Payment-Gateways
- Verwaltung von Bestellungen
- ...

Aber es gibt, wie bereits angedeutet, mehr als ein Shop-System für TYPO3. Einige dieser Alternativen zu `tt_products` wollen wir Ihnen ebenfalls vorstellen und deren Installation wie auch *Grund*konfiguration kurz beschreiben. Wir haben uns zu diesem Zweck direkt mit den jeweiligen Entwicklerteams in Verbindung gesetzt und deren Hinweise in unsere Beschreibung einfließen lassen, so dass Sie auch hier größtmögliche Authentizität und „Überparteilichkeit“ voraussetzen dürfen.

Unser Anliegen – ganz im Sinne des Vorworts von Kasper Skårhøj – unsere Erfahrung bei der Arbeit an Shop-Lösungen mit TYPO3 an andere weiterzugeben, findet in diesem Buch einen ersten greifbaren Niederschlag. Die Thematik ist komplex, die Software permanenten Veränderungen unterworfen und praxistaugliche Dokumentation im Internet rar. Viele Stunden und Wochen des Tüftelns flossen so manchmal in nur wenigen Zeilen Code oder kurzen Textabschnitten zusammen – kurz: die (Vor)Arbeit war mühsam. Aber sie entsprach damit wohl den Erfahrungen, die viele andere TYPO3-Anwender beim Versuch, einen Webshop zum Laufen zu bringen, gemacht haben. Wenn wir die Durststrecken für andere nun wenigstens deutlich verkürzen können, so hätten wir unser Ziel erreicht. Und damit das Buch in hoffentlich künftigen Auflagen vielleicht Manches noch behandeln kann, was in der ersten Auflage aus technischen oder zeitlichen Gründen außen vor bleiben musste, würden wir uns über Ihre Kommentare und Verbesserungsvorschläge sehr freuen! Schreiben Sie uns, beispielsweise an:

`typo3webshop@opensourcepress.de`

Danksagungen

Mein Dank gilt den Entwicklern von TYPO3 und TYPO3-Extensions, die dazu beitragen, dass sich auch Leute mit kleinem Budget einen soliden Internet-Auftritt leisten können. Durch einen Auftrag von Michael Stucki habe ich mich schließlich an die Weiterentwicklung von `tt_products` gemacht. Ich danke allen Auftraggebern für deren Geduld beim Warten auf die Fertigstellung der gewünschten Funktionalitäten. Durch die Bereitstellung des deutschen Shop-Templates des BananaGuard.de-Betreibers Marcel Schwarz und eines Demo-Shops seiner Homepage konnten sich viele von der Extension `tt_products` am praktischen Beispiel schon ein Bild machen. Christian Lang hat mit der Extension `mbi_products_categories` die Verwendung eines hierarchischen Kategoriebaums im Backend ermöglicht. Viele haben mit ihren Beiträgen in Foren, auf der TYPO3-Wiki-Seite, durch Übersetzungen und Fehlerkorrekturen mitgeholfen, das System zu verbessern und zu verbreiten. Insbesondere Andrea Herzog-Kienast von der Kienast DV in der Messerstadt Solingen hat durch ihre Tutorials zum Shop einen wertvollen Beitrag geleistet. Außerdem bedanke ich mich bei allen Mitwirkenden, Veranstaltern und Teilnehmern am Nightfever im Kölner Dom und Papst Benedikt XVI. Dadurch werden auch weiterhin Menschen aus den Erlebnissen und Erfahrungen aus den Jahren des Weltjugendtags 2005 in Köln Kraft durch den Glauben schöpfen können.

Franz Holzinger

Köln, November 2007

Mein Dank gilt besonders meiner Familie, die mich während der „heißen Phase“ immer wieder ins normale Leben zurückgeholt hat. Danke an Jana und Jörg für Euer Verständnis und Eure Geduld! Danke auch an Erwin Hontheim für seine Vorschläge und geduldiges Korrekturlesen des ersten Manuskripts. Besonderer Dank auch an Günther Heger für seine Testeinkäufe als Nutzer einer Braillezeile und seine konstruktiven Ideen hinsichtlich einer einfacheren Nutzung von Shopsystemen durch blinde Surfer. Danke auch an die Entwickler der verschiedenen Shop-Extensions, besonders Rainer Kuhn, Mauro Lorenzutti, Steve Ryan und Ingo Schmitt für Tipps und Korrekturen. Danke auch an Mario Matzulla für seine Anbindung an den Calendar Base und die Unterstützung bei diesem Kapitel.

Danke auch an Ingo Fischer und Stephan Robotta vom I&I-Team und Udo Gerhards für ihre Unterstützung bei den Bezahlssystemen. Danke auch an Franz für die unermüdliche Arbeit an tt_products und an Henrik Jahn, Reinhard Niedermeier und natürlich an Kasper. Ohne all diese Menschen wäre dieses Buch nicht zustande gekommen.

Ganz besonderer Dank gilt unserem Verleger, Dr. Markus Wirtz, dessen Zeit und Geduld wir sicher in besonderem Maße in Anspruch genommen haben.

Andrea Herzog-Kienast

Solingen, November 2007

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://www.kreativkraft.de>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

Grundlagen der Installation in ein bestehendes TYPO3-System

In diesem einführenden Kapitel geht es darum, möglichst schnell ein rudimentäres Shopsystem auf der Grundlage der Extension `tt_products`¹ in eine vorhandene TYPO3-Installation zu integrieren. In einer Schritt-für-Schritt-Anleitung vollziehen Sie die notwendigen Handgriffe nach und lernen die Konzepte kennen, die der TYPO3-Shoplösung `tt_products` zu Grunde liegen. Die Bereitstellung der Grundfunktionalitäten entspricht aber natürlich noch keinem produktiv einsetzbaren Webshop! Unbedingt notwendige Konfigurationsschritte und Anpassungen an Ihr Produktangebot oder spezifische Liefer- und Zahlungsbedingungen sind fortgeschrittene Themen der späteren Kapitel.

Für die ungeduldigen (und bereits TYPO3-erfahrenen) Leser geben Querverweise an, wo das jeweilige Thema detailliert behandelt ist. Es wird jedoch empfohlen, zunächst diesen praktischen Einstieg nachzuvollziehen.

¹ <http://www.ttproducts.de> und <http://www.ttproducts.org>

Folgende Voraussetzungen werden in diesem Schnelleinstieg geschaffen:

- Installation von tt_products und – falls noch nicht vorhanden – notwendiger „Hilfs-Extensions“
- Anlegen eines TypoScript-Shop-Templates
- Einbinden des neuen TypoScript-Shop-Templates
- Anlegen eines SysOrdnerns für die Produkte
- Erstellen der notwendigen Shopseiten über das Einfügen von Plugins

1.1 Das Shopsystem tt_products

Die Extension tt_products dient als Basis für diesen Shop, der im Wesentlichen folgende Bereiche bzw. Funktionalitäten umfasst:

Listenansicht

Hier erscheinen die Produkte eines Shops – gegebenenfalls nach Produktgruppen gegliedert – in einer Übersicht; die Möglichkeiten, Produktinformationen darzustellen und nach eigenen Vorstellungen aufzubereiten, sind in dieser Ansicht eingeschränkt. Hier erfolgt auch die Auswahl zum Kauf, ggf. über eine Zahlenangabe in einem Mengenfild.

Einzelansicht

Über einen Klick auf ein Produktbild oder den Titel des Produkts in der Listenansicht gelangt der Kunde zur Einzelansicht, in der dann umfassende Produktinformationen hinterlegt sind. Hier erfolgt, wenn gewünscht, ebenfalls die Auswahl zum Kauf, ggf. über eine Zahlenangabe in einem Mengenfild.

Warenkorbansicht

Die vom Kunden für den Kauf ausgewählten Produkte sind hier in einer Übersicht abrufbar.

Bezahlungsansicht

In Verbindung mit Preisinformationen, Lieferbedingungen, Lieferkosten u. Ä. erhält der Kunde eine letzte Übersicht über seinen Einkauf, bevor er mit der eigentlichen

Bestellung

also in der Regel einem Klick auf einen Button wie **Bestellung absenden**, den Kaufvorgang rechtsverbindlich abschließt.

Diese recht schematische Abwicklung findet ihren Niederschlag auch im Frontend, denn jedem dieser genannten Schritte ist in der Regel eine Seite im Seitenbaum der Webpräsenz zugeordnet.

Natürlich lassen sich zu diesen grundlegenden Funktionalitäten weitere hinzufügen, beispielsweise ein *Gutscheinsystem* (vgl. S. 103) oder ein *Merkzettel* (vgl. S. 115).

1.2 Benötigte Extensions

Um `tt_products` in ein bestehendes TYPO3-Projekt einzubinden, sind die folgenden (nach Drucklegung des Buches möglicherweise bereits in höheren Versionsnummern vorliegenden) Extensions zu installieren²:

`tt_products` 2.6.0

Die eigentliche Shop-Extension.

Table Library (`table`) v0.1.13

Enthält den Code für die Datenbankabfragen einschließlich der Berücksichtigung verschiedener Sprachen. Verwenden Sie immer die Version, die bei `tt_products` im Extension Manager angegeben ist. Andernfalls könnte es zu unerwarteten Datenbankfehlern kommen.

FE BE Library (`fh_library`) v0.0.19

Enthält Erweiterungen, die künftig in die Extension `div` (s. u.) des Extension Coordination Teams (ECT) ausgelagert werden.

Static Methods for Extensions (`div`) v0.0.13

Library Extension des ECT. Sie enthält Bibliotheksfunktionen und wird in Zukunft die `fh_library` ablösen; hier ist sie aus Kompatibilitätsgründen notwendig.

`xAjax` (`xajax`)

Diese Ajax-Funktionalität wird bei `tt_products` 2.6.0 u. a. dann benötigt, wenn Sie Kategorie-Menüs in Select-Boxen verwenden.

Bei entsprechender Systemkonfiguration sollten bei der Installation von `tt_products` diese ergänzenden Extensions automatisch mit installiert werden. Die Erweiterung `xajax` benötigen Sie u. U. gar nicht und können sie über die Auswahl `Ignore` ausschließen.

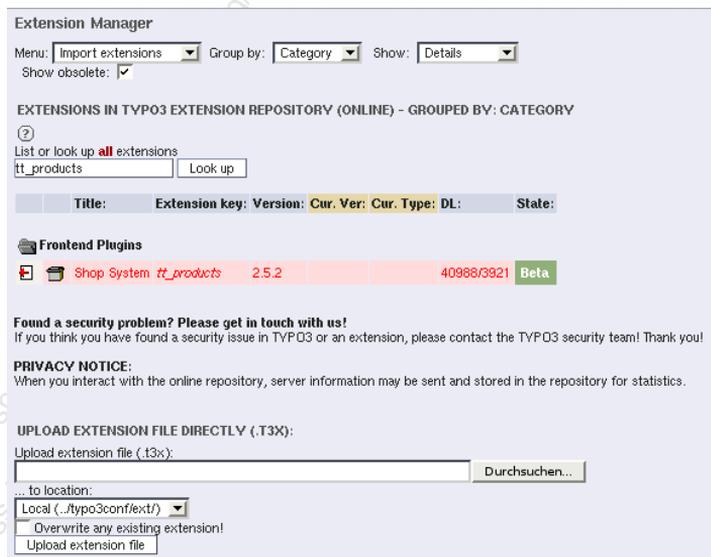
Wählen Sie also nun in der Modulleiste des Backend den Extension Manager (**Ext Manager**) und in dessen Menü die Option **Import extensions**. Im daraufhin erscheinenden Suchfeld geben Sie `tt_products` ein und senden die

² Grundsätzlich benötigen Sie für die Installation Administratorrechte im System.

Anfrage mit **Look up** ab. Möglicherweise – insbesondere bei einer Default-Konfiguration des Systems – bleibt die Suche erfolglos, denn `tt_products` befindet sich – wie übrigens fast alle Extensions – *nicht* in der Liste der sog. „reviewed extensions“.³

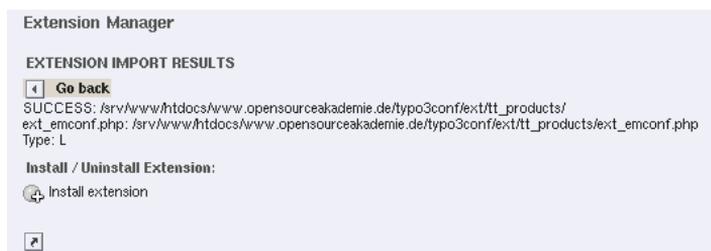
Darum müssen Sie zunächst die Liste auszuwählender Extensions erweitern: Wählen Sie im Menü des Extension Managers den Punkt **Settings** und machen Sie ein Häkchen bei der ersten Option **Enable extensions without review (basic security check)**. Kehren Sie anschließend in das **Import**-Menü zurück und versuchen Sie Ihr Glück bei der Suche nach `tt_products` erneut. Nun sollte die Extension erscheinen (Versionsangabe beachten!) und über das Pfeil-Icon ganz links in der Zeile importierbar sein (Abbildung 1.1).

Abbildung 1.1:
Erfolgreiche Suche
im Extension
Manager



Nach dem Klick meldet sich das System mit einer Erfolgsmeldung (Abbildung 1.2).

Abbildung 1.2:
Erfolgreicher Import



³ Weitere Auskünfte zum Extension Manager gibt Anhang B ab Seite 263.

Beim Klick auf den Link **Install extension** wird entweder direkt die Installation von `tt_products` angestoßen oder eben auf die bereits erwähnten Abhängigkeiten mit anderen Extensions hingewiesen, um diese zunächst zu installieren (Abbildung 1.3).

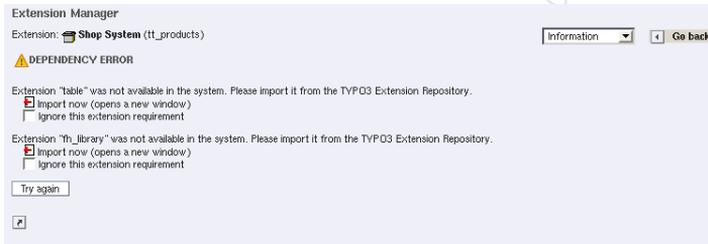


Abbildung 1.3:
Abhängigkeiten von
anderen Extensions

In diesem Fall müssen Sie zunächst die geforderten Extensions einzeln importieren und installieren (jeweils über die entsprechenden Links **Import now** bzw. **Install now**) – Sie können sie auch ignorieren und später nachinstallieren. Sind diese Voraussetzungen geschaffen, prüft ein Klick auf **Try again** ein weiteres Mal die Abhängigkeiten, um festzustellen, dass nun der Installation von `tt_products` selbst nichts mehr im Wege steht. Die dabei notwendigen, recht umfangreichen Schritte, die insbesondere die Schaffung und Modifizierung von Datenbanktabellen betreffen, werden aufgelistet und können (von kundigen Administratoren) entweder einzeln per Häkchen oder (wie für unsere Beispielininstallation) durch Klick auf **Update** am Ende der Seite komplett bestätigt werden (Abbildung 1.4).

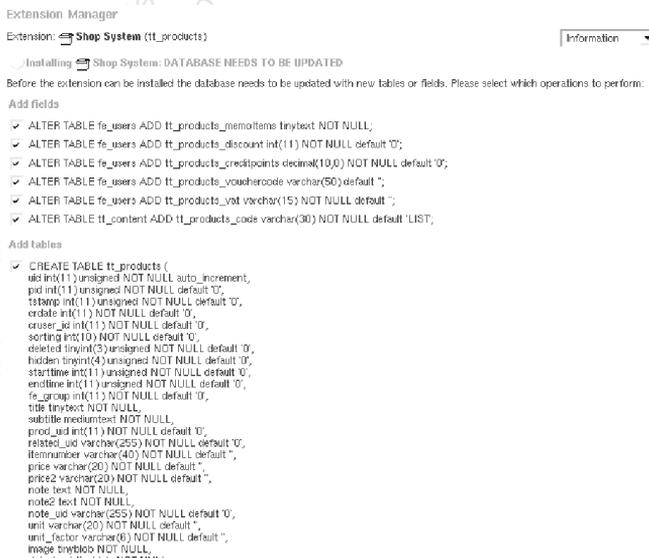
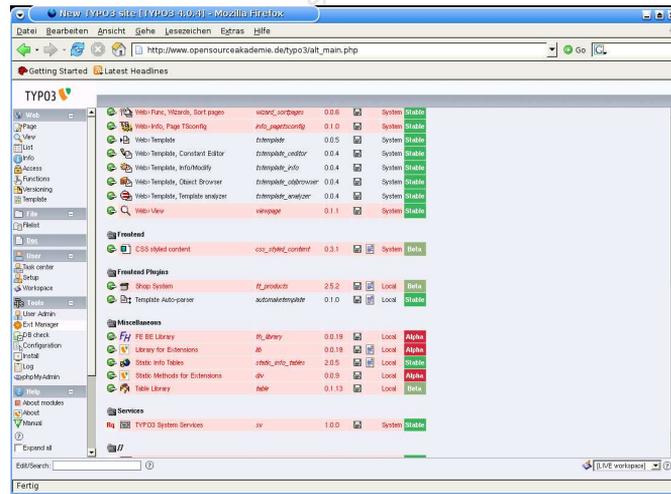


Abbildung 1.4:
Notwendige
Veränderungen der
System-Datenbank

Nach dem Update ist die Installation abgeschlossen, was nicht eigens mit einer Erfolgsmeldung quittiert wird; vielmehr wird wieder der Extension Manager angezeigt, wo Sie weitere Extensions auswählen und installieren können. Zur Kontrolle lassen Sie sich im Menü **Loaded Extensions** anzeigen, ob **tt_products** nun in Ihr System integriert ist, was in der Kategorie **Frontend Plugins** auch tatsächlich zu erkennen ist (Abbildung 1.5).

Abbildung 1.5:
tt_products steht zur
Verfügung



Noch ein wichtiger Hinweis zur eventuell notwendigen Fehlersuche: Im Install-Tool (im Webmodul unter **Installation** zu finden) können Sie sich Fehlermeldungen anzeigen lassen, indem Sie folgende Variable setzen:

```
[SYS][displayErrors] = 1
```

Dazu wählen Sie **All Configuration**, suchen **displayErrors** und setzen den Wert 1 (Abbildung 1.6).

Abbildung 1.6:
Installation: Fehler
anzeigen



Probleme mit den Extensions lassen sich auf diese Weise „einkreisen“ und entsprechend einfacher lösen.

```
[SYS][sqlDebug] = 1
```

Ebenfalls unter **All Configuration** finden Sie die Einstellung `sqlDebug`. Wenn Sie diese aktivieren, werden alle SQL-Fehlermeldungen ganz oben am Bildschirm ausgegeben. Dies ist insbesondere während der Entwicklung sinnvoll, da Fehler in einer SQL-Anweisung meist darauf hinweisen, dass auch die dahinterliegende Extension nicht richtig funktionieren wird.

1.3 TS-Templates

TYPO3 nutzt zur Steuerung bzw. Formatierung der Ausgabe bekanntlich so genannte *TS-Templates*, die wiederum über eine TYPO3-eigene Beschreibungssprache namens *TypoScript* (TS) erstellt werden.⁴

Während das Haupttemplate, das der Root-Seite der Webpräsenz zugeordnet ist, das Gesamtlayout des Webauftritts bestimmt, bringen auch Extensionen ihre jeweiligen Templates mit oder lassen sich über Templates anpassen. Das gilt auch für `tt_products`; dessen und die Templates anderer Extensionen liegen üblicherweise in einem SysOrdner.

Gibt es einen solchen SysOrdner in Ihrem System noch nicht, so erstellen Sie nun einen namens **Templates** und legen darin einen neuen Datensatz an (**Neuen Datensatz anlegen**). Wählen Sie **Template** aus und geben Sie als Titel `+tt_products` an, so dass Sie unmittelbar erkennen können, welche Extension hier modifiziert wird (entsprechend bezeichnet man diese Dateien auch als *Extension-Templates*).

In **Setup** und **Constants** nehmen Sie später Einstellungen des Shops vor, wie Sie es auch von Ihrem Haupttemplate gewohnt sind. Weisen Sie jetzt in einem ersten Schritt diesem Extension-Template unter **Include static (from extensions)** das statische Template **Shop System Old Style tt_products** zu, das nach der Installation von `tt_products` in der Auswahlliste rechts zur Verfügung steht (Abbildung 1.7).

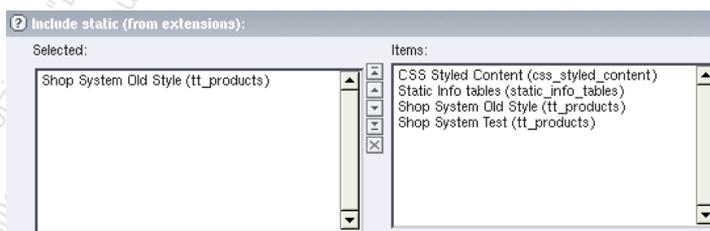


Abbildung 1.7:
Zuweisung des
statischen Template

⁴ Eine ausführliche Beschreibung dieser TYPO3-Grundlagen finden Sie in dem Standardwerk von Altmann/Fritz/Hinderink, „TYPO3 – Enterprise Content Management“, das ebenfalls bei Open Source Press erschienen ist.

Gehen Sie nun zurück zum Haupttemplate des Webauftritts, wählen Sie dort [Klick here to edit whole template record](#) und binden Sie das neue Template `tt_products` über **Include basis template** ein. Extension und Template sind damit installiert und in das bestehende System integriert.⁵

Ein Shop benötigt nun allerdings noch Produkte, die zum Verkauf stehen; diese werden im folgenden Abschnitt im System bereitgestellt.

1.4 Der Produkte-SysOrdner

Produkte werden bei `tt_products` üblicherweise in einem (oder mehreren) speziellen Ordnern gepflegt, so genannten *SysOrdnern*. Diese bilden den Datenpool, auf den die einzelnen Shop-Bereiche (Listenansicht, Einzelansicht etc.) zugreifen, und darum legen Sie zunächst einen solchen Warenbestand an.

In größeren Shops empfiehlt sich übrigens eher eine Pflege verschiedener Produktgruppen auf mehreren Seiten (und SysOrdnern), so dass Sie für jede dieser Gruppe einfach eine Listenansicht erzeugen können. Die Übersicht über die Produkte bleibt so auch bei einem größeren Produktangebot gewahrt.

Hier erstellen Sie jedoch über **Liste | Neuen Datensatz anlegen** als „Container“ eine Seite vom Typ **SysOrdner** (Abbildung 1.8).

Abbildung 1.8:
SysOrdner für die
Produkte

The screenshot shows the 'Neuen Datensatz anlegen' (Create new record) form in TYPO3. The form is divided into several sections:

- Seite verstecken:** A checkbox labeled 'Wenn Sie dies markieren, wird die Seite nicht angezeigt.' is currently unchecked.
- Typ:** A dropdown menu is set to 'SysOrdner'. A tooltip explains: 'Wählen Sie den Seitentyp aus. Dies hat Einfluss darauf, ob die Seite als normale Webseite dargestellt oder für andere Zwecke genutzt wird.'
- Titel:** A text input field contains the word 'Produkte'. A tooltip explains: 'Geben Sie den Titel der Seite ein! Dieses Feld ist ein Pflichtfeld. Der Titel wird normalerweise in den Frontend Menüz gezeigt.'
- Tsconfig:** A large empty text area for 'Spezielle TypoScript-Konfiguration dieser Seite.' with a 'TS' icon on the right. Below it, 'Seitenbaum stoppen:' and 'Bearbeitung erfordert Admin-Rechte:' are both unchecked.
- Allgemeine Datensatzsammlung:** A dropdown menu is set to 'Seite'. A tooltip explains: 'Legen Sie fest, welche "Allgemeinen Datensatzsammlungen" von dieser Seite verwendet werden sollen.' Below it, 'Ist Anfang der Webseite:' is unchecked.

⁵ Dieses Verfahren können Sie auch weiter ausbauen: Das Haupttemplate muss nicht notwendigerweise über Setup und Constants verfügen, sondern kann auch nur dem Verweis auf einzelne Templates im Templates-SysOrdner dienen; so können Sie für alle Bereiche wie Sprachen, einzelne Marker, Menüs etc. eigene Templates erstellen und einbinden. Auch dazu finden Sie weitere Informationen in dem erwähnten Handbuch.

Darin wiederum erzeugen Sie per Klick auf **Neuen Datensatz anlegen** und die Auswahl **Produkte** einen ersten Produkt-Datensatz (Abbildung 1.9).



Abbildung 1.9:
Datensatz anlegen:
Auswahlliste
Produkte

Im Backendformular vergeben Sie einen Titel, eine Artikelnummer und den Preis. Auch einen kurzen Informationstext sowie ein Bild können Sie hier hinterlegen und mit dem Datensatz speichern (Abbildung 1.10).

Abbildung 1.10:
Produkt im Backend
anlegen
(Ausschnitt)

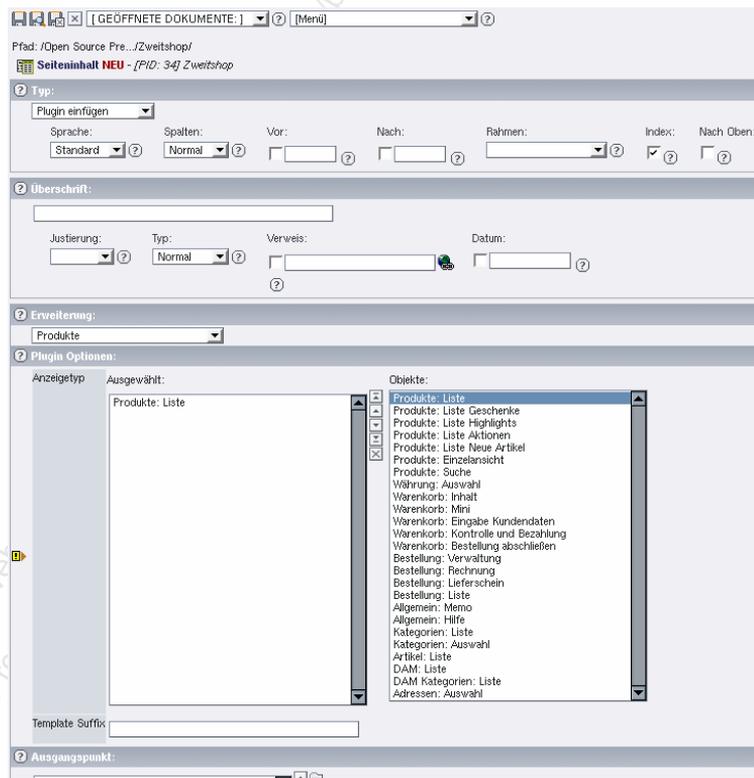
1.5 Das tt_products-Plugin

Die Vorbereitungen – nämlich die systemweite Bereitstellung der Extension samt Template sowie eines Produktverzeichnisses mit einem ersten Produkt – sind damit abgeschlossen, so dass Sie nun die Shop-Erweiterung über Plugins in die Webseiten einbinden können.

1.5.1 Plugin einfügen

Erstellen Sie eine Hauptseite für Ihren Shop und legen Sie darin ein neues Seiteninhaltelement **Produkte** an, das Sie unter **Plugins** finden (legen Sie die Seite über den Assistenten an, heißt das Plugin übrigens **Shop System**). Mit dieser Auswahl ändert sich erwartungsgemäß das Backendformular (Abbildung 1.11).

Abbildung 1.11:
Plugin einfügen



Wählen Sie in den **Plugin Optionen** nun das Plugin-Objekt **Produkte: Liste** und speichern Sie den Datensatz ab. Ihre Seite verfügt nun über einen Container, in den im Folgenden die einzelnen Produkte eingepflegt werden, denn auch das Shopsystem folgt dem TYPO3-eigenen Prinzip, Seiten und *Seiteninhalte* zu trennen.

Damit das Shop-Plugin auf den oben bereits angelegten SysOrdner zugreifen kann, muss dieser eingebunden werden. Fügen Sie darum in dem soeben angelegten Shop-Plugin (Inhaltelement) den Produkte-SysOrdner als **Ausgangspunkt** (Startingpoint) dieses Elements hinzu. Das Backend sollte dann aussehen wie der Ausschnitt in Abbildung 1.12.

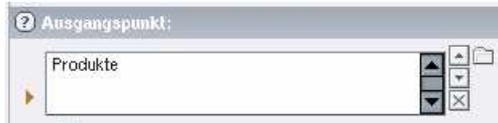


Abbildung 1.12:
Ausgangspunkt

Es ist essentiell, den Produkte-SysOrdner ins Plugin der Shopseiten einzubinden, denn dort wird vom Shopsystem nach den Produkten gesucht. Wenn kein SysOrdner zugewiesen ist, der Ordner leer ist oder ein falscher SysOrdner zugewiesen wurde, kann das Shopsystem die Produkte nicht finden und damit auch nicht anzeigen! Das gilt für alle Seiten mit dem Shop-Plugin und für alle SysOrdner, wenn Sie mehrere für z. B. verschiedene Produktgruppen nutzen.

1.5.2 Shopseiten erstellen

Kommen wir also zu den weiteren Shopseiten bzw. zu den am Beginn von Abschnitt 1.1 genannten Basisfunktionalitäten.

Listenansicht der Produkte

Die Listenansicht haben Sie bereits mit dem ersten Plugin auf der Hauptseite des Shops erstellt (Abbildung 1.11). Ein Klick auf das **Shop**-Menü gibt einem Besucher Ihrer Webseite eine Übersicht über die von Ihnen angebotenen Produkte – in diesem Fall zunächst nur das eine Testprodukt.

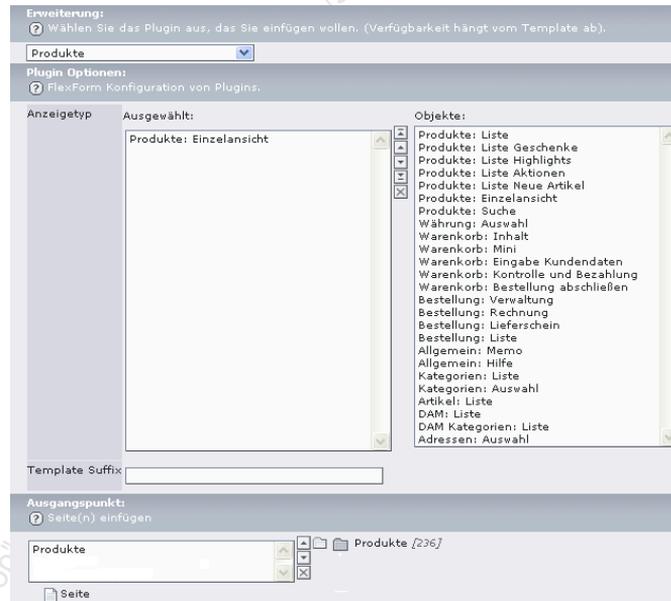
Erstellen Sie unterhalb dieser Listenansicht weitere Seiten und setzen Sie auf jeder Seite – außer der Listenansicht – das Häkchen bei **Nicht im Menü**, damit diese Seiten später nicht im Menü auftauchen. Da diese Inhalte keine regulären Seiteninhaltelemente sind, sondern die Inhalte abhängig vom Startpunkt bzw. von der Auswahl des Frontendbenutzers erstellt werden, dürfen die Seiten nicht im Menü sichtbar sein. Machen Sie sich dies am Beispiel der Einzelansicht eines Produkts klar: Die Einzelansicht kann nur aus der Listenansicht heraus aufgerufen werden und wird dann vom System generiert. Wäre eine Seite **Einzelansicht** im Menü sichtbar, würde der Klick darauf eine Fehlermeldung nach sich ziehen, denn die Seite kann ja nicht „wissen“, zu welchem Produkt die Einzelansicht angezeigt werden soll.

Eine Ausnahme bildet die Warenkorbseite: Möchten Sie diese im Menü anzeigen, damit der Kunde jederzeit über das Menü auf den Warenkorb zugreifen kann, lassen Sie diese Seite im Menü sichtbar. Sofern noch keine Produkte in den Warenkorb gelegt wurden, erscheint eine Nachricht, dass der Warenkorb leer ist.

Einzelansicht der Produkte

Erstellen Sie eine Seite, fügen Sie das Shop-Plugin ein und legen Sie ein Inhaltselement vom Typ **Shop Plugin** hinein. Wählen Sie darin unter den **Plugin Optionen** den Punkt **Produkte: Einzelansicht** (Abbildung 1.13).

Abbildung 1.13:
Einzelansicht eines
Produkts



Weisen Sie als **Ausgangspunkt** den SysOrdner mit Ihren Produkten zu. Sie benötigen nicht unbedingt eine separate Seite für die Einzelansicht. Wenn Sie diese nicht explizit zuweisen, wird sie auf der gleichen Seite angezeigt wie die Listenansicht, was allerdings aus Gründen der Übersichtlichkeit nicht zu empfehlen ist. Auch auf Grund des weiteren Ausbaus Ihres Shops in den folgenden Kapiteln sollten Sie unbedingt eine eigene Seite für die Einzelansicht erstellen.

Im Frontend führt ein Klick auf den Produkttitel in der Listenansicht nun auf eine neue Seite mit der Einzelansicht und weiteren Detailinformationen, in diesem Fall zunächst nur dem kurzen Infotext.

Warenkorb

Im nächsten Schritt erstellen Sie die Seite, auf der im Frontend der Inhalt des Warenkorbs angezeigt wird.

Legen Sie die Seite an, fügen Sie das Plugin ein und legen Sie ein Inhaltselement vom Typ **Shop Plugin** hinein. Wählen Sie darin unter den **Plugin**

Optionen den Punkt **Warenkorb: Inhalt**. Dem Warenkorb muss der SysOrdner ebenfalls als **Ausgangspunkt** zugewiesen werden.

Warenkorb – Kontrolle und Bezahlung

Diese neue Seite ermöglicht im Frontend die Auswahl der Liefer- und Zahlungsmethode. Erstellen Sie die Seite, fügen Sie das Plugin ein und legen Sie ein Inhaltselement vom Typ **Shop Plugin** hinein. Wählen Sie darin unter den **Plugin Optionen** den Punkt **Warenkorb: Kontrolle und Bezahlung** aus.

Warenkorb – Bestellung abschließen

Die tatsächliche Bestellung wird über die Bereitstellung der folgenden Seite ermöglicht: Legen Sie eine neue Seite an, fügen Sie das Plugin ein und legen Sie ein Inhaltselement vom Typ **Shop Plugin** hinein. Wählen Sie darin unter den **Plugin Optionen** den Punkt **Warenkorb: Bestellung abschließen** aus.

Wenn Sie alle hier beschriebenen Shopseiten erstellt haben, sollte der Seitenbaum Ihres Systems in etwa dem in Abbildung 1.14 entsprechen.



Abbildung 1.14:
Seitenbaum des
Shops

Checkliste

Falls etwas nicht korrekt funktioniert, sollten Sie die folgende Checkliste durchgehen und abhaken. Insbesondere falsches oder fehlendes Einbinden von Plugins und SysOrdnern sind zu Beginn „beliebte“ Fehler.

- Sind alle notwendigen Erweiterungen in den korrekten Versionen installiert?
- Ist das Statische Template **Shop System Old Style tt_products** über **Include static (from extensions)** eingebunden?
- Ist das **TS-Shop-Template** auch dem Haupttemplate zugewiesen (**Include basis template**)?

- Ist der SysOrdner als Ausgangspunkt gesetzt?⁶
- Ist der SysOrdner, das Produkt oder die Shopseite versteckt?
- Ist der Frontend-Cache gelöscht?⁷

⁶ Sofern Sie im Setup `pid_list` verwenden muss der Ausgangspunkt bei den einzelnen Shopseiten nicht gesetzt werden. Da Sie evtl. mehrere SysOrdner für Ihre Produkte verwenden, ist in der Beispielinstallation der (oder die) SysOrdner jeweils der Seite zugewiesen um die Fehleranfälligkeit zu reduzieren.

⁷ Für die Testphase können Sie den Cache zunächst komplett abschalten. Fügen Sie dazu in Ihrem noch leeren Extension-Template folgenden Befehl ins Setup ein: `config.no_cache=1`.

2

Grundkonfiguration

Das Shop-Gerüst steht, einsetzbar ist es aber ganz offensichtlich noch nicht. Darum wird es im Folgenden um einige Grundfragen der Konfiguration gehen.

2.1 TS-Template

Die Konfiguration, also die Anpassung des Shops an die jeweiligen Bedürfnisse nehmen Sie in dem auf S. 23 angelegten TS-Template `+tt_products` vor, und zwar sowohl in den Constants als auch im Setup des Extension-Template.¹

¹ Während im Setup komplexere Konfigurationen für Bilder und andere Objekte der Extensions oder TYPO3 stehen, werden in den Constants nur einfache, konstante Werte für Zahlen und Texte definiert, die vom Setup ausgelesen werden können. Jede dieser Konstanten taucht bei `tt_products` fast immer unter demselben Namen im Setup auf.

2.1.1 Constants

Wechseln Sie in Ihren Template-Ordner zum Template `tt_products`; der *Constant Editor* hilft bei einigen grundlegenden Konfigurationsschritten. Klicken Sie zunächst im Webmodul auf **Templates**, um vom Seiten- in das Template-Modul zu wechseln. Rufen Sie danach den Constant Editor über die obere rechte Combobox auf und setzen Sie zunächst Häkchen dort, wo Sie Änderungen vornehmen werden, zunächst also auf die Bereiche, die Sie auch in Abbildung 2.1 sehen. Klicken Sie anschließend den **Update**-Button an, um die ausgewählten Felder bearbeiten zu können.

Abbildung 2.1:
Constant Editor

The screenshot shows the Constant Editor interface with the following settings:

- TAX included in database prices?** Defaults: 1
- TAX percentage** Type= Double (means, "use . as decimal point") Defaults: 25
- always in Stock** The stock will never get reduced. Defaults: 1
- PID for list items display** Set this in your setup manually for defining where conditions for several pages. Range: 0 - Defaults:
- PID for basket page** [plugin_tt_products.PIDbasket] If you want every change in the number of items anywhere to go to a specific page (eg. the shopping basket page), enter the PID here. Range: 0 - Defaults:
- PID for store root** [plugin_tt_products.PIDstoreRoot] This is the PID of the rootPage of the store. If not set the store will operate over all pages from the root of the site. But if the site has many pages, performance may improve. Defaults:
- PID for search page** [plugin_tt_products.PIDsearch] If you want all product searches to go to a specific page, enter the PID it here! NOTE: If you set this PID, all searchqueries will (must) be handled with a list content element with the code "search" on that page. Defaults:
- PID for single item display** [plugin_tt_products.PIDItemDisplay] Set this in your setup manually for defining where conditions for several pages. Range: 0 - Defaults:
- PID for payment page** [plugin_tt_products.PIDpayment] Range: 0 - Defaults:
- PID for finalize page** Range: 0 - Defaults:

Zunächst ändern Sie nur die folgenden Werte:

Tax percentage

Der deutsche MwSt.-Satz wird mit 19% angegeben.²

always in Stock

Dieses Häkchen veranlasst, dass die Produkte immer im Shop gelistet sind, also der Warenbestand nicht durch Bestellungen reduziert wird. Nach den notwendigen Testbestellungen müssen Sie die Produkte also nicht erneut einpflegen. Nachträglich können Sie diese Option jederzeit entfernen und einen „echten“ Bestand pflegen.

² Das wird in unserem Beispielshop später noch geändert, da hier Bücher verkauft werden sollen, für die der reduzierte MwSt.-Satz von 7% gilt.

Des Weiteren werden die PIDs der jeweiligen Seiten zugeordnet; die PID (*Page Identifier*) ist in diesem Fall ein eindeutiger Identifier einer Seite im Seitenbaum der Webpräsenz. Durch Überfahren des jeweiligen Seiten-Icons in Ihrem Seitenbaum (Seitenansicht) mit dem Mauszeiger erhalten Sie diesen numerischen Wert, der in den Constants bzw. im Constant Editor gesetzt werden muss.

PID_list

PID des SysOrdnerns

PID for single item display

PID der Einzelansicht

PID for basket page

PID des Warenkorbs

PID for list items display

PID der Listenansicht

PID for payment page

PID der Seite Kontrolle und Bezahlung

PID for finalize page

PID der Bestellseite

Wechseln Sie jetzt in den **Template Tools** über die obere rechte Combobox aus dem Constant Editor zur Ansicht **Info/Modify** und wählen über das Bleistiftsymbol **Constants**, sehen Sie dort bereits den entsprechenden TS-Code (Abbildung 2.2).

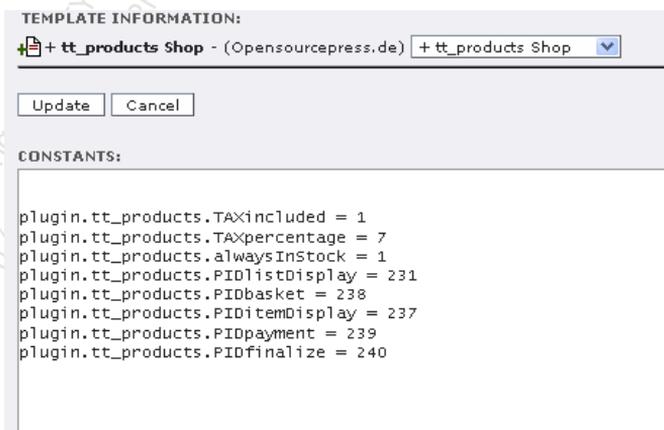


Abbildung 2.2:
Constants des Shops

Grundsätzlich sollten Sie jede neue oder per Constant Editor veränderte Definition kommentieren und thematisch Verwandtes in zusammenhängenden und entsprechend benannten Blöcken verwalten.

Mit dem weiteren Ausbau des Shops kommen Definitionen hinzu, und Sie werden dieses Vorgehen aus Gründen der Übersicht schnell schätzen lernen. Der Constant Editor fügt neue Definitionen stets am Ende des Dokuments an; verschieben Sie diese später in Bereiche, zu denen sie logisch gehören, und kommentieren Sie, was dort definiert ist.

Ein Beispiel-Setup finden Sie auch in der Datei `setup.txt` im Verzeichnis `tt_products/static/css_styled`. Allerdings sollten Sie niemals alles in Ihr Setup kopieren, sondern nur das wirklich Notwendige, das Sie auch ändern werden. Ebenso können Sie auch Abschnitte aus der Datei `constants.txt` im selben Verzeichnis in Ihre Constants kopieren. Wir empfehlen jedoch, zumindest die Grundkonfiguration anhand der im weiteren Verlauf beschriebenen Einstellungen vorzunehmen.

Vervollständigen Sie nun Ihre Konstanten-Definitionen wie folgt:

```
##### Constants +tt_products Shop #####

### Steuern ###
plugin.tt_products.TAXincluded = 1
plugin.tt_products.TAXpercentage = 7

### Immer ab Lager ###
plugin.tt_products.alwaysInStock = 1

### PIDS ###

## Listenansicht##
plugin.tt_products.PIDlistDisplay = 231

## Warenkorbansicht ##
plugin.tt_products.PIDbasket = 238

## Einzelansicht ##
plugin.tt_products.PIDitemDisplay = 237

## Kontrolle und Bezahlung ##
plugin.tt_products.PIDpayment = 239

## Bestellung abschließen
plugin.tt_products.PIDfinalize = 240

## Nur ein Produkt bestellen dürfen
plugin.tt_products.basketMaxQuantity = 1
```

Selbstverständlich sind die PIDs (ebenso wie im Folgenden Firmen- und Servernamen, E-Mail-Adressen u. Ä.) den lokalen Gegebenheiten Ihres Systems anzupassen! Einige der oben gesetzten Werte können Sie nicht (mehr) über den Constant Editor setzen: Mit `plugin.tt_products.TAXincluded = 1` zum Beispiel haben Sie definiert, dass Preise als Bruttopreise (also inkl. MwSt.) beim Anlegen der Datensätze eingegeben werden. Ebenso sind die Preise in den Setups dann als Bruttopreise zu verstehen. Diese Einstellung ist manuell in den Constants zu ergänzen.

Mit der Option `plugin.tt_products.basketMaxQuantity = 1` ist festgelegt, dass von jedem Produkt nur ein Stück bestellt werden darf, damit an dieser Stelle die auf 1 Exemplar kalkulierten Lieferkosten korrekt ausgewiesen werden. Würde in diesem Zustand des Shops mehr als ein Produkt bestellt, würden die später im Setup gesetzten Porto- und Versandkosten pauschal für die gesamte Bestellmenge erhoben. Im Kapitel 6 ab Seite 131 wird das Thema Lieferbedingungen und Versandkosten ausführlich behandelt.

Damit ist ein sehr einfacher Shop bereits lauffähig, und Sie können erste Testbestellungen durchführen.

Solange weder eine Benutzerverwaltung noch das Zahlungssystem eingerichtet ist und auch die juristischen Minimalanforderungen (Stichwort: AGB) nicht erfüllt sind, darf dieser Shop selbstverständlich nicht produktiv eingesetzt werden! Mit der Implementierung und Konfiguration verschiedener Zahlungsmethoden befasst sich Kapitel 6. Eine erste, einfache Zahlungs- und Lieferkonfiguration wird weiter unten in diesem Kapitel im TS-Setup behandelt.

Checkliste

- Sind alle PIDs korrekt gesetzt?
- Ist der Ausgangspunkt samt Rekursion auf jeder notwendigen Shopseite gesetzt? Oder sind `pid_list` und `recursive` so gesetzt, dass alle Seiten mit Produkten damit erfasst werden?
- Sind alle notwendigen Seiten sichtbar?

2.1.2 Setup

Im Setup definieren Sie Funktionalitäten Ihres Shops, die nicht über die Constants oder später über die Marker im HTML-Template gesetzt werden. Hier soll es zunächst nur um *grundsätzliche* Elemente der Zahlung und der Lieferung gehen. In den folgenden Kapiteln werden Sie die TS-Setup-Einstellungen sukzessive erweitern.

Ein kleiner Tipp: Wenn Sie Ihren Shop zunächst nur testen, können Sie den Cache ausschalten, um Probleme mit zwischengespeicherten und vom System genutzten „alten“ Angaben zu vermeiden. Erfahrene Benutzer werden diesen Stolperstein kennen. Den Cache schalten Sie aus, indem Sie am Anfang des Setup Folgendes setzen:

```
config.no_cache = 1
```

In diesem rudimentären Beispielshop werden also zunächst einige Möglichkeiten der Zahlung (*Payment*) und der Lieferung (*Shipping*) in Abhängigkeit voneinander definiert.

Einige wichtige Bemerkungen zur Logik der Implementierung jedoch vorweg: Die (später zahlreicher werdenden) Zahlungsoptionen wie „Kreditkarte“, „Rechnung“, „Vorkasse“ u. Ä., aus denen der Kunde auswählen kann, sind nicht frei wählbar; sie sind vielmehr abhängig von der Art der Lieferung. Ein Beispiel: Während Sie einem Selbstabholer die Möglichkeit der Barzahlung einräumen, soll eine Lieferung ins außereuropäische Ausland nur gegen Zahlung per Kreditkarte möglich sein. Kurz: Der Weg der Sendung bestimmt die Zahlungsweise (und natürlich die Höhe der Versandkosten).

Dieses Prinzip soll hier zunächst auf der Grundlage folgender Optionen veranschaulicht werden, die so auch in den Bereichen **Versand und Verpackung** bzw. **Zahlungsweise** im Warenkorb auftauchen und damit dem Kunden zur Auswahl stehen:

Versand und Verpackung

- Versand innerhalb Deutschlands (Versandpauschale EUR 4,50)
- Selbstabholer (versandkostenfrei)
- Nachnahme innerhalb Deutschlands (Versandpauschale EUR 8,50)

Zahlungsweise

- Rechnung
- Barzahlung
- Nachnahme

Payment

Auch wenn dies nicht der oben beschriebenen Shop-Logik entspricht, definieren Sie zunächst drei grundsätzlich mögliche Zahlungsweisen im Setup des TS-Template:

```
#### Payment ###

plugin.tt_products.payment >

plugin.tt_products.payment {

    radio = 0

    TAXpercentage = 7

    10.title = Rechnung

    20.title = Barzahlung

    30.title = Nachnahme
}
```

Diese Optionen (oder eben nur eine Auswahl) werden in einer Combobox auf der Warenkorbseite unter **Zahlungsweise** zur Auswahl angezeigt. Wenn Sie anstelle der Comboboxen Radiobuttons bevorzugen, setzen Sie `radio = 1`.

Jede dieser Optionen wird nun erweitert bzw. genauer definiert:

```
## Shipping config:
plugin.tt_products.shipping {

    #radio = 0
    TAXpercentage = 19

    10.title = Versand innerhalb Deutschlands
    10.price = 4.50
    10.TAXincluded = 1
    # Über Exclude wird die Auswahl der Zahlungsmethoden eingeschränkt.
    # Wer Versand innerhalb Deutschland auswählt, kann nicht
    # Selbstabholer (hier 20) auswählen, hier geht nur Rechnung
    10.excludePayment = 20, 30

    20.title = Selbstabholer
    20.price =
    20.TAXincluded = 1
    # Über Exclude wird die Auswahl der Zahlungsmethoden eingeschränkt.
    # Wer Selbstabholer ist, kann nicht per Nachnahme (hier 30) oder
    # per Rechnung (hier 10) auswählen, hier funktioniert nur
    # Barzahlung
    20.excludePayment = 10, 30

    30.title = Nachnahme innerhalb Deutschlands
    30.price = 8.50
    30.TAXincluded = 1
    # Über Exclude wird die Auswahl der Zahlungsmethoden eingeschränkt.
    # Wer Nachnahme auswählt, kann nicht Rechnung (hier 10) oder
```

```
# Selbstabholer (hier 20) auswählen, hier funktioniert nur
# Nachnahme
30.excludePayment = 10, 20
}
```

Der MwSt.-Satz für die Versandkosten (TAXpercentage), der nicht mit dem MwSt.-Satz der bestellten Ware zu verwechseln ist, wird hier zunächst bei sämtlichen Liefermethoden auf den in Deutschland üblichen Satz von 19 gesetzt.

Jede Methode ist durch eine Zahl (10, 20, 30) eindeutig ansprechbar; der title, z. B. Versand innerhalb Deutschlands, liefert die Beschreibung, die so auch für den Kunden in der Auswahlbox erscheint. Sie sollte darum kurz und aussagekräftig sein. Zudem wird jeder dieser Methoden eine fixe Versandkostenpauschale zugewiesen, im Falle der Lieferung innerhalb Deutschlands EUR 4,50.³ Die Pauschale ist ein Bruttobetrag (price), weil TAXincluded auf 1 gesetzt ist.

Wie oben geplant, wurden drei Versandoptionen definiert, aus denen der Kunde eine auszuwählen hat. Abhängig davon sind sowohl die ausgewiesene Höhe der Versandkosten (die hier der Einfachheit halber pauschal angegeben und nicht dynamisch an den Umfang der konkreten Bestellung angepasst ist) wie auch die Zahlungsmöglichkeiten. excludePayment schließt also jeweils die Optionen aus, die beim jeweiligen Lieferweg nicht zur Verfügung stehen und darum im Warenkorb auch nicht als Radiobutton oder in der Combobox erscheinen.

2.2 HTML-Templates

Bis zu diesem Zeitpunkt bezieht der Shop Informationen zur Darstellung ausschließlich aus dem Default-HTML-Template der Extension tt_products, dem Template example_template_bill_de.tpl, auch „BananaGuard-Template“ genannt. Da Sie aber auch darauf Einfluss nehmen sollten, installieren Sie nun das auf der Webseite von Open Source Press bereitgestellte HTML-Template,⁴ das Sie im Laufe der weiteren Arbeit sukzessive an die Darstellung anpassen werden.

Laden Sie das Template tt_products_css.html sowie die zugehörige CSS-Datei tt_products.css herunter und speichern Sie beide in geeigneten Ordnern im Fileadmin-Verzeichnis Ihrer TYPO3-Installation – beispielsweise in den Ordnern templates bzw. styles. Die Art der Einbindung die-

³ Sie können grundsätzlich wählen, ob Sie den Dezimalpunkt oder das in Deutschland übliche Komma verwenden. Allerdings akzeptiert PayPal (vgl. S. 151) derzeit kein Dezimalkomma bei der Kreditkartenzahlung, weshalb im Code der Punkt empfohlen ist.

⁴ Unter: <http://www.opensourcepress.de/fileadmin/osp/download/typo3webshop.zip>

ser Dateien hängt nun von der Anzahl und damit von der grundsätzlichen Verwaltung von CSS-Dateien in Ihrem System ab; ob es nun eine zentrale Datei gibt, über die per `include url` verteilte Style-Vorgaben verfügbar gemacht werden, oder ob diese Dateien jeweils einzeln über Constants und Setup bereitgestellt werden, sei hier nicht weiter diskutiert. Weisen Sie der Einfachheit halber das HTML-Template in den Constants des Extension-Template `+tt_products` zu, und zwar mit folgender Anweisung:

```
##### Contants +tt_products Shop #####

### Template Datei ###

plugin.tt_products{
    file.templateFile = fileadmin/templates/tt_products_css.html
}
```

Die CSS-Datei fügen Sie über das TS-Setup der Extension ein, und zwar gleich zu Beginn. Damit nicht auf das Default-CSS der Extension selbst zugegriffen wird, sollten Sie dieses, wie im Codebeispiel aufgeführt, entfernen.

```
### Setup +tt_products Shop ###

#### Default PAGE object:
page = PAGE
page.typeNum = 0
page.stylesheet = fileadmin/styles/tt_products.css

#### Default CSS entfernen

plugin.tt_products._CSS_DEFAULT_STYLE =
```

Nun können Sie im HTML-Template beispielsweise den Liefer- und Zahlungsmethoden jeweils einen Kommentar, also einen Hinweistext für den Kunden zuweisen und damit den Shop ein wenig Ihren Vorstellungen anpassen.

Hier geht es zunächst nur um das Verfahren; konkrete Beispiele finden Sie bei den jeweiligen Bearbeitungsschritten des Shops in den folgenden Kapiteln. Hier also lediglich ein Beispiel, in dem die Hinweise zum Versand geändert werden. Laden Sie zunächst das Template in einen Editor und suchen Sie den Bereich `MESSAGE_SHIPPING_10`:

```
<!-- ###MESSAGE_SHIPPING_10### -->
<p>Die Ware wird per Postpaket versendet.<br />
Bitte überweisen Sie den Rechnungsbetrag nach Erhalt der
Rechnung auf unser Konto.
</br>
</p>
<!-- ###MESSAGE_SHIPPING_10### -->
```

Sie sehen, dass auch hier die Nummerierung der Optionen wie im TS-Setup zum Tragen kommt. Achten Sie darauf, dass diese Zahlen immer mit der entsprechenden Zahlungs- oder Versandmethode übereinstimmen. Die gewünschten Kommentare sollten in ein <p>- Tag eingeschlossen sein.

In der ersten Combobox bestimmen Sie die Versandoption, aus der sich dann die möglichen Zahlungsweisen ergeben. Darunter finden sich jeweils die im HTML-Template eingetragenen Hinweise (Abbildung 2.3).

Abbildung 2.3:
Warenkorb:
Lieferung und
Bezahlung

02.07.2007 - 11:46:16 :: Print

| Produkte: | Einzelpreis: | Anzahl: | Gesamt in Euro: |
|-----------------|--------------|--------------------------------|-------------------|
| Ein Testprodukt | 39.90 | <input type="text" value="1"/> | inkl. MWSt. 39.90 |

Versand und Verpackung:

Pauschale:
inkl. Versandkosten

Versand innerhalb Deutschlands inkl. MWSt. 4.50

Zahlungsweise: Rechnung

Rechnung inkl. MWSt. 0.00

Die Ware wird per Postpaket versendet.

Bitte überweisen Sie den Rechnungsbetrag nach Erhalt der Rechnung auf unser Konto.

Gesamtbetrag: **inkl. MWSt. 44.40**
(exkl. MWSt. 41.07)
(MWSt. 3.33)

Checkliste

- Die Zahlenwerte in Payment und Shipping stimmen nicht überein.
- Die Zahlenwerte zu Message Payment/Shipping im HTML-Template sind nicht korrekt.
- Wenn Sie eine ältere Version von tt_products einsetzen, müssen Sie im Default-Setup der Extension die dortigen Angaben für Shipping und Payment auskommentieren, da sie nicht durch Ihr Extension-Template überschrieben werden.
- Falsche oder keine PID gesetzt.

2.2.1 Adressdaten

Nach Bestätigung des Warenkorbs durch den Kunden erscheint im Frontend die Seite zur Eingabe der Adressdaten.⁵ Bei der Adresseingabe sind auch getrennte Rechnungs- und Lieferanschrift möglich (Abbildung 2.4).

Rechnungsanschrift:

Bitte geben Sie Ihre Postanschrift in diesem Formular an. Der angegebene Name und die Adresse werde auf der Rechnung der Bestellung erscheinen. Sie können auch alternativ eine abweichende Lieferadresse in den unteren Feldern angeben.

Anrede:

Name: *

Adresse: *

PLZ: *

Stadt: *

Land: *

Telefon: *

Email: *

Hinweis: Felder mit * sind Pflichtfelder.
 Ihre Daten werden nicht an Dritte weitergegeben!

Abbildung 2.4:
 Warenkorb:
 Adressdaten
 eingeben
 (Ausschnitt)

Alternative Lieferadresse:

Wenn Ihre Bestellung zu einer anderen Anschrift geliefert werden soll, so geben Sie hier bitte die vollständige Adresse an.

Anrede:

Auch hier ändern/definieren Sie die Texte über das HTML-Template, und zwar im Bereich `BASKET_INFO_TEMPLATE`. Die Länderauswahl wird in diesem Fall bereits aus der Tabelle `static_info_tables` übernommen. Weitere Informationen darüber finden Sie im Kapitel 3.

⁵ Wir gehen hier vorläufig davon aus, dass noch keine Benutzerverwaltung im System implementiert ist. Dieses wichtige Thema wird in Kapitel 3 ab Seite 49 behandelt, obwohl es sich dabei um ein allgemeines TYPO3-, nicht um ein Shop-spezifisches Feature handelt.

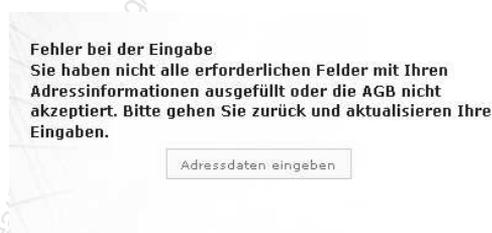
Die als Pflichtfelder markierten Adressfelder müssen im Setup oder den Constants des Shop-Template definiert werden. Die jeweiligen Feldnamen finden Sie im HTML-Template im Bereich BASKET_INFO_TEMPLATE.

```
##### Pflichtfelder ausfüllen
```

```
plugin.tt_products.requiredInfoFields = name, address, city, zip, email
```

Füllt ein Kunde die Pflichtfelder nicht aus, wird im HTML-Template das Template BASKET_REQUIRED_INFO_MISSING aktiviert und der Shop-Besucher aufgefordert, die Daten zu vervollständigen. Eine Plausibilitätsprüfung findet allerdings nicht statt.

Abbildung 2.5:
Aufforderung, die
Daten zu
vervollständigen



Auch hier lässt sich der Text im HTML-Template ändern. Laden Sie zunächst das Template in einen Editor und suchen Sie den Bereich BASKET_REQUIRED_INFO_MISSING. Ändern Sie beispielsweise die Hinweise innerhalb der Klasse `<div class="BASKET_REQUIRED_INFO_MISSING">`:

```
<div class="BASKET_REQUIRED_INFO_MISSING">  
  Fehler bei der Eingabe<br />  
  <b>Sie haben nicht alle erforderlichen Felder mit Ihren  
  Adressinformationen ausgefüllt oder die AGB nicht akzeptiert.  
  Bitte gehen Sie zurück und aktualisieren Ihre Eingaben.</b>  
</div>  
<form method="post" action="###FORM_URL###">  
  <input type="button" id="button2_BASKET_REQUIRED_INFO_MISSING"  
  name="products_info" value="Adressdaten eingeben"  
  onclick="document.forms[0].action='###FORM_URL_INFO###';">  
</form>
```

Nach Eingabe der Adressdaten und Akzeptieren der AGB (s. u.) schließt der Kunde die Bestellung ab.

2.2.2 AGB

Wer in Deutschland einen Webshop betreibt, muss die AGB offen legen. Diese sollten Sie mit Ihrem Anwalt abstimmen, um unnötigen und möglicherweise kostspieligen Ärger zu vermeiden. Wir werden uns hier allerdings

ausschließlich um die technischen Aspekte im Zusammenhang mit der Einbindung der AGB in den Bestellvorgang kümmern.

Erstellen Sie zunächst eine Seite im Seitenbaum, die Sie auch im Menü anzeigen lassen. Fügen Sie Ihre AGB als Inhaltselement in diese Seite ein.

Im Warenkorb muss der Kunde die AGB akzeptieren, sonst kann er nicht einkaufen. Diese Funktionalität ist im Bereich `BASKET_INFO_TEMPLATE` des HTML-Template definiert.

Wichtig ist die „Verknüpfung“ der AGB-Seite mit der Auswahlfunktion im Warenkorb; diese erfolgt, wie bereits auf Seite 33 beschrieben, über eine entsprechende Zuweisung der PID. Ermitteln Sie also per Mauszeiger die PID der AGB-Seite und weisen Sie diesen Wert über den Constant Editor (`PID of AGB page`) oder direkt über die Constants der Variable `plugin.tt_products.PIDagb` zu.



Abbildung 2.6:
PID der AGB
zuweisen

Damit haben Sie Folgendes erreicht: Die Seite mit Ihren AGB kann wie alle anderen regulären Seiten Ihres Webauftritts jederzeit über das Menü aufgerufen werden und hat damit zunächst einmal nichts mit dem Shop zu tun. Erst mit der PID-Zuweisung in den Constants binden Sie diese Seite in Ihren Shop so ein, dass der Link vom `BASKET_INFO_TEMPLATE` auf die AGB-Seite gesetzt wird, der Shop-Besucher also die AGB auch direkt aus dem Shop heraus aufrufen und lesen kann. Zudem muss er auf der Shop-seite ein Häkchen setzen, dass er die AGB akzeptiert; nur so kann er im Bestellvorgang fortfahren. Setzt er das Häkchen nicht, erscheint eine Meldung, dass die Angaben unvollständig sind.

Da die AGB-Seite eine reguläre TYPO3- und keine Shop-spezifische Seite ist, gibt es von dort auch keinen Link zurück in den Shop. Das ist grundsätzlich auch kein Problem, da die AGB in einem eigenen Fenster angezeigt werden und der Warenkorb weiterhin aktuell bleibt.

Wenn Sie allerdings kein eigenes Fenster öffnen lassen möchten, müssen Sie anders vorgehen, und zwar mit Hilfe eines JavaScript-Befehls, den Sie als HTML-Element auf der AGB-Seite platzieren:

```
<a href="javascript:history.back()">zurück</a>
```

Darüber hinaus können Sie im HTML-Template das Target des Links ändern, um das Öffnen eines eigenen Fensters zu verhindern. Ein Link bringt Sie damit von der AGB- zur Warenkorb-Seite zurück, von der aus sie aufgerufen wurde.

2.2.3 Bestelldaten zusammenfassen

Indem der Kunde die AGB akzeptiert und den Bestellvorgang fortsetzt, werden die Eingaben auf einer eigenen Seite zusammengefasst. Im HTML-Template ist das BASKET_PAYMENT_TEMPLATE für diese Ansicht zuständig.

Abbildung 2.7:
Ansicht der
zusammengefassten
Adress- und
Lieferdaten im
Frontend

| | | |
|---|-------------------------------|----------------|
| Ihre Bestellung: | | |
| Produkte: | Menge: | Gesamt: |
| Ein Testprodukt | 1 | 39.90 |
| Rechnungsanschrift: | | |
| Herr | | |
| Testname | | |
| Teststr. 4 | | |
| 11111 Teststadt | | |
| Germany | | |
| 11111111 | | |
| yourname@mytypo3domain.de | | |
| (An diese Email-Adresse wird die Bestätigung Ihrer Bestellung geschickt!) | | |
| Lieferanschrift: | | |
| Testname2 | | |
| Teststr. 44 | | |
| 11111 Teststadt | | |
| 22222222 | | |
| yourname@mytypo3domain.de | | |
| (An diese Email-Adresse wird die Bestätigung Ihrer Bestellung geschickt!) | | |
| gewünschtes Lieferdatum: | | |
| Kommentar: | | |
| Versand und Verpackung: | | |
| inkl. Versandkosten | 4.50 | |
| Zahlungsweise: | | |
| Rechnung | 0.00 | |
| Gesamtbetrag in EURO: | 44.40 (exkl. MwSt.: 41.07) | |

2.2.4 Bestellung abschließen

Mit einer letzten Bestätigung ist der Kaufvorgang von Kundenseite aus abgeschlossen, und im HTML-Template wird das `BASKET_ORDERCONFIRMATION_TEMPLATE` aktiv, das die Bestellung auslöst, eine Bestellnummer ver gibt und alle Daten nochmals zusammenfasst.

Vielen Dank für Ihre Bestellung!

Ihre Bestellung ist nun abgeschlossen und hat folgende Nummer:

258

Die Bestellbestätigung wird ebenfalls an die Emailadresse `tester@mytypo3domain.de` geschickt.

Zahlungsart:

Überweisung:
Die Ware wird nach Eingang Ihrer Zahlung auf unserem Konto versendet. Bitte überweisen Sie den Rechnungsbetrag auf folgendes Konto:

Empfänger:
Bankinstitut:
Konto Nr.:
Bankleitzahl:
Angabe Verwendungszweck: Vollständiger Name + komplette Adresse

IBAN:
BIC:

Betreff: Ihre Bestellnummer 258 unbedingt angeben!

Liefer- und Bestellstatus

Die Ware bleibt bis zur vollständigen Bezahlung Eigentum des Verlages.

Ihr Bestelldatum: 14-10-2007.

Klicken Sie hier um den Status jetzt zu sehen.

| Produkte: | Menge: | Gesamt: |
|----------------------------|--------|--|
| Ein Testprodukt: Hardcover | 1 | inkl. MwSt. 34.90 (exkl. MwSt. 32.62) |

Rechnungsanschrift:

tester tester
Teststr. 2
tester@mytypo3domain.de
(An diese Email-Adresse wird die Bestätigung Ihrer Bestellung geschickt!)

Username: tester User ID: 2132

Lieferanschrift:

tester tester
Teststr. 2
11111 Teststadt
Germany
tester@mytypo3domain.de
(An diese Email-Adresse wird die Bestätigung Ihrer Bestellung geschickt!)

gewünschtes Lieferdatum:
Kommentar:

Versand und Verpackung:

| | |
|------------------------------------|--|
| Pauschale: Versand nach Gewicht | inkl. MWST. 4.50 (exkl. MwSt. 4.50) |
| Zahlungsweise: | |
| Rechnung | inkl. MWST. 0.00 (exkl. MwSt. 0.00) |
| Gesamtbetrag in EURO: | inkl. MwSt. 39.40 (exkl. MwSt. 37.12) |

Abbildung 2.8:
Bestellung
abschließen

Die Texte können Sie wie gewohnt im HTML-Template editieren. Suchen Sie die entsprechende Stelle und ändern Sie bei Bedarf den Text:

```
<div id="text_agb_BASKETINFO">
  ###GW1B### Sie bestätigen hiermit, dass Sie
  die AGB gelesen haben und diese akzeptieren.<br /><br />
  ###GW1E###
</div>
<div id="floatbox_BASKETINFO4">
  <input type="checkbox" name="recs[personinfo][agb]"
```

```
value="###PERSON_AGB###" id="checkboxagb"/>
<div id="text_best_agb_BASKETINFO">
  ###GW1B###<!--###LINK_AGB###-->Die Allgemeinen
  Geschäftsbedingungen <!--###LINK_AGB###--> erkenne ich an.
</div>
</div>
<div id="SUBMIT_BASKETINFO">
  <input type="submit" id="button5" name="products_update"
  value="zurück zum Warenkorb" />
  <input type="submit" id="button4" name="products_payment"
  value="ABG akzeptiert"
  onclick="document.forms[0].action='###FORM_URL_FINALIZE###';">
  <!--<input type="submit" id="button4" name="products_payment"
  value="ABGs ansehen" />-->
</div>
<!--</div>-->
<!-- ###BASKET_INFO_TEMPLATE### end -->
```

2.2.5 E-Mail-Benachrichtigung

Sobald der Shop-Besucher seine Bestellung abgeschlossen hat, wird eine E-Mail an den Kunden generiert, die die Angaben zusammenfasst und bestätigt. Es ist auch möglich, die E-Mail statt an die Rechnungsadresse an die vom Kunden genannte Lieferadresse zu senden. Aktivieren Sie dazu im Template `+tt_products` mit dem Constant Editor **send email to delivery**.

Damit die E-Mails korrekt versendet werden, sind noch einige Einstellungen in den Constants bzw. im Constant Editor notwendig. Wenn Sie HTML-E-Mails versenden möchten – was von einigen Kunden allerdings gar nicht gerne gesehen wird! –, aktivieren Sie im Constant Editor **HTML email**.

Die Betreffzeile füllen Sie über **subject of email**: Tragen Sie den gewünschten Text ein. Füllen Sie das Feld nicht aus, wird die erste Zeile des E-Mail-Textes gesetzt. Der Absender des Shops wird über **from email name** bestimmt, die E-Mail-Adresse des oder der Shop-Administratoren über **to email addresses**. Mehrere Empfänger werden durch Komma getrennt.

In den Constants erscheinen die Angaben dann wie folgt:

```
## PID der AGB - Seite
plugin.tt_products.PIDagb = 241

## Email und Web
plugin.tt_products.orderEmail_subject = myCompany GmbH: Ihre Bestellung
plugin.tt_products.orderEmail_fromName = myCompany Shop
plugin.tt_products.orderEmail_to = shopmaster@mytypo3domain.de
plugin.tt_products.domain = www.mytypo3domain.de
plugin.tt_products.orderEmail_from = shopmaster@mytypo3domain.de
```

Die Texte der E-Mails zur Benachrichtigung werden im HTML-Template im Bereich `EMAIL_PLAINTEXT_TEMPLATE` editiert.

Sobald eine Bestellung versendet wird oder eine Änderung im Tracking des Shops stattgefunden hat, wird eine E-Mail-Benachrichtigung versendet, und zwar sowohl an den Kunden als auch an den Shop-Administrator. Diese E-Mails sind identisch. Sie können in `tt_products` auch unterschiedliche E-Mails versenden. Dazu fügen Sie einen Subpart `EMAIL_PLAINTEXT_TEMPLATE_SHOP` ein. Dieser wird dann für den E-Mail-Versand an den Shopbetreiber oder die E-Mail-Adressen in den Kategorien verwendet. Beachten Sie beim Test Ihres Shops, dass Ihre Kunden-E-Mail-Adresse und die des Administrators unter Umständen identisch sind, weil Sie für die Testbestellungen und den Administrator dieselbe Adresse bzw. dasselbe Konto benutzen. Damit sollten Sie korrekterweise jeweils zwei identische E-Mails erhalten.

Mit dem Eintrag in die Constants

```
orderEmail_htmlmail = 1
```

stellen Sie ein, ob eine HTML- oder eine reine Text-E-Mail versendet wird.

Die verschiedenen E-Mail-Benachrichtigungen

Je nach Aktion oder Statusänderung einer Bestellung lassen sich sowohl an den Administrator wie auch den Kunden verschiedene Nachrichten per E-Mail versenden:

Bestellbestätigung

an den Shop-Administrator und den Kunden. Die Bestellbestätigung `BASKET_ORDERCONFIRMATION_TEMPLATE` wird für HTML-E-Mails verwendet.

`EMAIL_PLAINTEXT_TEMPLATE` wird zusätzlich zur HTML-E-Mail mit versendet für den Fall, dass der Empfänger kein HTML lesen kann oder will.

Sofern der Subpart `EMAIL_PLAINTEXT_TEMPLATE_SHOP` im HTML-Template steht, wird dieser für die E-Mail an den Shopbetreiber verwendet.

Bestellverwaltung

Der Kunde kann Nachrichten an den Shopbetreiber versenden und umgekehrt. Bei jeder Änderung des Bestellstatus wird eine solche E-Mail an beide versendet (`TRACKING_EMAILNOTIFY_TEMPLATE`).

neu angelegter Benutzer

Bei der ersten Bestellung eines Kunden kann automatisch ein Benutzerkonto angelegt werden. Der Kunde erhält dann eine E-Mail mit

den Zugangsdaten (EMAIL_NEWUSER_TEMPLATE). Dies gilt nur, wenn die Registrierung über tt_products vorgenommen wird.

Warnung über die Reduktion des Warenbestandes

Sofern aktiviert, kann der Shopadministrator eine E-Mail erhalten, wenn der Warenbestand eines Produkts auf eine bestimmte Anzahl reduziert wird.

AGB als E-Mail-Anhang

Wie bereits erwähnt, sind die AGB in Deutschland besonders wichtig. Es ist darum gängige Praxis, die Geschäftsbedingungen an jede Bestellung anzuhängen. Um die AGB also als Anhang zu versenden, binden Sie die Datei genauso wie Ihr HTML-Template über die Constants des Extension-Template ein.

```
plugin.tt_products{
    file.templateFile = fileadmin/osp/templates/tt_products_css.html
    AGBAttachment = fileadmin/osp/templats/tt_products/agb.txt
}
```

Damit haben Sie einen ersten, funktionsfähigen Basis-Shop in deutscher Sprache aufgebaut. Dieses Gerüst werden Sie in den folgenden Kapiteln erweitern und verbessern; beginnen Sie mit einer komfortablen Benutzerverwaltung, die den Kunden ein Login auf der Webseite – nicht nur für den Shop – erlaubt und Ihnen unter anderem die lästige Aufgabe abnimmt, bei jeder Test-Bestellung erneut alle Kontaktdaten eingeben zu müssen.

3

Kapitel

Benutzerregistrierung

Soll Ihr Shop in Teilen oder grundsätzlich nur registrierten Benutzern zur Verfügung stehen, empfiehlt sich die Implementierung einer systemweiten Benutzerregistrierung sowie eines Login-Moduls auf Ihrer Webseite.

Zwar bietet auch tt_products selbst eine Registrierungsfunktion, diese erfolgt aber ausschließlich über den Einkauf im Shop und ist längst nicht so komfortabel wie die systemweite Frontendbenutzerregistrierung. Der Einsatz von tt_products setzt diese aber eben nicht voraus; möchten Sie die Adresseingabe und -verwaltung allein über tt_products abwickeln, ist lediglich der folgende Code in den Constants des Extension-Templates notwendig:

```
createUsers=1  
memberOfGroup=1  
PIDuserFolder=52
```

Wenn Sie jedoch Größeres vorhaben, sollten Sie die Installation – wie auch im Folgenden gezeigt – auf der Grundlage anderer Extensions vornehmen.

Die Verwendung eingeloggter Frontendbenutzer lohnt sich bereits für Ihre Testbestellungen, denn als angemeldeter Benutzer müssen Sie Ihre Adressdaten nur einmalig eingeben, was, gerade wenn Sie Ihr HTML-Template anpassen, eine Menge Zeit spart.

Im Folgenden wird die *New Login Box* von TYPO3-Erfinder Kasper Skårhøj mit der derzeit von Stanislas Rolland und Franz Holzinger entwickelten Frontendbenutzerregistrierung kombiniert. Wir haben uns für diese Extensions entschieden, weil sie nach wie vor gepflegt werden und deren Entwickler erreichbar sind – das ist eben nicht bei allen Extensions gewährleistet. Außerdem gehören sie zu den am häufigsten heruntergeladenen Erweiterungen im *TYPO3 Extension Repository* (TER) überhaupt.

Einige Funktionen von `tt_products` stehen übrigens nur angemeldeten Benutzern zur Verfügung, zum Beispiel die Merkliste, Preise zu Händlerkonditionen oder das Einlösen von Gutscheinen; diese Funktionen werden im weiteren Verlauf des Buches noch ausführlicher beschrieben.

3.1 New Login Box

Installieren Sie also zunächst das Login-Modul *New Login Box*. Wechseln Sie in den Extension Manager, wählen Sie dort **Import extensions** und tragen Sie in das Suchfeld den Extension Key `newloginbox` ein. Klicken Sie auf den Button **Look up** und installieren Sie die Extension (hier Version 3.0.1) in gewohnter Weise. Grundsätzlich gibt es zwei Wege, die New Login Box einzurichten: Als Plugin auf einer Seite oder über einen Marker im Setup, damit sie auf allen Seiten Ihres Internetauftritts erscheint.

3.1.1 Installation über Marker

Um die Extension über Marker einzufügen, gehen Sie folgendermaßen vor: Wechseln Sie in das HTML-Basistemplate Ihres Internetauftritts und fügen Sie einen Marker `LOGIN` ein.

In Ihrem Root-Template weisen Sie den Marker wie gewohnt im Markerbereich zu:

```
LOGIN < plugin.tx_newloginbox_pi1
```

Erstellen Sie nun ein Extension-Template namens `+loginbox` in Ihrem Template-Ordner (Webmodul: `Liste | Neuen Datensatz anlegen | Template`). Wenn Sie nicht mit verteilten Templates arbeiten, fügen Sie den Bereich für die New Login Box in das TS-Template Ihrer Rootseite ein. Ergänzen Sie den unten aufgeführten Code in Ihrem TS-Setup. Für mehrsprachige Seiten legen Sie die Übersetzungen fest, die Sie in dem Code-Beispiel ebenfalls fin-

den.¹ Sie müssen zumindest die Übersetzung für die genutzte Seitensprache ins Setup einpflegen, da hier alle Anpassungen vorgenommen werden, nicht etwa im HTML-Template der Extension. Das gilt jedoch nur für die Installation über Marker, da die Texte im Plugin (s. u.) über die Flexforms definiert werden.

Im Beispiel wurde auf die Überschriften und Hinweistexte der New Login Box verzichtet. Lediglich Benutzername, Passwort und ein Submit-Button sollen vor dem Login angezeigt werden. Dem eingeloggten Benutzer wird eine kurze Erfolgsmeldung (Abbildung 3.1) angezeigt.

Anmeldung erfolgreich

Sie sind
angemeldet als
tester

Abbildung 3.1:
Die Erfolgsmeldung

Schlägt die Anmeldung fehl, erscheint ebenfalls ein entsprechender Hinweis (Abbildung 3.2).

Anmeldefehler

Bitte versuchen
Sie es erneut

Benutzername:

Passwort:

Login

Abbildung 3.2:
Die Fehlermeldung

Um die New Login Box zu positionieren, wurde eine CSS-Datei erstellt, eingebunden und entsprechend angepasst (vgl. dazu das ab Seite 39 beschriebene Vorgehen).

Der folgende Code zeigt das Setup des Extension-Template:

¹ Weitere Informationen zur Mehrsprachigkeit finden Sie in Abschnitt 4.5 ab Seite 85.

```
# New Login Box über Marker

# Nur für die Installation über Marker. Adernfalls auskommentieren.
marks.LOGIN < plugin.tx_newloginbox_pil

plugin.tx_newloginbox_pil.templateFile =
    fileadmin/templates/newloginbox.html

plugin.tx_newloginbox_pil.email_from = webmaster@mytypo3domain.de
plugin.tx_newloginbox_pil.email_fromName = Webmaster
plugin.tx_newloginbox_pil.replyTo = webmaster@mytypo3domain.de

# Mehrsprachigkeit
plugin.tx_newloginbox_pil {

    _LOCAL_LANG {
    default {
        oLabel_header_welcome =
        oLabel_msg_welcome =
        oLabel_header_logout =
        oLabel_msg_logout =
        oLabel_header_error = Failure
        oLabel_msg_error = Please try again
        oLabel_header_success = Success
        oLabel_msg_success = You are logged in named ###USER###
        oLabel_header_status =
        oLabel_msg_status =
        username = username:
        password = password:
        login = login
        logout = logout
        send_password = send password
        your_email = your email-address:
        forgot_password = forgot your password?
    }

    de
        {
            oLabel_header_welcome =
            oLabel_msg_welcome =
            oLabel_header_logout =
            oLabel_msg_logout =
            oLabel_header_error = Anmeldefehler
            oLabel_msg_error = Bitte versuchen Sie es erneut
            oLabel_header_success = Anmeldung erfolgreich
            oLabel_msg_success = Sie sind angemeldet als ###USER###
            oLabel_header_status =
            oLabel_msg_status =
            username = Benutzername:
            password = Passwort:
            login = Login
```

```

logout = Abmelden
send_password = Passwort senden
your_email = Ihre Email-Adresse:
forgot_password = Passwort vergessen?
}
}
}

##### Default CSS entfernen #####

plugin.tx_newloginbox_pi1._CSS_DEFAULT_STYLE =

```

Arbeiten Sie mit verteilten Templates, binden Sie das neue TS-Template über `Include basis template` in das Haupttemplate Ihrer Seite ein, wie bereits in Abschnitt 1.3 ab Seite 23 beschrieben.

3.1.2 Installation als Plugin

Wenn Sie die Extension als Plugin auf lediglich einer Seite Ihres Internetauftritts einfügen möchten, erstellen Sie, wie oben beschrieben, ein Extension-Template, allerdings ohne den Bereich für den Marker im HTML-Template zu definieren. Kommentieren Sie diesen Bereich aus dem Setup einfach aus.

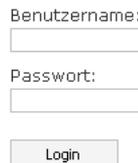
Abbildung 3.3:
Flexforms der New
Login Box

Fügen Sie nun das Plugin `Besseres Anmeldeformular` als Seiteninhalts-element auf der gewünschten Seite des Webauftritts ein, also beispielsweise auf der Startseite.

Über die Flexforms der Seite können Sie nun die Einträge für die Willkommensnachricht, die Nachricht bei fehlgeschlagenem Login etc. vornehmen, wie sie auch bei der Markervariante im TS-Setup angelegt wurden. Der Link **Passwort vergessen?** erzeugt ein Eingabefeld für die E-Mail-Adresse, in das der Benutzer seine E-Mail-Adresse einträgt und daraufhin je nach Konfiguration automatisiert sein bei der Registrierung hinterlegtes Passwort zugesendet bekommt oder ein neues erzeugt wird.²

Damit steht im Frontend die Login Box zur Verfügung, die allerdings in diesem Beispiel nur in Verbindung mit der Frontendbenutzerregistrierung sinnvoll einzusetzen ist – schließlich sollen die Registrierungsdaten der Benutzer ja auch in irgendeiner Weise verwaltet werden.

Abbildung 3.4:
New Login Box



Benutzername:

Passwort:

Aus diesem Grunde lassen Sie die zur New Login Box gehörende Listenfunktion beiseite und wenden sich sogleich den umfangreichen Funktionen der zweiten in diesem Zusammenhang wichtigen Extension zu.

3.2 Frontendbenutzerregistrierung

Installieren Sie über den Extension Manager die Extension `sr_feuser_register` nach dem schon bekannten Verfahren, für den Beispielshop also `sr_feuser_register 2.5.7`.³

Danach installieren Sie über den Extension Manager die Sprachen, in denen Sie die Extension betreiben wollen. Weitere Informationen über das Handling der Extension-Sprache finden Sie im Anhang B ab Seite 264.

Nach der Installation fügen Sie das Plugin auf der gewünschten Seite z. B. oberhalb der Login Box ein: **Neuen Datensatz anlegen | Seiteninhalt | Plugins | Allgemeines Plugin**. Bearbeiten Sie diesen und wählen Sie unter **Erweiterungen** die Option **Front End Benutzer Registrierung (Anzeigetyp Normal)**; das Plugin steht damit zur Verfügung, muss aber noch in einen größeren Zusammenhang gestellt werden.

² Ein deutsches Tutorial zur New Login Box finden Sie im TER unter dem Namen `tut_newloginbox_de`.

³ Falls Sie die Frontendbenutzerregistrierung ohne die Shop-Installation vornehmen, werden für die Installation die Extensions Static Info Tables und FE BE Library benötigt.

3.2.1 SysOrdner und Benutzergruppen

Erstellen Sie zunächst einen leeren SysOrdner namens Frontendbenutzer, in dem später die Frontendbenutzer(-Gruppen) verwaltet werden.

Benutzergruppen anlegen

Legen Sie in dem soeben erstellten SysOrdner zunächst zwei FE-Benutzergruppen an (Neuen Datensatz anlegen | Website Benutzergruppe): eine für Benutzer, die ihre Registrierung noch nicht bestätigt haben, eine weitere Gruppe für Benutzer, deren Registrierung abgeschlossen ist (Abbildung 3.5). Merken oder notieren Sie sich die UIDs dieser Gruppen und jene des SysOrdners, da Sie diese später über die Constants zuweisen müssen.

Abbildung 3.5:
Benutzergruppe
erstellen

Über das Webmodul Liste können Sie die Gruppen und später auch die Benutzer jederzeit editieren. Über das Webmodul Liste lassen sich auch die UIDs der Gruppen sichtbar machen (Abbildung 3.6).

| | [Ref] | [uid] | [pid] |
|-----------------------|-------|-------|-------|
| Registrierte Benutzer | | 1 | 8 52 |
| Unregistriert | | 9 | 52 |

Abbildung 3.6:
Benutzergruppen in
der Ansicht
Webmodul: Liste

Klicken Sie auf das Plus-Zeichen hinter der Überschrift **Website Benutzergruppe**, erscheint eine Auswahl, aus der Sie Informationsfelder wählen können, u. a. [uid]; indem Sie dieses Feld setzen, erscheint die jeweilige Angabe zur Gruppe. Setzen Sie, falls noch nicht geschehen, den Haken bei **Erweiterte Ansicht**, erscheinen diese Angaben permanent.

3.2.2 Seiten erstellen und Plugins einfügen

Es wird hier nun eine einfache Version der Benutzerregistrierung auf der Basis einer Installation via Plugin implementiert, nachdem, wie oben beschrieben, ein erstes Plugin für den Link zur Registrierung auf der Startseite eingefügt worden ist. Als Ausgangspunkt des Plugins müssen Sie den soeben erstellten SysOrdner mit den Benutzergruppen eintragen (unter **Allgemeine Datensatzsammlung**).⁴

Abbildung 3.7:
Plugin einfügen

Erstellen Sie nun die benötigten Seiten für die Benutzerregistrierung, zum Beispiel als erste Unterseiten Ihrer Startseite. Bei der ersten Seite (**Anlegen**) setzen Sie den Haken bei **Im Menu verstecken**, fügen das Plugin als Inhaltselement auf dieser Seite ein, wählen unter **Erweiterung** den Punkt **Front End Benutzer Registrierung** und als **Anzeigetyp** **Anlegen**. Der SysOrdner mit den beiden zuvor angelegten Benutzergruppen ist der Ausgangspunkt.

Im nächsten Schritt erstellen Sie auf derselben Ebene eine weitere Seite, vergeben den Namen **Editieren** und setzen den Haken bei **Im Menu verstecken**. Fügen Sie das Plugin auch auf dieser Seite ein und wählen Sie als

⁴ Wenn Sie die New Login Box über Marker erstellt haben, setzen Sie den SysOrdner mit den Frontendbenutzern für die Login Box auf der Rootseite Ihrer Internetpräsenz als Ausgangspunkt – und zwar auf der Seite selbst, nicht im Content – und vererben ihn damit allen Seiten.

Anzeigetyp **Bearbeiten**. Auch hier ist der SysOrdner mit den beiden Benutzergruppen der Ausgangspunkt.

Die dritte Seite, nach demselben Muster angelegt, dient der Bestätigung der Registrierung. Vergeben Sie hier den Namen **Bestätigung** und fügen Sie ein weiteres Plugin vom Anzeigetyp **Normal** ein. Wieder ist der SysOrdner mit den beiden Benutzergruppen der Ausgangspunkt.

Damit sind alle notwendigen Seiten angelegt, und Ihr Seitenbaum für die Benutzerregistrierung könnte wie in Abbildung 3.8 aussehen.



Abbildung 3.8:
Seitenbaum der
Frontendbenutzer-
registrierung

Im letzten Schritt müssen Sie nun ein Extension-Template `+feuser` im Template-SysOrdner anlegen. Wenn Sie nicht mit verteilten Templates arbeiten, fügen Sie den Bereich für die Frontendbenutzerregistrierung in das TS-Template Ihrer Webpräsenz ein.

3.2.3 TS-Template zur Benutzerregistrierung

Wechseln Sie nun in Ihren Template-Ordner und legen Sie ein Extension-Template `+feuser` zur Frontendbenutzerregistrierung an.

Constants

Ergänzen Sie folgenden Code in den Constants:

```
## HTML-Template zuweisen
plugin.tx_srfеuserregister_pil.templateFile =
    fileadmin/templates/feuser.html

plugin.tx_srfеuserregister_pil.requiredFields =
    username, password, first_name, last_name, company, email,
    city, zip, address, telephone

## E-Mail-Adresse des Webmasters, der für die Benutzerregistrierung
## zuständig ist und der die Mails versendet und erhält
plugin.tx_srfеuserregister_pil.email = webmaster@mytypo3domain.de

## Formularfelder der Registrierung
plugin.tx_srfеuserregister_pil.formFields = username, password, gender,
```

```
first_name, last_name, status, email, usergroup, address, city,
zone, static_info_country, zip, telephone, company,
module_sys_dmail_category, comments, disable

## Domain der Webseite
plugin.tx_srfeuserregister_pil.siteName = mytypo3domain.de

## Ersetzen einiger Marker, z.B. in der E-Mail -- andernfalls würde der
## Titel auf der Seite sr_feuser_register Seite genommen
plugin.tx_srfeuserregister_pil.linkToPIDAddButton = 1

## Template für HTML-Mails -- falls solche versendet werden sollen,
## Kommentarzeichen entfernen
#plugin.tx_srfeuserregister_pil.HTMLMailCSS =
#     EXT:sr_feuser_register/pil/tx_srfeuserregister_htmlmail.css

## UID der Benutzergruppe nach abgeschlossener Registrierung
plugin.tx_srfeuserregister_pil.userGroupAfterConfirmation = 8

## UID der Benutzergruppe während des Registrierungsvorgangs
plugin.tx_srfeuserregister_pil.userGroupUponRegistration = 9

## Anzahl der hochzuladenden Bilder -- hier keins
plugin.tx_srfeuserregister_pil.maxImages = 0

## PID der Hauptseite, auf der das Plugin der Registrierung
## eingefügt ist
plugin.tx_srfeuserregister_pil.registerPID = 232

## PID der Seite mit dem Editierformular
plugin.tx_srfeuserregister_pil.editPID = 234

## PID der Login-Seite
plugin.tx_srfeuserregister_pil.loginPID = 232

## PID der Bestätigungsseite
plugin.tx_srfeuserregister_pil.confirmPID = 235

## PID der Seite, auf die gewechselt wird, nachdem der Benutzer sein
## Profil geändert und gespeichert hat.
plugin.tx_srfeuserregister_pil.linkToPID = 8

## PID des SysOrdnerns mit den Benutzergruppen und Benutzern
plugin.tx_srfeuserregister_pil.pid = 52

## Beim Ändern des Profils und in dessen Vorschau wird ein zusätzlicher
## Button 'Speichern' angezeigt. Bei Klick wird gespeichert und auf die
## Seite linkToPID gewechselt.
plugin.tx_srfeuserregister_pil.linkToPIDAddButton = 1

## Überschreibt den Titel des Folders der Frontendbenutzer -- wird als
## Marker verwendet.
```

```
plugin.tx_srfuserregister_pil.pidTitleOverride = mytypo3domain.de
```

```
## Mindestanzahl an Zeichen für Benutzernamen
plugin.tx_srfuserregister_pil.usernameAtLeast = 6
```

Ändern Sie alle Einträge entsprechend Ihrer Konfiguration; die Kommentare im Code sollten ausreichende Hinweise dazu geben.

Setup

Fehlt nun noch der Code im TS-Setup. Für mehrsprachige Seiten bestimmen Sie hier die Übersetzungen. Beachten Sie, dass Sie für Ihre Seitensprache die Übersetzung festlegen müssen, auch wenn Sie keine mehrsprachige Seite nutzen. Alle Texte werden hier und nicht im HTML-Template der Extension editiert.

Da dieser Code recht umfangreich ist, wird hier auf den kompletten Ausdruck verzichtet. Sie finden den vollständigen Code zur Übernahme in Ihre Beispielseite zum Download auf der Webseite des Verlags.⁵

Um die Beispielkonfiguration auch weiterhin nachvollziehen zu können, sollten Sie dieses Setup übernehmen und, wo nötig, an Ihre Bedürfnisse anpassen. Wenn Sie mit verteilten Templates arbeiten, binden Sie auch `+feuser` über **Include basis template** in das Haupttemplate Ihrer Seite ein, andernfalls kopieren Sie Setup und Constants ins Haupttemplate Ihrer Rootseite.

Vergessen Sie nicht, die statischen Templates *Feuser Registration CSS-styled* (`sr_feuser_registration`) und *Static Info Tables* (`static_info_tables`) in Ihrem TS-Template `+feuser` einzubinden (**Include static (from extensions)**).



Abbildung 3.9:
Statisches Template
zuweisen

Im Frontend können Sie nun Ihren ersten Testbenutzer anlegen (Abbildung 3.10).

⁵ Unter: <http://www.opensourcepress.de/fileadmin/osp/download/typo3webshop.zip/>

Abbildung 3.10:
Registrierungs-
formular

Die Felder, die mit einem Stern (*) sind Pflichtfelder!

Bevorzugter Benutzername:

Passwort: *
Wiederholung:

Herr/Frau: Herr Frau

Vorname: *
Nachname: *

Status:

Firma: *
Straße: *
Wohnort: *

County/province:

Land:

PLZ: *
Telefon: *
E-Mail: *

Kurzbeschreibung:

Klicken Sie den Link **Registrierung** an, füllen Sie alle Pflichtfelder im Formular aus und speichern Sie die Daten ab. Nach Bestätigung des Formulars und Absenden der Angaben (Abbildung 3.11) erscheint eine Nachricht, dass Ihre Registrierung entgegengenommen wurde, Sie diese aber bitte noch bestätigen mögen.

Abbildung 3.11:
Benutzerdaten
prüfen

**Guten Tag Testerin Testerin,
Bitte überprüfen Sie, ob alle Angaben korrekt sind:**

| | |
|---------------------------|---------------------------|
| Bevorzugter Benutzername: | Tersterin |
| Herr/Frau: | Frau |
| Vorname: | Testerin |
| Nachname: | Testerin |
| Status: | 1111 |
| Firma: | Testfirma |
| Straße: | Teststr. 5 |
| Wohnort: | Testort |
| County/province: | Nordrhein-Westfalen |
| Land: | Germany |
| PLZ: | 11111 |
| Telefon: | 22222 |
| E-Mail: | testerin@mytypo3domain.de |

Möchten Sie Informationen zu folgenden Themen erhalten?
Kurzbeschreibung:

Sobald Sie die Daten überprüft und die Anmeldung abgeschickt haben, erhalten Sie eine E-Mail mit einem Link, über den Sie die Registrierung komfortabel bestätigen und damit abschließen können. Erst jetzt werden Sie als registrierter Benutzer gelistet und können sich im Frontend mit den gewählten Zugangsdaten anmelden.

Haben Sie sich als Benutzer angemeldet, erscheint auf der Startseite nicht mehr der Link mit der Aufforderung zur Registrierung, sondern einer zum Formular, in dem Sie Ihre Benutzerdaten ändern und aktualisieren können.

Guten Tag
Bearbeiten Sie die Angaben zu Ihrem Konto
Die Felder, die mit einem Stern () sind Pflichtfelder!*

Konto löschen

Passwort: *
 Wiederholung:

Herr/Frau: Herr
 Frau

Vorname: *
 Nachname: *

Status

Firma: *
 Straße: *
 Wohnort: *
 County/province:
 Land:

PLZ: *
 Telefon: *
 E-Mail: *

Möchten Sie Informationen zu folgenden Themen erhalten?
 Kunden - Newsletter
 Presse Newsletter

Kurzbeschreibung:

Abbildung 3.12:
Benutzerdaten
ändern

Auch dieses Formular wird Ihnen zur Überprüfung nochmals vorgelegt und Sie können die Änderungen endgültig bestätigen oder verwerfen (Abbildung 3.13).

Abbildung 3.13:
Benutzerdaten
aktualisieren

**Guten Tag Testerin Testerin,
Bitte überprüfen Sie, ob alle Angaben korrekt sind, bevor Ihr Konto tester aktualisiert wird:**

| | |
|------------------------------|---------------------------|
| Bevorzugter Benutzername: | tester |
| Herr/Frau: | Herr |
| Vorname: | Testerin |
| Nachname: | Testerin |
| Status | 1111 |
| Firma: | Testfirma |
| Straße: | Teststr. 5 |
| Wohnort: | Testort |
| County/province: | Nordrhein-Westfalen |
| Land: | Germany |
| PLZ: | 11111 |
| Telefon: | 22222 |
| E-Mail: | testerin@mytypo3domain.de |
| Kurzbeschreibung: | |

Alle Texte – auch für die E-Mail-Benachrichtigungen – editieren Sie im TS-Setup in den Spracheinstellungen. Nicht benötigte Textausgaben lassen Sie nach dem = einfach leer.

Im Backend können Sie die Benutzerdaten bearbeiten, indem Sie den Sys-Ordner mit den Frontendbenutzern anklicken und im Webmodul entweder Seite oder Liste auswählen. Falls Sie viele Frontendbenutzer zu betreuen haben, ist Liste vorzuziehen, denn hier können Sie sich alle Felder in Tabellenform ansehen.

3.2.4 HTML-Template

Im HTML-Template der Extension soll noch ein Setup-Wert erwähnt und erläutert werden: `static_info_country`. Dieser wurde im Setup der Shop-Extension bereits für die Vorauswahl des Landes herangezogen (vgl. S. 57). Es sorgt dafür, dass in einer Combobox die Vorauswahl für ein Land getroffen wird. In den Beispieldateien ist der Marker bereits gesetzt. Wenn Sie diese Dateien nutzen, werden die Einstellungen also übernommen.

```
## Vorauswahl des Landes

plugin.tx_srfeuserregister_pil.create.defaultValues.static_info_country =
    DEU
```

3.2.5 Verschlüsselung

An dieser Stelle sei ein wichtiger Aspekt angesprochen, der weniger Shop-spezifisch ist, sondern Webserver allgemein betrifft, und darum hier auch nicht vertieft werden soll: die Sicherheit der zu übertragenden Daten. Bei jeder Webseite, auf der Kundendaten – egal, ob über eine Registrierung, Formulare, Umfragen oder Ähnliches – verarbeitet werden, sollten Sie sich Gedanken über die Verschlüsselung der zu übertragenden Daten machen, denn unter Normalbedingungen werden alle Texte im Klartext über das Internet übertragen. Das bedeutet, dass diese Daten auch von Dritten mitgeschritten werden könnten. Um das zu verhindern, können Sie ein SSL-Zertifikat bei Ihrem Webhoster oder direkt bei der Zertifizierungsstelle beantragen, das allerdings zusätzliche Kosten verursacht; SSL steht für *Secure Socket Layer*. Der URL der Seite steht dann `https` (Secure HTTP) als Protokollbezeichnung voran. Der Browser des Kunden erkennt daran, dass er von Ihrem Webserver ein Zertifikat anfordern und annehmen muss, anhand dessen er prüft, ob er wirklich mit dem Server verbunden ist, der in der URL angegeben ist. Ist das der Fall, signalisiert der Browser dem Kunden eine sichere Verbindung.

Bei der SSL-Verschlüsselung kann im Prinzip nur die gesamte Domain verschlüsselt werden; beachten Sie jedoch, dass dadurch der Webserver *deutlich* mehr belastet wird. Sofern Sie nur den Shop verschlüsseln möchten, können Sie für diesen eine Subdomain einrichten und diese verschlüsseln (lassen).

Für unsere Zwecke soll diese Installation zunächst genügen, auch wenn die Frontendbenutzerregistrierung weitere Features und Module bietet. Widmen wir uns im Folgenden lieber dem Ausbau vorhandener sowie neuer Shop-Funktionalitäten.

4

Kapitel

Produktpflege

Wenn Sie ein Produkt, wie in Abschnitt 1.4 ab Seite 24 beschrieben, anlegen, stehen vier *Registerkarten* zur Verfügung, in denen Sie dessen Spezifikationen im Detail eingeben und pflegen. Dieses System hat aber den gravierenden Nachteil mangelnder Flexibilität; man ist also auf definierte Felder angewiesen und kann nicht eigene Felder definieren, die auf das eigene Produktsortiment zugeschnitten sind. Dazu müsste die Extension um entsprechende Felder erweitert werden, was allerdings für Entwickler ohne Weiteres möglich ist. Auch nach einem Update des Shopsystems bleiben somit die entsprechenden Erweiterungen erhalten. Es könnten allerdings geringfügige Anpassungen der Hook-Funktionen notwendig werden.

Im Folgenden soll es um diese Masken und Felder und damit die für die Darstellung des Produkts im Frontend wie auch für das Payment relevanten Angaben gehen. Aktivieren Sie im Seitenmodul des TYPO3-Backend in der Eingabemaske eines Produkts **Zweite Optionspalette anzeigen** und **Feldbeschreibung anzeigen**. Damit erhalten Sie alle Felder mit einer Kurzbeschreibung des dahinterliegenden Datenbank-Tabellenfeldes.

Das Shopsystem greift auf vielfältige Weise auf die hier hinterlegten Informationen zu, sei es zur rasch überprüfbar Darstellung im Frontend, aber eben auch zur Umsetzung der in Kapitel 6 behandelten Versand- und Zahlungsmodalitäten. Hier sind Irrtümer oder Inkonsistenzen deutlich schwieriger zu erkennen (aber möglicherweise weitaus folgenschwerer), so dass Sie der Produktpflege besondere Aufmerksamkeit widmen sollten.

Schauen Sie sich nun jedoch die vier Registerkarten (**Allgemein**, **Varianten**, **Beziehungen** und **Versand**) samt ihren Eingabefeldern der Reihe nach an.

4.1 Registerkarte Allgemein

Die „allgemeinen“ Angaben zu einem Produkt sind in einzelne Felder einzutragen. Sie korrespondieren mit den gesetzten *Markern* im HTML-Template des Shops. Es handelt sich um eine Art Platzhalter, um den Inhalt des jeweiligen Feldes an der durch den Marker bestimmten Stelle auszugeben. Oder anders formuliert: Der Marker im HTML-Template des Shops wird durch den Inhalt des entsprechenden Feldes ersetzt. Das klappt allerdings nur innerhalb gewisser Subpart-Marker wie `ITEM_SINGLE_DISPLAY`. Eine Liste vieler Subparts finden Sie im Manual von `tt_products`, das der Extension beigefügt ist.

Die Syntax des Markers für ein Feld lautet:

```
PRAEFIX_NAME_DES_FELDES
```

Als Präfix dient hier `PRODUCT` für Produkte und `CATEGORY` für Kategorien. Haben Sie die Option **Feldbeschreibung anzeigen** ausgewählt, sehen Sie die Namen der Felder. Alternativ verwenden Sie die Datei `ext_tables.sql` aus dem Extension-Verzeichnis oder das Tool `phpMyAdmin` (vgl. Anhang C). Wenn im Beispiel-Template `tt_products_css.html` ein Subpart oder Marker, den Sie für Ihre Shop-Konfiguration benötigen, nicht enthalten ist, werden Sie in den anderen Beispiel-Templates der Extension `tt_products` fündig. Diese HTML-Templates finden Sie in der Extension im Verzeichnis `template`. Rufen Sie die Extension im Extension Manager auf, wählen Sie sie durch einen Klick auf den Namen aus und wählen Sie in der Combobox **Edit files**. Die meisten Subparts und Marker finden Sie in der Datei `example_template_bill_de.tpl`. Beachten Sie bitte, dass eine Template-Datei immer nach ISO-8859-1 kodiert sein muss.

4.1 Registerkarte Allgemein

Abbildung 4.1:
Registerkarte
Allgemein

Produkt (2) Ein Testprodukt

Allgemein | Varianten | Beziehungen | Versand

Einzelprodukt

Untername:

Hersteller:

Webseite:

EAN:

Jahr Lager (%):

Merkmal:

Preis: **Einkaufspreis:** **Netto:** **Aktion:** **Besonderheit:**

Bemerkung (Kb):

Sperre: **Spezialanfertigung:** **Materialität:** **Faktor der Materialität:**

Beschreibung: **Absterb:** **Text:**

Bemerkung (P): **Absterb:** **Text:**

Datei:

Dateiblick: **Herz Pross Webshop 2017**

Konzeption: **Verkaufsfähig:** **Verkaufsende:** **Statt:** **Shop:** **Zugriff:**

Name

Der Eintrag des Produktnamens ist dem Marker `PRODUCT_TITLE` im HTML-Template zugewiesen.

Untertitel

Der Eintrag des Untertitels Ihres Produkts wird dem Marker `PRODUCT_SUBTITLE` im HTML-Template zugewiesen. Bleibt der Eintrag leer, wird der Marker ignoriert. Im Beispielshop wird das Feld **Untertitel** für den oder die Autoren eines Buches verwendet.

WWW

Der Eintrag in diesem Feld zeigt die URL zu einer Webseite an und weist diese dem Marker `PRODUCT_WWW` im HTML-Template zu. Im Beispielshop entspricht dies der Webseite des oder der Autoren; ein Klick im Frontend leitet zur entsprechenden URL weiter. Im HTML-Template finden Sie den Link in der Form

```
<a href="###PRODUCT_WWW###" >Name des Autors</a>
```

Artikelnummer

Der Eintrag der Artikelnummer wird dem Marker `PRODUCT_ITEMNUMBER` im HTML-Template zugewiesen. Er sollte immer ausgefüllt und eindeutig sein, denn nur so ist die eindeutige Zuordnung von Kaufvorgang zu Produkt/Artikel gewährleistet. Im Beispielshop entspricht dies der ja ebenfalls eindeutigen ISBN eines Buches.

Am Lager

Hier tragen Sie die Anzahl der lieferbaren Exemplare/Einheiten eines Produkts ein. Sobald eine Bestellung ausgelöst wurde, reduziert sich der Warenbestand um die bestellte Menge. Im Beispielshop ist in den Constants jedoch `alwaysInStock=1` gesetzt, wodurch die Produkte grundsätzlich als verfügbar gesetzt sind. Wenn Sie das nicht wünschen, kommentieren Sie den Code in den Constants durch ein Raute-Zeichen (`#`) am Zeilenbeginn aus.

Haben Sie die Anzahl der vorrätigen Produkte hier eingetragen, erhalten Sie je nach Konfiguration eine Meldung, sobald Lieferengpässe zu befürchten sind. Dazu setzen Sie in den TS-Constants `warningInStockLimit -`

das gilt dann für alle Produkte –, um zu definieren, ab welchem Lagerbestand eine Warn-E-Mail an den Shopbetreiber versendet wird. Die Zeile `alwaysInStock=0` muss in den Constants ebenfalls gelöscht sein, da andernfalls der Lagerbestand nicht reduziert wird.

```
### Warnung ausgeben, wenn der Warenbestand reduziert wird
warningInStockLimit = 10
```

```
## Meldung, wenn Produkte nicht ab Lager vorrätig sind
notInStockMessage = Nicht vorrätig
```

Den Lagerbestand erhöhen Sie bei `tt_products`, indem Sie die Einträge im Feld **Am Lager** um die entsprechende Anzahl der Artikel erhöhen. Das ist im laufenden Betrieb nicht ganz unproblematisch, da ja zeitgleich Abgänge durch Bestellungen möglich sind. Sie können den Datensatz für Bestellungen darum kurzfristig sperren, indem Sie den Haken bei **Versteckt** setzen und die Anzahl ändern.

Kategorien

Wenn Sie mit Kategorien arbeiten, wählen Sie über eine Combobox die Kategorie aus, der das Produkt zugeordnet werden soll. Weitere Informationen finden Sie im Abschnitt 5.4 ab Seite 117.

Adressen

Wenn Sie mit der Adressverwaltung der Extension `tt_address` arbeiten, wählen Sie über die Combobox die entsprechende Referenzadresse aus. Allerdings ist die Ausgabe im Frontend in Version 2.6.0 noch nicht vollständig implementiert.

Preis

Der Eintrag des Produktpreises kann verschiedenen Markern zur Preisberechnung im HTML-Template zugewiesen werden, abhängig davon, welche Marker sie nutzen, etwa `PRICE_TAX` oder `PRICE_NO_TAX`. Weitere Informationen zu den Preismarkern finden Sie in den entsprechenden Abschnitten von Kapitel 6.

Die folgende Tabelle bietet eine Übersicht über die Preismarker.

Tabelle 4.1:
Preis- und
Versandmarker

| Marker | Beschreibung |
|----------------------------|---|
| PRICE_TOTAL_TAX | Gesamtpreis inkl. Steuern |
| PRICE_TOTAL_NO_TAX | Gesamtpreis exkl. Steuern |
| PRICE_TOTAL_ONLY_TAX | Steueranteil am Gesamtpreis |
| PRICE_TOTAL_O_TAX | wie PRICE_TOTAL_TAX, aber ohne Rabattabzug (ab Version 2.7.0) |
| PRICE_TAX_DISCOUNT | Nachlass auf den Gesamtpreis (seit Version 2.6.0) |
| PRICE_TAX_DISCOUNT_PERCENT | Nachlass in Prozent auf den Gesamtpreis (seit Version 2.6.0) |
| TAX_INCL_EXCL | gibt einen Text aus, ob der Preis inkl. oder exkl. Steuer ist |
| PRICE | usePriceTag Setup, um den Preis über TypoScript generiert darzustellen |
| PRICE_GOODSTOTAL_TAX | Gesamtwarenpreis inkl. Steuern |
| PRICE_GOODSTOTAL_NO_TAX | Gesamtwarenpreis exkl. Steuern |
| PRICE_GOODSTOTAL_ONLY_TAX | Steueranteil am Gesamtwarenpreis |
| PRICE2_GOODSTOTAL_TAX | Gesamtwarenpreis2 inkl. Steuern |
| PRICE2_GOODSTOTAL_NO_TAX | Gesamtwarenpreis2 exkl. Steuern |
| PRICE2_GOODSTOTAL_ONLY_TAX | Steueranteil am Gesamtwarenpreis2 |
| NUMBER_GOODSTOTAL | Anzahl der Artikel |
| PRICE_TAXRATE_GOODSTOTAL1 | Gesamtwarenpreis der Artikel mit Steuersatz 1. |
| PRICE_TAXRATE_ONLY_TAX1 | Steueranteil am Gesamtwarenpreis der Artikel mit Steuersatz 1. Steuersatz 2 erhält analog Ziffer 2 am Ende. |
| PRICE_SHIPPING_TAX | Versandkosten inkl. Steuern |
| PRICE_SHIPPING_NO_TAX | Versandkosten exkl. Steuern |
| PRICE_SHIPPING_ONLY_TAX | Steueranteil an den Versandkosten |
| SHIPPING_TITLE | Bezeichnung des Versands |
| SHIPPING_WEIGHT | Versandgewicht |
| DELIVERYCOSTS | Summe der Versand- und Bezahlkosten |
| PRICE_PAYMENT_TAX | Kosten der Zahlungsmethode |
| PRICE_TAX_DISCOUNT | Gesamtrabatt |
| PRICE_VAT | Steuersatz |
| PRICE_VOUCHERTOTAL_TAX | Gesamtpreis inkl. Steuer abzüglich eingelöster Gutscheinpunkte |
| PRICE_VOUCHERTOTAL_NO_TAX | Gesamtpreis exkl. Steuer abzüglich eingelöster Gutscheinpunkte |
| PRICE_TOTAL_CENT_TAX | Gesamtpreis inkl. Steuer in Cent |

Fortsetzung:

| Marker | Beschreibung |
|-----------------|--|
| CREDIT_DISCOUNT | Betrag der Ermäßigung durch eingelöste Kontopunkte |

Preis2

In Ihrem Shop können nicht nur Endkunden einkaufen, sondern beispielsweise auch Fachhändler, im Beispielshop etwa der Buchhandel. Dieser erhält natürlich andere Konditionen als die Endkunden, so dass Sie Händler einer bestimmten Benutzergruppe zuordnen müssen.

Den für diese Gruppe gültigen Preis eines Produkts ordnen Sie im Backendformular über das Feld **Preis2** zu.

Im TS-Setup Ihres Shop-Templates müssen Sie nun die Bedingung für die Benutzergruppe zuweisen. Hier verwenden Sie die globalen Zuweisungen von TypoScript.

Die Syntax lautet:

```
[usergroup=1]
priceNoReseller = 2
[global]
```

Diese Globale definiert, dass alle Mitglieder der Benutzergruppe mit der UID 1, den Preis2 anstelle des normalen Endkundenpreises erhalten, wenn Sie als eingeloggter Benutzer im Shop einkaufen.

Um mehreren Benutzergruppen diesen Preis zuzuordnen, lautet die Syntax:

```
[usergroup=1] || [usergroup=2]
priceNoReseller = 2
[global]
```

Ein weiteres Beispiel für die Anwendung der globalen Zuordnungen könnte auch eine bestimmte Uhrzeit sein, also eine Art „Happy Hour“ im Shop.

```
[usergroup=1] && [hour=0]
priceNoReseller = 2
[global]
```

Damit wird der Benutzergruppe mit der UID 1 der in Preis2 angegebene Betrag zur Stunde 0, also von Mitternacht bis 0:59 Uhr eingeräumt. Sie können die Happy Hour natürlich auch ohne die Zuordnung zu einer Benutzergruppe feiern.

Ein weiteres Anwendungsbeispiel ist die IP-Adresse. Weisen Sie allen Frontendbenutzern aus einem bestimmten IP-Bereich den Preis2 zu:

```
[usergroup=1] && [IP=123.*.*.*]  
priceNoReseller = 2  
[global]
```

Beachten Sie aber, dass Preis2 ein festes Feld ist und der dort definierte Betrag einmalig gesetzt werden kann. Weitere Möglichkeiten ähnlicher Zuordnungen finden Sie in der TSRef.

Wenn für jeden Kunden eigene Konditionen gelten sollen, ist es sinnvoller, den Rabatt über den Frontendbenutzer-Account zuzuordnen. Im Feld **Rabatt** tragen Sie einen Prozentwert ein. Sofern Sie für die Gruppe dieses FE-Benutzers den obigen Code und damit **Preis2** zugewiesen haben, wird der Rabatt auf dieses Feld angewendet. Haben Sie **Preis2** nicht zugewiesen, wird der im Feld eingegebene Rabatt auf den regulären Preis angerechnet.

Einkaufspreis

Im Backendformular des Produkts können Sie Ihren Einkaufspreis zuweisen und im Frontend – zum Beispiel in der Einzelansicht im Subpart `ITEM_SINGLE_DISPLAY` über den entsprechenden Marker `DIRECTCOST_TAX` bzw. `DIRECTCOST_NO_TAX` – anzeigen lassen. Der Marker ist im Template `tt_products_css.html` bereits gesetzt.

Das Feld **Einkaufspreis** steht derzeit nur für Produkte, nicht für Artikel zur Verfügung (zur Unterscheidung von Artikeln und Produkten vgl. Seite 96). Mit der Ausgabe informieren Sie Ihre Kunden lediglich über Ihren Einkaufspreis. Inwieweit oder in welchen Fällen eine solche Angabe sinnvoll ist, muss natürlich jeder Shopbetreiber selbst entscheiden.

MwSt.-Sätze

Über das TS-Setup des Shops ist bereits ein Default-Wert für die MwSt.¹ gesetzt. Wenn Sie Produkte vertreiben, die einem anderen MwSt.-Satz unterliegen, setzen Sie diesen im Backendformular des Produkts. Der dort eingetragene Wert überschreibt den Wert im TS-Setup. Wenn Sie Produkte ohne MwSt. ausweisen möchten (z. B. Gutscheine), finden Sie das Verfahren im Abschnitt 5.1.8 ab Seite 106 beschrieben.

Wenn Sie per Default keine MwSt. ausweisen möchten, lassen Sie im Setup die Definition `TAXpercentage=19` einfach weg und tragen Sie für alle Produkte die jeweils gültige Steuer direkt im Produkt ein. Lassen Sie das Feld für die Steuer leer, ist der Satz 0. Denken Sie daran, in diesem Fall `TAXpercentage` entsprechend zuzuweisen. Andernfalls wird der Default-

¹ Mag – gerade international – ein steuerrechtlicher Unterschied zwischen „Umsatzsteuer“ und „Mehrwertsteuer“ bestehen, so verwenden wir beide Begriffe hier synonym.

wert von 25 herangezogen. Der Eintrag in den Constants muss dann wie folgt definiert werden:

```
plugin.tt_products.TAXpercentage =
```

Als Faustregeln zur Ausweisung/Berechnung der MwSt. in Ihrem Shop kann Folgendes gelten:

Lieferung Inland

Brutto-Betrag berechnen, USt. ausweisen

Lieferung EU an Privatkunden (keine USt.-ID)

Brutto-Betrag berechnen, USt. ausweisen

Lieferung EU an Firmenkunden (USt.-ID ist dann Pflichtangabe)

Netto-Betrag berechnen, keine USt. ausweisen

Lieferung weltweit

Netto-Betrag berechnen, keine USt. ausweisen

Grundsätzlich sollten Sie sich aber insbesondere bei Lieferungen ins Ausland mit Ihrem Steuerberater über das korrekte Vorgehen verständigen.

Aktion

Mit einem Haken bei **Aktion** wählen Sie Produkte für eine Sonderaktion aus. Aktionsprodukte sind solche, die Sie besonders bewerben möchten. Aktionen sind zeitlich begrenzt, und es können – müssen aber nicht – besondere Preise gelten. Im Beispielshop sind Neuerscheinungen Aktionsprodukte. Legen Sie eine neue Seite mit dem Titel **Aktion** an, fügen Sie das Shop-Plugin ein und wählen Sie über den Anzeigentyp/Option **Produkte: Aktionen** aus. Auf dieser Seite werden sämtliche Aktionsprodukte gelistet. Sie können dem Aktionsprodukt ein Start- und ein Stoppdatum zuweisen, um die Aktion nicht manuell beenden zu müssen. Allerdings wird das Produkt nach Ablauf der Aktion dann auch nicht mehr in der regulären Listenansicht angezeigt. Sofern Sie Aktionsprodukte bewerben, die nach der Aktion wieder in Ihren normalen Warenbestand zurück fließen sollen, muss die Aktion manuell beendet werden. Sie müssen dann den Haken bei **Aktion** entfernen und evtl. den Preis zurücksetzen.

Ist Ihnen dieser Weg zu mühsam, behelfen Sie sich mit einem eigenen Sys-Ordner für die Aktionsprodukte. Weisen Sie der Seite mit dem Titel **Aktion** diesen SysOrdner als Ausgangspunkt zu und vergeben Sie ein Start- und Stoppdatum, das Sie entweder im SysOrdner oder auf der Aktionsseite setzen. Wenn Sie keine eigene Seite erstellen, sondern ein Plugin in eine bestehende Seite eingebunden haben, vergeben Sie das Datum für das Content-Element mit Ihrem Shop-Plugin.

Wünschenswert wäre eine Erweiterung des Shopsystems, durch die Aktionsprodukte zeitlich begrenzt in einer Aktion angezeigt werden. Sobald der Haken bei Aktion gesetzt ist, sollte es ein Start- und Stopppdatum geben; nach Ablauf der Aktion würde das Produkt (wieder) im regulären Warenbestand angezeigt, jedoch nicht mehr auf der Aktionsliste. Außerdem sollte diese Funktion auch für Artikel genutzt werden können, was derzeit nicht möglich ist, weil Artikel keine Start- und Stoppzeit nutzen.

Besonderheit

Mit der Aktivierung dieses Häkchens können Sie Ihre Produkte hervorheben. Die Funktionsweise ist analog zu den Aktionsprodukten.

Erstellen Sie eine Shopseite **Besonderheit**, fügen Sie das Plugin in diese Seite ein und wählen Sie in den Plugin-Optionen das Objekt **Produkte: Liste Highlights**. Geben Sie als Ausgangspunkt den Produkte-SysOrdner an und speichern Sie den Datensatz. Es werden nun alle Produkte mit Häkchen bei **Besonderheit** auf dieser Seite gelistet. Die reguläre Listenansicht der Produkte ist davon nicht berührt.

Wenn Sie die Anzeige von „besonderen“ Produkten in der normalen Listenansicht nicht wünschen, erstellen Sie für die Besonderheiten einen eigenen SysOrdner und weisen Sie diesen als Ausgangspunkt der Seite **Besonderheiten** und allen anderen Shopseiten zu, nicht jedoch der regulären Listenansicht. Bitte beachten Sie, dass dieses Feature nur für Produkte, nicht für Artikel zur Verfügung steht.

Gewicht

Hier tragen Sie das Gewicht des Produkts ein. Wenn Sie keinen Eintrag vornehmen, wird der entsprechende Marker `PRODUCT_WEIGHT` ignoriert. Das Gewicht des Produkts können Sie u. a. zur Berechnung der Versandkosten heranziehen (vgl. Abschnitt 6.2 ab Seite 142).

Das Gewicht in diesem Feld wird in Kilogramm angegeben. Beachten Sie, dass im TS-Setup Angaben in Gramm notwendig sind. Weitere Informationen finden Sie in dem oben bereits erwähnten Abschnitt.

Sperrige Produkte

Sperrige Produkte benötigen aufgrund Ihrer Maße einen besonderen Transport und verursachen damit meist zusätzliche Lieferkosten.

Wenn Sie im Backendformular des Produkts den Haken bei **sperrig** gesetzt haben, wird zunächst nur die Ausgabe eines Textes veranlasst, die sog.

„Bulkily Warning“. Im Template ist dafür der Marker `BULKILY_WARNING` zuständig. Den entsprechenden Text setzen Sie in den Constants oder über den Constant-Editor.

In den Constants des Shop-Templates `+tt_products` muss ein Code für die „Bulkily Addition“ gesetzt werden, wenn Sie für das Produkt zusätzliche Versandkosten veranschlagen. Damit addiert sich, wie der Name schon sagt, der dort gesetzte Betrag zu den normalen Lieferkosten, im folgenden Beispiel sind das zusätzliche EUR 100,-.

```
### Textausgabe für sperrige Produkte
plugin.tt_products.bulkilyWarning = Achtung: Dieser Artikel erfordert
speziellen Transport

### Additive Zusatzkosten für sperrige Produkte
plugin.tt_products.bulkilyAddition = 100
```

Wenn Sie lediglich die Warnung ausgeben, aber keinen Betrag ausweisen möchten, müssen Sie den Wert auf 0 setzen. Im Constant Editor greift andernfalls der Default von 25, so dass es nicht hilft, lediglich den Bereich in den Constants auszukommentieren. Weitere Informationen zum Thema Versandkosten finden Sie im Kapitel 6.

Im HTML-Template `tt_products_css.html` ist der Marker in der Einzelansicht des Produkts, im Template-Suffix `SPEZIELL` und im Template-Suffix `REGISTRIERT` zu Beispielpurposes gesetzt. Sie können den Marker jedoch auch in der normalen Listenansicht einsetzen. Ein Template-Suffix ist ein spezieller Subpart für ein Plugin. Wenn Sie eine zusätzliche Listenansicht für Produkte ohne Bilder eventuell in der rechten Spalte wünschen, können Sie die Listenansicht kopieren, nach der normalen Listenansicht wieder einfügen, den Subpart entsprechend umbenennen und den Bildmarker entfernen. Diesem Plugin weisen Sie dann das entsprechende Suffix zu. Im Abschnitt 5.6 ab Seite 128 finden Sie weitere Informationen über die Erstellung von Template-Suffixen.

Spezialanfertigung

Produkte dieses Typs unterscheiden sich grundlegend von „normalen“ Produkten, als sie nicht bestellt, sondern lediglich angefragt werden können.

In den TS-Constants des Shops stellen Sie Folgendes ein:

```
specialpreparation = Spezialanfertigung möglich!
<a href=?id=999=###PRODUKT_ID###>hier bestellen</a>
```

Der Link führt zu einem Formular, mit dem das Produkt vom Kunden nachgefragt werden kann. Obwohl dieser Code in den Constants gesetzt ist, kann

das Produkt immer noch in den Warenkorb gelegt und damit bestellt werden. Das kann durch ein Template-Suffix verhindert werden. Dieser Subpart enthält dann keine Warenkorbfunktion.

Faktor der Maßeinheit

Dieses Feld macht es möglich, den Preis eines Produkts pro Kilogramm, Liter oder nach einer anderen beliebigen Maßeinheit anzugeben. Dazu müssen die Marker `PRICE_TOTAL_TOTUNITS_TAX` und `PRICE_TOTAL_TOTUNITS_NO_TAX` gesetzt werden. Der Preis errechnet sich dann aus dem Preis pro Einheit multipliziert mit der Anzahl der gekauften Einheiten.

Beschreibung

Dieses Textfeld dient einer Produktbeschreibung. Sofern verwendet, können Sie alle Funktionen des Rich Text Editors darin nutzen. Das Textfeld wird über den Marker `PRODUCT_NOTE` im Subpart `ITEM_SINGLE_DISPLAY` angesprochen. Dieser Marker kann auch in der Listenansicht `LIST_ITEM_DISPLAY` genutzt werden.

Kommentar

Seit Version 2.5.0 stellt `tt_products` ein zweites Editorfeld zur Verfügung. Sofern verwendet, können Sie auch hier alle Funktionen des RTE nutzen. Das Textfeld wird über den Marker `PRODUCT_NOTE2` im Subpart `ITEM_SINGLE_DISPLAY` angesprochen. Der Vorteil eines zweiten Textfeldes besteht darin, dass dieser Text via CSS leicht platziert werden kann, ohne dass Sie den Editor um eigene Klassen erweitern müssen. Selbstverständlich steht Ihnen auch diese Option offen.

Wenn Sie keine weiteren Definitionen zur Platzierung des Textfeldes vornehmen, wird dieser Bereich im Frontend direkt unter dem ersten Textfeld angezeigt.

So können Sie das Textfeld `PRODUCT_NOTE` beispielsweise direkt neben Ihrem Produktbild positionieren und den zweiten Textbereich `PRODUCT_NOTE_2` darunter oder darüber setzen.

Beschreibende Seiten

In der Einzelansicht des Produkts können Sie einen Link zu beschreibenden Seiten setzen und im Backendformular auf eine beliebige Seite des Internetauftritts verweisen, die zusätzliche Informationen bereitstellt. Titel und Inhalt der Seite erscheinen unterhalb der Produktinformationstexte.

Die beiden Marker `PRODUCT_NOTE_UID_1_title` und `PRODUCT_NOTE_UID_1_1_TEXT_bodytext` definieren die erste beschreibende Seite. Im Beispielshop sehen Sie in der Einzelansicht, dass der Marker `PRODUCT_NOTE_UID_1_title` durch „Der TYPO3 Webshop“ ersetzt wird, und `PRODUCT_NOTE_UID_1_1_TEXT_bodytext` durch „Diese Seite dient als zusätzliche Beschreibungsseite für dieses Produkt“.

Um mehrere beschreibende Seiten zuzuweisen, muss ein weiterer Marker gesetzt werden. Hierbei ist, wie Sie im Beispielcode sehen, die Nummerierung wichtig.

`PRODUCT_NOTE_UID_1_title` gibt den Seitentitel aus. Die Nummer (hier 1) in diesem Marker entspricht der Reihenfolge der zugeordneten Seiten zum Produkt. Beim Marker `PRODUCT_NOTE_UID_1_1_TEXT_bodytext` wird mit der ersten Zahl bestimmt, dass es sich um die erste beschreibende Seite handelt (wie beim Marker Seitentitel), die zweite Zahl weist die Nummer des Inhaltselements vom Typ `TEXT` dieser Seite aus, das dort angezeigt werden soll. In diesem Fall enthält die beschreibende Seite lediglich einen Titel und ein Inhaltselement, so dass diese zugeordnete Zahl ebenfalls die 1 ist. Wenn mehrere Text-Inhaltselemente auf dieser Seite stehen, Sie aber nur das dritte Textelement einfügen möchten, würde hier eine 3 stehen. Das Element ist dem Feld `CType` der Tabelle `tt_content` entnommen. Ein normales Inhaltselement mit Texten besitzt den `CType` `text`.

Für die zweite zugewiesene beschreibende Seite gilt dann die Zahl 2 für den ersten Marker, denn es ist das zweite zugewiesene Element. Für den zweiten Marker gilt zunächst auch die Zahl 2, gefolgt von der Zahl 1, weil es auch auf dieser Seite lediglich ein Inhaltselement gibt, und das die Nummer des Inhaltselements ist.

In unserem Beispielshop finden Sie im Subpart `ITEM_SINGLE_DISPLAY` dazu folgenden Code.

```
<div id="PRODUCT_NOTE_SINGLE">
    ###PRODUCT_NOTE### <br />
    ###PRODUCT_NOTE2### <br /><br />

    <strong>###PRODUCT_NOTE_UID_1_title###</strong><br />
    ###PRODUCT_NOTE_UID_1_1_TEXT_bodytext### <br /><br />

    <strong>###PRODUCT_NOTE_UID_2_title###</strong><br />
    ###PRODUCT_NOTE_UID_2_1_TEXT_bodytext### <br /><br />
</div>

<div id="PRODUCT_RELATED_UID_SINGLE">
```

Um das Beispiel nachzuvollziehen, legen Sie zwei reguläre Seiten mit je einem Textelement an. Diese beiden Seiten weisen Sie dann über das Feld **Beschreibende Seiten mit Durch Datei Browsen zu**. Im Frontend finden Sie die Seitentitel und die Inhaltselemente unterhalb des Produkts gelistet.

Bild

Der Eintrag des Produktbildes wird dem Marker `PRODUCT_IMAGE` im HTML-Template zugewiesen. Leider bietet `tt_products` keine „echte“ Möglichkeit, den einzelnen Bildern einen Alternativ- oder Titeltext zuzuweisen. Allerdings können Sie den Titeltext als Alternativtext verwenden (das Verfahren beschreibt Abschnitt 5.6 ab Seite 129).

Sie können jedem Produkt beliebig viele Bilder zuweisen. Geben Sie am Ende des jeweiligen Bilder-Markers die Nummer des gewünschten an. Für die Anzeige ist die Reihenfolge des Einfügens der Bilder ins Produkt ausschlaggebend.

```
PRODUCT_IMAGE1, PRODUCT_IMAGE2 ... PRODUCT_IMAGE100
```

Diese Marker sind im Beispiel-Template nicht gesetzt und müssen hinzugefügt werden, sofern Sie mehrere Bilder verwenden möchten.

Folgender Code im Setup `tt_products` erlaubt die Verwendung von zehn Bildern in der Listenansicht und 100 Bildern in der Einzelansicht:

```
limitImage = 10
limitImageSingle = 100
separateImage = 1
```

Mit `separateImage` legen Sie fest, dass jedes Bild über einen eigenen Marker angezeigt wird und nicht alle Bilder aneinandergereiht werden und den Marker `PRODUCT_IMAGE` erhalten.

Datenblatt

Sie können jedem Produkt eine Datei zuordnen, die `tt_products` als „Datenblatt“ bezeichnet. Diese Datei kann z. B. eine Gebrauchsanweisung oder eine Konstruktionszeichnung sein, was immer Ihnen als Zusatzinformation wichtig erscheint.

Das Datenblatt kann in einem der Dateiformate DOC, HTM, HTML, PDF, SXW, TXT, XLS, GIF, JPG oder PNG vorliegen und eingebunden werden. Speichern Sie die entsprechende Datei im Fileadmin-Verzeichnis ab, wählen Sie das Produkt aus und weisen Sie die Datei über **Durch Datei Browsen** zu. Sie können diese Datei auch direkt von Ihrem lokalen System laden.

In der Einzelansicht finden Sie dann einen Link, der zu dieser Datei leitet, die im gleichen Browserfenster geöffnet wird. Leider gibt es keinen Automatismus, der Sie vom Datenblatt wieder zurück zum Produkt leitet. Diese Möglichkeit bieten z. B. HTML-Dateien, denen Sie ein `history.back()` zuweisen können.

Im HTML-Template finden Sie den Marker für das Datenblatt im Subpart `ITEM_SINGLE_DISPLAY`:

```
<!--###LINK_DATASHEET###--> Datenblatt <!--###LINK_DATASHEET###-->
```

4.2 Registerkarte Varianten

Das Shopsystem unterscheidet Produkte und Produktvarianten, Artikel genannt. Diese Unterscheidung ist allerdings weniger systematisch als historisch bedingt. Zunächst gab es in tt_products ausschließlich Produkte.

Pfad: /Start/Shop/Produkte/
Produkte [1] - Ein Testprodukt

Allgemein Varianten Beziehungen Versand

Farbe (Variante 1):
Hardcover; Paperback

2. Farbe (Variante 6):

3. Farbe (Variante 7):

Größe (Variante 2):

2. Größe Breite (Variante 8):

3. Größe Höhe (Variante 9):

Beschreibung (Variante 3):

Staffellungen (Variante 4):

Material (Variante 10):

Qualität (Variante 11):

Zusätzliches (Variante 9):

nur einzeln:

ist ein Bild:

immer auf Lager:

kein Mindestbestellwert:

Zweite Optionspalette anzeigen
 Feldbeschreibung anzeigen
 Schalte Rich Text Editor (RTE) ab

Abbildung 4.2:
Die Registerkarte
Varianten

Produkte können jedoch Varianten haben: Ein Buch kann als „Hardcover“ oder „Paperback“ angeboten werden, dasselbe T-Shirt(-Modell) kann es in verschiedenen Größen und Farben geben. Es ändern sich also nur einige (wenige) Eigenschaften desselben Produkts, wodurch dann einzelne Varianten/Artikel entstehen.

Damit müssen Sie, um zahlreiche Varianten eines Produkts anbieten zu können, nicht unzählige Angaben wiederholt machen, sondern legen soz. ein „Elternelement“ an, dem Sie die verschiedenen „Kindelemente“ zuweisen.

Auf dieser Registerkarte geht es um eben diese Varianten. Für unser Beispiel nutzen wir die erste Variante und tragen dort durch Semikolon getrennt die Varianten Hardcover und Paperback ein.

Abbildung 4.3:
Varianten



Im Frontend erscheint daraufhin eine zusätzliche Combobox zur Auswahl (Abbildung 4.4).

Abbildung 4.4:
Varianten im
Frontend



Nun müssen noch die einzelnen Artikel erstellt werden, in unserem Fall also Hardcover- und Paperbackausgabe des Buches. Wechseln Sie zunächst in die TS-Constants des Shop-Templates und setzen Sie dort

```
## Artikeltabelle benutzen
plugin.tt_products.useArticles = 1
```

Erst damit greift das Shop-System tatsächlich auf die Artikeltabelle zu.

Wählen Sie anschließend im Webmodul Liste den SysOrdner mit Ihren Produkten an und **Neuen Datensatz anlegen**. Aus der Liste wählen Sie **Produkt Artikel**.



Abbildung 4.5:
Datensatz anlegen
Produkt Artikel

Daraufhin erhalten Sie das Backendformular für den Artikel; tragen Sie dort den Titel **Ein Testprodukt: Hardcover**, eine Artikelnummer und, wenn gewünscht, einen Untertitel ein. Unter **Produkt** wählen Sie das Elternelement, tragen ins Feld **Variante 1** **Hardcover** ein und speichern den Datensatz ab.

The image shows a backend form for creating a new article. The form is titled 'Produkt Artikel [2] - Ein Testprodukt: Paperback'. It contains the following fields:

- Verstecken:** A checkbox that is currently unchecked.
- Titel:** A text input field containing 'Ein Testprodukt: Paperback'.
- Untertitel:** An empty text input field.
- Produkt:** A dropdown menu with 'Ein Testprodukt' selected and a 'Produkte' button below it.
- Artikel Nr.:** A text input field containing 'PB 123456'.
- Am Lager (St.):** A text input field containing '1'.
- Preis:** A text input field containing '29.90'.
- Preis (2):** A text input field containing '0.00'.
- Gewicht (kg):** A text input field containing '0.00'.
- Farbe (Variante 1):** A text input field containing 'Paperback'.

Abbildung 4.6:
Backendformular:
Artikel

Für den nächsten Artikel gehen Sie genauso vor, vergeben allerdings den Titel Ein Testprodukt: Paperback, eine andere Artikelnummer und einen anderen Preis.

Alle Eigenschaften, die Sie im Produkt gesetzt haben, werden in den Artikel übernommen, sofern Sie die Felder im Backendformular nicht ausfüllen und damit überschreiben. Lediglich ein Titel muss vergeben und die entsprechende Variantenzuordnung vorgenommen werden. Alle Felder, die Sie darüber hinaus ausfüllen, überschreiben die Werte aus dem Produkt.

Wenn Sie nun im Frontend das „Paperback“ aus der Combobox auswählen, ändert sich der Preis in der Listenansicht zunächst nicht. Das funktioniert erst mit Version 2.7.0. In der hier verwendeten Version 2.6.0 wird der Preis erst im Warenkorb korrekt geändert angezeigt, nicht in der Listen- oder Einzelansicht.

Weitere Varianten weisen Sie auf die gleiche Weise zu.

Im HTML-Template müssen die Subparts und Marker für alle benötigten Varianten gesetzt sein. Das Code-Beispiel zeigt die Varianten 1 und 2 für die Listen- und Einzelansicht.

```
<!--###display_variant1### -->

###GW2B### Art: ###GW2E###
<SELECT style="font-size: 10px"
name="###FIELD_COLOR_NAME###">###PRODUCT_COLOR###</SELECT>

<!--###display_variant1### -->

<!-- ###display_variant2### -->

###GW2B### Größe: ###GW2E###
<SELECT style="font-size: 10px"
name="###FIELD_SIZE_NAME###">###PRODUCT_SIZE###</SELECT>

<!-- ###display_variant2### -->
```

Im Warenkorb-Template setzen Sie lediglich die Variantenmarker ###PRODUCT_COLOR### ###PRODUCT_SIZE###. Im Beispiel-Template sind die Varianten 1 und 2 bereits eingefügt.

Bei Variante 5 gibt es mehrere Möglichkeiten; eine ist die mit `isSingle`. Dabei handelt es sich um einen Artikel, der einmalig bestellt werden kann.

Der Subpart im HTML-Template wird wie folgt unter die anderen Varianten gesetzt:

```
<!-- ###display_variant5_isSingle### -->
```

```

Einzelprodukt, das nur einmal gewählt werden kann.
###PRODUCT_ADDITIONAL_SINGLE###

<!-- ###display_variant5_isSingle### -->

```

Rufen Sie zu Ihrem Produkt die Registerkarte **Varianten** auf und setzen Sie einen Haken bei **nur einzeln**. Im Frontend sehen Sie bei Ihrem Produkt nun den Text und eine Checkbox, um diese Variante zu bestellen.

Checkliste

- Sie haben kein Produkt als Elternelement angelegt.
- Sie haben keine Varianten zugewiesen.
- Sie haben die Varianten nicht durch Semikolon voneinander getrennt.
- Sie haben dem Artikel kein Elternprodukt zugewiesen.
- Sie haben dem Artikel keinen Varianteneintrag zugewiesen oder einen Tippfehler in der Schreibweise.
- Sie haben dem Artikel keinen neuen Preis zugewiesen.
- Sie haben in den Constants `useArticles` nicht auf 1 gesetzt.

4.3 Registerkarte Beziehungen

In der Registerkarte **Beziehungen** definieren Sie Beziehungen zwischen Produkten.



Abbildung 4.7:
Backendformular:
Beziehungen

4.3.1 Template Produkte

Dieses Feature steht leider noch nicht zur Verfügung.

4.3.2 Verwandte Produkte

Ein verwandtes Produkt kann z. B. ein Buch zum gleichen oder einem ähnlichen Thema sein, ein Buch desselben Autors oder das Buch in anderer Sprache. In der Einzelansicht können Sie verwandte Produkte in Form der Listenansicht unterhalb des ausgewählten Produkts anzeigen. Dort können diese auch direkt bestellt werden.

Die Zuordnung erfolgt über den Link **Durch Datei Browsen** und Auswahl eines entsprechenden Produkts.

Der Marker `PRODUCT_RELATED_UID` greift im Subpart `ITEM_SINGLE_DISPLAY` auf den Subpart `ITEM_LIST_RELATED_TEMPLATE` zu, der als Vorlage für die Anzeige des Produkts verwendet wird.

4.4 Registerkarte Versand

Sofern der pauschale Versandkostenzuschlag von EUR 100,- bei sperrigen Produkten (vgl. Seite 74) nicht ausreicht, können Sie spezifische Versandkosten je Produkt im Backendformular über die Registerkarte **Versand** setzen.

Abbildung 4.8:
Backendformular:
Versand

The screenshot shows a web-based form for product shipping settings. At the top, the path is '/Start/Shop/Produkte/' and the product is identified as 'Produkte [1] - Ein Testprodukt'. There are four tabs: 'Allgemein', 'Varianten', 'Beziehungen', and 'Versand', with 'Versand' being the active tab. The form is divided into several sections:

- Versandkosten pro Produkt:** A text input field containing '0.00'.
- Zusätzliche Versandkosten:** A text input field containing '0.00'.
- Bearbeitungsgebühr:** A text input field containing '0.00'.
- Lieferung:** A dropdown menu with three options: 'auf Kundenwunsch', 'sofort lieferbar' (which is currently selected and highlighted in blue), and 'in kürze lieferbar'.

At the bottom of the form, there are standard window control icons (minimize, maximize, close, save, print, etc.).

Das gilt natürlich nicht nur für sperrige Güter, sondern kann auch bei anderen Besonderheiten (z. B. Gefahrgut, Bearbeitungsgebühren etc.) genutzt werden. Künftig (ab Version 2.7.0) stehen Ihnen dann drei verschiedene Felder zur Verfügung: **Versandkosten pro Produkt**, **Zusätzliche Versandkosten** und **Bearbeitungsgebühr**.

Beachten Sie, dass all diese Zusatzkosten zuletzt zu den im TS-Setup definierten Default-Werten addiert werden.

Der Haken bei **sperrig** ist dann nicht mehr zu setzen, allerdings erhalten Sie dann auch keinen entsprechenden Hinweis („Bulkily Warning“).

Lieferung

Über den Marker `PRODUCT_DELIVERY` im Subpart `ITEM_LIST_TEMPLATE` kann ein Icon z. B. hinter dem Preis des Produkts ausgegeben werden. Per Default sind das Punkte (Bullets) in den Farben Gelb („auf Kundenwunsch“), Rot („in kurze lieferbar“) und Grün („sofort lieferbar“).

Wenn Sie in der Box unter **Lieferung** eine Auswahl treffen, wird das entsprechende Standard-Icon gesetzt. Treffen Sie hier keine Auswahl, nutzen aber den Marker im Template, wird das gelbe Icon gesetzt. Wenn Sie dieses Feature gar nicht nutzen möchten, müssen Sie den Marker aus dem HTML-Template entfernen.

4.5 Mehrsprachigkeit

Da TYPO3 mehrsprachig genutzt werden kann, sind auch viele Extensions für verschiedene Sprachen ausgelegt. Das gilt auch für Shop-Extensions. Ob Sie Mehrsprachigkeit nutzen, um die Sprachlabels der Programme zu überschreiben oder einen mehrsprachigen Einsatz des gesamten Shops planen – die Sprachmodule können für beide Szenarien angepasst werden.

Die Entscheidung, ob ein solch „internationaler“ Shop sinnvoll ist, sollte allerdings reiflich überlegt sein. Länder wie die Schweiz oder Belgien mit mehreren Landessprachen, aber einem Steuersystem und einer Währung, könnten eine solche Umsetzung nahelegen. Grundsätzlich muss der Verkauf in verschiedene Länder aber eine Fülle von Aspekten berücksichtigen (Versandkosten, Ausfuhrbestimmungen, Umsatzsteuer, Währung etc.), vor deren Hintergrund die mehrsprachige Darstellung rasch zur Marginalie wird.

Grundsätzlich kann `tt_products` auf mehrsprachigen Seiten eingesetzt werden.² Um die dazu notwendigen Konfigurationsschritte soll es im Folgenden gehen.

Es genügt jedoch nicht, lediglich das HTML-Template und die `loccallang`-Dateien zu bearbeiten, denn Produkte und Artikel müssen ebenso übersetzt werden wie alle für die Textausgaben des Shops zuständigen Bereiche in Constants und Setup des Extension-Template.

4.5.1 TS-Template

Beginnen Sie mit der allgemeinen Zuweisung der Sprachen in den Constants. Übernehmen Sie den folgenden Code für die Einstellung der Sprach-Variablen `L` ins TS-Setup Ihres Shops, die in den Links weitergereicht werden muss.

```
#
# Es ist auch Deutsch, wenn es keinen Parameter gibt.
config.linkVars = L

config.sys_language_uid = 0
config.language = de
config.locale_all = german

# Deutsche Sprache, sys_language.uid = 0
[globalVar = GP:L = 0]
config.sys_language_uid = 0

# Englische Sprache, sys_language.uid = 1
[globalVar = GP:L = 1]
config.sys_language_uid = 1
config.language = en
config.locale_all = english

# Französische Sprache, sys_language.uid = 2
[globalVar = GP:L = 1]
config.sys_language_uid = 2
config.language = fr
config.locale_all = french
```

Damit ist die Grundlage für verschiedene Sprachausgaben geschaffen: Zunächst werden das HTML-Template und der E-Mail-Anhang mit den AGB in der Default-Sprache angesprochen. Die folgende Bedingung geht analog für alternative Seitensprachen vor. Ergänzen Sie die von Ihnen gewünschten Sprachen, wie in diesem Beispiel Französisch, das im Beispielshop nicht verwendet wird.

² In diesem Buch wird eine entsprechende Konfiguration der Webpräsenz allerdings vorausgesetzt und als allgemeines TYPO3-Thema nicht eigens ausgeführt.

```

plugin.tt_products {
    file.templateFile = fileadmin/templates/tt_products_css.html
    AGBattachment = fileadmin/templates/agb_de.txt
}

[globalVar = GP:L = 1]
plugin.tt_products{
    file.templateFile = fileadmin/templates/tt_products_css_en.html
    AGBattachment = fileadmin/templates/agb_en.txt
}
[global]

[globalVar = GP:L = 2]
plugin.tt_products{
    file.templateFile = fileadmin/templates/tt_products_css_fr.html
    AGBattachment = fileadmin/templates/agb_fr.txt
}
[global]

```

4.5.2 HTML-Template

Es folgt ein wenig Fleißarbeit: Das HTML-Template des Shops muss in die gewünschten Sprachen übersetzt und, wie im Codebeispiel beschrieben, zugewiesen werden. Durch ausgiebiges Testen finden Sie auch noch die letzten Reste der Default-Sprache im HTML-Template.

Nutzen Sie nur ein einziges HTML-Template für alle Sprachen, können Sie mit Template-Suffixen arbeiten. Erstellen Sie dann anstelle der Templates für die verschiedenen Sprachen für jeden Subpart ein Template-Suffix, indem Sie das Suffix jeweils unter den Default-Subpart setzen und diesen Teil in die gewünschte Sprache übersetzen. Dem Plugin der Seitenübersetzung müssen Sie dann lediglich das neue Suffix zuweisen.

locallang-Dateien

Im Setup des TS-Templates können Sie – müssen aber nicht – die Zuordnungen zu den Feldnamen ändern. Interessant sind folgende Dateien:

locallang.xml

enthält nur einen Satz zur Beschreibung der Extension und die Begriffe „Standard“, „Erweitert“ und „Varianten“.

locallang_db.xml

enthält die Namen der Datenbankfelder.

pii/locallang.xml

enthält einige Warnhinweise, Texte zur Steuer etc. in der Extension.

Das Vorgehen entspricht dem bei anderen Extensions. Notieren Sie die gewünschten Texte im TS-Setup, wie der folgende Code beispielhaft zeigt:

```
plugin.tt_products_p11._LOCAL_LANG {
de (
    tax_included = enthält MwSt.
    ...
    ...
)
}

plugin.tt_products._LOCAL_LANG {
de (
    tt_products.note = Produktbeschreibung
    tt_products.itemnumber = ISBN:
    ...
    ...
)
}
```

Mit solchen Setup-Einträgen können Sie alle Texte aus den xml-Dateien im `tt_products`-Verzeichnis ersetzen und übersetzen. Möchten Sie Ihre Übersetzungen bislang noch nicht implementierter Sprachen der Allgemeinheit zur Verfügung stellen, wenden Sie sich bitte an das TYPO3-Übersetzungsteam.³

Seiten und Produkte

Nun stehen die einzelnen Shopseiten zur Übersetzung an. Erstellen Sie für jede Seite eine alternative Seitensprache und die entsprechenden Übersetzungen. Das Plugin können Sie mit der Funktion der Standardübersetzung kopieren und dort lediglich den Teil, der den Text „(Kopie)“ enthält, entfernen.

Um die Produkte in der alternativen Sprache anzulegen, wählen Sie den Produkte-SysOrdner, klicken auf **Neuen Datensatz anlegen** und wählen dort **Produkte (alternative Sprache)**. Wenn Sie mit Kategorien arbeiten, können Sie diese auf die gleiche Weise übersetzen. Wählen Sie dazu **Produkte Kategorie (alternative Sprache)**. Für Artikel ist momentan keine entsprechende Übersetzungsmöglichkeit vorhanden.

³ <http://translation.typo3.org>



Abbildung 4.9:
Produkte (alternative
Sprache)

Mit Klick auf **Produkte (alternative Sprache)** erhalten Sie ein leeres Backend-formular in der alternativen Sprache. In der Combobox für die Sprachauswahl (in Abbildung 4.10 mit „Englisch“ belegt), wählen Sie die neue Sprache aus. Voraussetzung ist natürlich, dass diese für TYPO3 bereits über den Extension Manager heruntergeladen wurde. Weitere Informationen zum Translation Handling finden Sie im Anhang 263.

 A screenshot of the TYPO3 backend form for editing a product in an alternative language. The form is titled "Produkte (alternative Sprache) [1] - A test product". It contains several sections:

- Sprache:** A dropdown menu with "Englisch" selected.
- Verstecken:** A checkbox that is unchecked, and three input fields for "Start:", "Stop:", and "Zugriff:".
- Produkte Datensatz:** A dropdown menu with "Ein Testprodukt" selected.
- Titel:** A text input field containing "A test product". Below it are two more input fields for "Untertitel:" and "WWW:".
- Bemerkung:** A section with a dropdown menu for "Absatz:" set to "Standard" and a "Text:" label. Below this is a rich text editor toolbar with various icons for text formatting (bold, italic, underline, etc.) and a text area containing the text "This is an english description".

Abbildung 4.10:
Übersetzung eines
Produkts

In der zweiten Combobox wählen Sie den Datensatz, also das Produkt, das Sie auf einer fremdsprachigen Seite anbieten möchten. In der Combobox stehen sämtliche Produkte des SysOrdnern zur Auswahl. Wünschenswert

wäre es, dass die Einträge aus dem Produkt zunächst übernommen würden, sobald man das Produkt in der Combobox auswählt; das ist bis einschließlich Version 2.6.0 aber leider nicht so. Der Shopadministrator muss also jedes Produkt in jeder Sprache komplett neu anlegen, was natürlich recht fehleranfällig ist.

Payment

Im TS-Setup des Shop-Template müssen Sie nun dem Payment Ihre Sprachen zuweisen, wie der folgende Code beschreibt:

```
plugin.tt_products.payment >
plugin.tt_products.payment {
    radio = 1
    TAXpercentage = 19
    10.title = Vorkasse
    20.title = PayPal
    30.title = Nachnahme
    30.percentOfTotalShipping = 2
}

[globalVar = GP:L = 0]

[globalVar = GP:L = 1]
plugin.tt_products.payment {
    10.title = cash in advance
    30.title = cash on delivery
}

[globalVar = GP:L = 2]
plugin.tt_products.payment {
    10.title = payement anticipé
    30.title = remboursement
}

[GLOBAL]
```

Shipping

Im TS-Setup des Shop-Template müssen Sie nun dem Shipping Ihre Sprachen zuweisen:

```
plugin.tt_products.shipping >

plugin.tt_products.shipping {
    radio = 1
    TAXpercentage = 19
```

```

10.title = Deutschland
10.image.params = align=absmiddle
10.price.type = count
10.price.1 = 2.5
10.price.20 = 4.8
10.price.50 = 10
10.price.100 = 25
10.price.120 = 30
10.price.300 = 150
10.percentOfGoodstotal = 0

20.title = Europa (nicht Deutschland)
20.image.params = align=absmiddle
20.price.type = count
20.price.1 = 8.8
20.price.20 = 11.8
20.price.50 = 11.8
20.price.100 = 25
20.price.120 = 30
20.price.300 = 150
20.percentOfGoodstotal = 0

30.title = Selbstabholung
30.price.1 = 0
}

[globalVar = GP:L = 0]

[globalVar = GP:L = 1]
plugin.tt_products.payment {
  10.title = Germany
  30.title = Europe (not Germany)
}

[globalVar = GP:L = 2]
plugin.tt_products.payment {
  10.title = Allemagne
  30.title = Europe (ne pas Allemagne)
}

[GLOBAL]

```

Tracking

Auch das Tracking (**Bestellung: Verwaltung**) ist zu übersetzen. Im TS-Setup des Shop-Template weisen Sie die lokalen Sprachen folgendermaßen zu:

```

plugin.tt_products.statusCodes {
# 1: Eingehende Bestellungen
  1 = Bestellung vom Käufer versendet
# 2-49: Vom Shop Administrator verwendet

```

```
    2 = Bestellung erhalten und vom Shop angenommen
    10 = Der Shop wartet auf die Lieferung der Waren von Dritten
    11 = Der Shop wartet auf die Bezahlung durch den Kunden
    12 = Der Shop wartet auf Materialien vom Kunden
    13 = Die Bestellung ist bezahlt worden
    20 = Die Waren sind an den Kunden geschickt worden
    21 = Die Geschenkgutscheine sind an den Kunden geschickt worden
    30 = Andere Meldungen vom Shop
# 50-99: Verwendbar durch den Kunden
# 50-59: Allgemeine Kunden Nachrichten
    50 = Kunden Anfrage nach Stornierung
    51 = Mitteilung vom Kunden an den Shop
# 60-69: Spezielle Kunden Nachrichten
    60 = Sende Geschenkgutschein Nachrichten an den Empfänger
# 100-299 : Bestellung abgeschlossen
    100 = Bestellung versendet und geschlossen
    101 = Bestellung geschlossen
    200 = Bestellung storniert
}

[globalVar = GP:L = 0]

[globalVar = GP:L = 1]
  plugin.tt_products.statusCodes {
# 1: Incoming orders
    1 = Order submitted by user
# 2-49: Useable by the shop admin
    2 = Order is received and accepted by store
    10 = Shop is awaiting goods from third-party
    11 = Shop is awaiting customer payment
    12 = Shop is awaiting material from customer
    13 = Order has been payed
    20 = Goods shipped to customer
    21 = Gift certificates shipped to customer
    30 = Other message from store
# 50-99: Useable by the customer
# 50-59: General user messages
    50 = Customer request for cancelling
    51 = Message from customer to shop
# 60-69: Special user messages
    60 = Send gift certificate message to receiver
# 100-299 : Order finalized.
    100 = Order shipped and closed
    101 = Order closed
    200 = Order cancelled
  }

[globalVar = GP:L = 2]
  plugin.tt_products.statusCodes {
# 1: Commande enregistrée
    1 = Commande envoyé
# 2-49: Utilisé de la boutique administrateur
```

```

    2 = Commande arrivé et accepté du boutique
    10 = Boutique attend produits d'autrui
    11 = Boutique attend paiement du client
    12 = Boutique attend choses du client
    13 = Commande païé
    20 = Produits envoyé chez le client
    21 = Cadeau bons envoyées chez le client
    30 = Autres messages du boutique
# 50-99: Utilisé du client
# 50-59: Generaux messages du client
    50 = Client demande l'interruption du commande
    51 = Messages du client par la boutique
# 60-69: Messages du client speciaux
    60 = Envoyer messages du cadeau bon par les récepteurs
# 100-299 : Commandes finies.
    100 = Commande envoyé et fermé
    101 = Commande fermé
    200 = Commande annulé
}

[GLOBAL]

```

Sie können nach Belieben weitere Meldungen einfügen, wobei Sie die Nummern der Meldungen in dieses System einfügen müssen. Die Nummerierung ist nicht frei wählbar; die verwendeten Nummern bleiben unverändert. Lediglich die auskommentierten Bereiche können Sie erweitern und dort Kommentare für den Administrator bzw. den Kunden einfügen. Die entsprechenden Nummern und Texte sind dann in den zugehörigen Comboxen für den Shopadministrator bzw. den Kunden auswählbar.

Checkliste

- Die Sprache ist nicht korrekt gesetzt, es wird z. B. ein falscher Sprachparameter übergeben.
- In den Constants ist die Sprache nicht eingetragen.
- Das Translation Handling ist nicht korrekt – evtl. noch einmal aus dem TER updaten.
- Dem Produkt in der alternativen Seitensprache ist keine Sprache zugewiesen; in der Listenansicht sollte das Feld **Language** aus dem Fieldset ausgewählt werden, dann sieht man, ob eine und welche alternative Sprache gesetzt ist.
- Das TS-Template des Shops ist dem Haupttemplate nicht zugewiesen (nur bei verteilten TS-Templates).

5

Kapitel

Shopseiten

Die Grundinstallation wurde in den vorangegangenen Kapiteln beschrieben und abgeschlossen; sie ist Voraussetzung für die nun folgenden Erweiterungen, mit der Sie die Funktionalität des Shops ausbauen.

5.1 Shopseiten und Plugin-Objekte

Im Abschnitt 1.5.2 ab Seite 27 wurden bereits die ersten, notwendigen Shopseiten angelegt. Dieses Gerüst soll nun um weitere Seiten und Plugin-Objekte erweitert werden.

5.1.1 Listenansicht

Die Listenansicht zeigt dem Besucher der Webseite alle Produkte, die der oder die zugewiesenen SysOrdner enthalten. Diese Produktliste kann über den Kategorie-Parameter `tx_ttproducts_pi1[cat]` gefiltert werden. Das

bedeutet, dass nur die Produkte einer aus einer Kategorieliste ausgewählten Kategorie angezeigt werden. Im HTML-Template ist der Subpart `ITEM_LIST_TEMPLATE` für die Anzeige zuständig. Das entsprechende Plugin-Objekt ist **Produkte: Liste**.

Sie können auch mehrere Seiten mit Listenansichten einrichten, beispielsweise für jede Produktpalette eine eigene. Für unterschiedliche Design-Elemente oder Layouts weisen Sie einer Produktliste ein anderes HTML-Template oder Template-Suffix zu. Jedes Produkt in einer Liste hat andere Merkmale und Felder.

Sollen in einer bestimmten Listenansicht nicht alle Produkte zu sehen sein, können Sie diese in verschiedenen SysOrdnern ablegen und danach als Ausgangspunkte nur die SysOrdner angeben, in denen die anzuzeigenden Produkte liegen. Oder Sie verwenden die Kategorien zur Filterung. Dazu tragen Sie im Setup `defaultCategoryID = 12` mit der Wunschkategorie ein. Das bewirkt, dass in der Liste immer nur Produkte dieser Kategorie angezeigt werden.

Ohne Filterung sind stets alle Produkte sichtbar, die über den Constants-Eintrag `pid_list` und `recursive` ermittelt werden. Die `PIDlistDisplay`-Einstellung dient dazu, den Link aus einer Produkteinzelsicht auf die Liste richtig zu setzen, denn je nach Produkt muss der Link auf eine andere Seite führen. `PIDlistDisplay` selbst hat keinerlei Filterfunktion für Produkte in der Listenansicht.

Weitere Informationen, wie Sie Produkte auf bestimmten Seiten oder nach Produktgruppen sortiert anzeigen lassen, finden Sie ab Seite 117.

Wenn Sie die Sortierung nach Produkttiteln vornehmen möchten, fügen Sie folgende Zeile ins Setup von `tt_products` ein:

```
plugin.tt_products.conf.tt_products.LIST.orderBy = title
```

Die Sortierung ist natürlich auch nach einem anderen Kriterium wie z. B. `subtitle` oder `price` möglich.

5.1.2 Einzelsicht

Aus Gründen der Übersicht empfiehlt sich eine eigene Seite für die Einzelsicht, für die im HTML-Template der Subpart `ITEM_SINGLE_DISPLAY` zuständig ist; das entsprechende Plugin-Objekt ist im Regelfall **Produkte: Einzelsicht**.

Einzelsicht der Artikelliste

In dieser Ansicht können Sie keine Produkte in den Warenkorb legen. Diese Liste ist ein Sonderfall und im Prinzip nur geeignet, wenn Sie eine be-

sondere Shop-Konfiguration ausschließlich mit Artikeln betreiben, denn sie enthält alle Artikel, nicht die Produkte samt Varianten. Zum Verständnis: `tt_products` unterscheidet *Produkte* und *Artikel*. Während Produkte eigenständig sind und entsprechend angelegt und verwaltet werden, entsprechen Artikel den Varianten dieser Produkte; sie sind also deren Kindelemente und erben deren Eigenschaften. Im Abschnitt 4.2 ab Seite 79 finden Sie das Verhältnis von Produkten und Artikeln ausführlicher behandelt.

Im HTML-Template ist der Subpart `ARTICLE_LIST_TEMPLATE` für die Artikelliste zuständig, das Plugin-Objekt ist **Artikel: Liste**. Die Einzelansichtsseite sollte im Menü nicht angezeigt werden; setzen Sie darum den Haken bei **Im Menü verstecken**.

5.1.3 Warenkorb

Auf der Warenkorbseite wählt der Kunde seine bevorzugte Liefer- und Zahlungsmethode und erhält eine detaillierte Übersicht über die von ihm zum Kauf ausgewählten Produkte samt aller Preise (Subpart `BASKET_TEMPLATE` im HTML-Template, Plugin-Objekt ist **Warenkorb: Inhalt**). Eine ausführliche Dokumentation des komplexen Themas „Versand- und Zahlungsmethoden“ bietet Kapitel 6 ab Seite 131.

Sie können die Warenkorbseite im Menü Ihrer Webseite anzeigen lassen und damit für Ihre Kunden jederzeit abrufbar machen. Ist der Warenkorb leer und der Kunde ruft diese Seite auf, erscheint ein Text, der im Subpart `BASKET_TEMPLATE_EMPTY` des HTML-Templates zu bearbeiten ist. In unserem Beispielshop ist darüber hinaus ein entsprechender Button sowohl in der Listen- wie auch in der Einzelansicht der Produkte eingefügt; die Benutzerfreundlichkeit erhöht sich damit insbesondere für Kunden, die z. B. eine Braillezeile verwenden und andernfalls den Warenkorb mühsam über das Menü suchen und ansteuern müssten. Versuchen Sie testweise einmal ohne Maus, d. h. über die Tabulatortasten durch Ihren Shop zu navigieren. Ohne Button für die Warenkorbansicht müssten Sie die gesamte Seite durchwandern, um zum nächsten Schritt der Bestellung zu gelangen. Ist dann der Warenkorb noch nicht einmal im Menü eingeblendet, können Sie nur raten, wie und wo es weitergeht.¹

5.1.4 Kontrolle und Bezahlung

Klickt der Kunde im Frontend den Submit-Button – im Beispielshop **Adressdaten eingeben** –, gelangt er zu einem Adressformular; ist er bereits angemeldet, werden seine Adressdaten automatisch übernommen und sind hier

¹ Interessant (allerdings nur für Windows-Benutzer) ist in diesem Zusammenhang das Freeware-Produkt WebFormator (<http://www.webformator.de/>).

bereits eingetragen.² Im HTML-Template wird der Subpart `BASKET_INFO_TEMPLATE` angesprochen.

Der Kunde muss im Frontend nun via Checkbutton die AGB bestätigen (im Beispielshop **AGB bestätigen**), damit die Bestellung – auch gemäß den gesetzlichen Vorschriften – fortgesetzt werden kann.

Dann erfolgt die Kontrolle, ob der Kunde alle Pflichtfelder des Adressformulars ausgefüllt und die AGB bestätigt hat. Ist das nicht der Fall, erhält er die Aufforderung, seine Daten zu vervollständigen; den entsprechenden Text können Sie im Subpart `BASKET_REQUIRED_INFO_MISSING` anpassen. Welche Felder als Pflichtfelder definiert werden können, ist ab Seite 42 beschrieben.

Nach der Eingabe aller Daten inklusive Bestätigung der AGB wird auf die Bezahlseite (Subpart `BASKET_PAYMENT_TEMPLATE`) übergegangen. Darin sind alle Bestelldaten zusammengefasst und können vom Kunden kontrolliert werden, der von dort entweder zurück zum Warenkorb geht oder seine Bestellung abschließt.

5.1.5 Bestellung abschließen

Die Seite **Bestellung abschließen** löst zuletzt die Bestellung aus (Subpart `BASKET_ORDERCONFIRMATION_TEMPLATE`). Der Kunde erhält hier nochmals die vollständigen Bestellinformationen sowie eine Bestellnummer angezeigt. Damit ist der Vorgang abgeschlossen, und sowohl der Kunde wie auch der Shop-Administrator werden per E-Mail darüber informiert.³ Alternativ können Sie über `BASKET_ORDERTHANKS_TEMPLATE` eine einfache Mitteilung ausgeben der Art: „Ihre Bestellung wird bearbeitet.“ Aktivieren Sie dies bei Bedarf durch das Setzen der `PIDthanks` in den Constants des Extension-Templates. In der E-Mail wird aber weiterhin wie oben die gesamte Bestellinformation versendet.

5.1.6 AGB

Die Seite mit den Allgemeinen Geschäftsbedingungen Ihres Unternehmens bzw. Ihres Shops ist eine normale TYPO3-Seite, die selbst kein Shop-Plugin enthält. Sie sollte im gesamten Webauftritt sichtbar und damit aufrufbar sein, um den gesetzlichen Vorgaben Genüge zu tun. Im Beispiel ist die Seite in den Seitenbaum des Shops eingefügt und über `PIDagb` zugewiesen.

² Wir gehen hier von der in Abschnitt 3.1 ab Seite 50 beschriebenen Registrierung und der Anmeldung über die „New Login Box“ aus, nicht von dem `tt_products`-eigenen Registrierungsmodul.

³ Man kann zu jeder E-Mail weitere E-Mail-Adressen als Empfänger der Bestellbestätigung eintragen. So lassen sich, durch Komma getrennt, mehrere E-Mail-Adressen für den Shop-Administrator hinterlegen.

Es wird empfohlen, die Geschäftsbedingungen an jede Bestellbestätigung anzuhängen. Binden Sie die entsprechende Datei wie Ihr HTML-Template über die Constants von `tt_products` ein und hinterlegen Sie sie im Fileadmin-Verzeichnis:

```
plugin.tt_products{
    file.templateFile = fileadmin/templates/tt_products_css.html
    AGBattachment = fileadmin/templates/tt_products/agb.txt
}
```

Die meisten der bis jetzt beschriebenen Seiten und Elemente sind bereits im Grundlagenkapitel ab Seite 27 zur Sprache gekommen. Widmen Sie sich darum nun Seiten und Plugin-Objekten, die Ihren Shop erweitern und flexibler machen.

5.1.7 Tracking

Als *Tracking* wird hier die Möglichkeit bezeichnet, den Käufer über den Status seiner Bestellung zu informieren: Ist die Bestellung eingegangen? Wurde das Produkt versendet? Fragen dieser Art beantwortet das Tracking; es ist bei `tt_products` also ein Informationsaustausch zwischen Kunde und Shopbetreiber. Folglich muss der Shop-Administrator jede Statusänderung erfassen und mit einer entsprechenden Aktion verbinden.

Der Shop-Administrator für den Tracking-Bereich ist immer der aktive Back-endbenutzer; die Bearbeitung erfolgt im Frontend, weshalb der Bereich für den Shop-Administrator immer angezeigt wird, solange er im Backend eingeloggt ist und zur Frontendansicht wechselt. Der Frontendbenutzer (Kunde) sieht diesen Administrationsbereich nicht; er kann lediglich seinen Tracking-Code eingeben und damit die Daten zu seiner Bestellung einsehen. Für erste Tests des Tracking als Frontendbenutzer sollten Sie darum vom Backend abgemeldet sein.

Im HTML-Template aktiviert der Subpart `TRACKING_DISPLAY_INFO` die Anzeige des Tracking.

Erstellen Sie zunächst eine Seite **Bestellung: Verwaltung**, wählen Sie in den Flexforms das Plugin-Objekt **Bestellung Verwaltung**, setzen Sie den Ausgangspunkt zu Ihrem Produkte-SysOrdner und speichern Sie den Datensatz ab. Über die Constants oder den Constant Editor weisen Sie nun die PID dieser Seite zu.

```
## Bestellung Verwaltung oder Tracking
plugin.tt_products.PIDtracking = 262
```

Bei aktiviertem Tracking wird im Anschluss an die Bestellung ein Tracking-Code generiert, der der Verfolgung des Auftrags dient. Dieser Code verbirgt

sich hinter dem Marker ORDER_TRACKING_NO und wird mit der E-Mail zur Bestellbestätigung versendet. Folgt der Empfänger/Kunde dem Link in dieser E-Mail, gelangt er zur Tracking-Seite (Abbildung 5.1) und findet dort alle Informationen zu seiner Bestellung; er kann eine Nachricht an den Shop versenden oder – wenn er einen anderen Tracking-Code eingibt – zu einer anderen Bestellung wechseln, sofern er bereits mehrere Bestellungen getätigt hat. Per Haken gibt er an, ob er via E-Mail über neue Einträge (also Statusänderungen) unterrichtet werden möchte. Der versierte Kunde muss nicht unbedingt dem E-Mail-Link folgen, sondern kann mit seiner Tracking-Nummer jederzeit über die Webseite auf die Informationen zugreifen, sofern die Tracking-Seite im Menü sichtbar ist.

Abbildung 5.1:
Tracking-
Informationen

Bestellung Verwaltung >
Ihre Bestellung (260) hat folgenden Status:
 (siehe letzte Zeile für aktuellsten Stand)

| Datum/Zeit: | Code: | Aktion: | Kommentar: |
|---------------------|-------|-------------------------|------------|
| 14-10-2007 15:00 | 1 | Order submitted by user | |

Shop ADMIN:
 Order: #260

Code:

Email-Benachrichtigung:
 Wenn Sie sich bei den Status Ihrer Bestellung informieren lassen wollen sobald sich der Status geändert hat, dann überprüfen Sie bitte folgendes:

Nachricht über Ablauf des Bestellvorgangs erhalten.

Nachricht an den Shop:
 Hier können Sie uns eine Nachricht bezüglich Ihrer Bestellung schicken.

Betreff auswählen:

Bemerkung:

Der Shop-Administrator

Der Shop-Administrator ist – wie bereits erwähnt – immer der angemeldete Backendbenutzer. Im Frontend ruft er die Seite **Bestellung: Verwaltung** auf und gibt sein `update_code`-Passwort ein (Abbildung 5.2), um Bestellinformationen zu bearbeiten.

Abbildung 5.2:
Anmeldung des
Administrators im
Tracking-Modul

Das Passwort lässt sich über den Constant Editor ändern. Default ist `password` (Abbildung 5.3).

Abbildung 5.3:
Passwort ändern

Sobald sich der Administrator angemeldet hat, erhält er die Liste der Bestellungen und kann auf jede zugreifen. Dazu wählt er in der Combobox **Zu einer anderen Bestellung wechseln** die entsprechende Bestellung, bestätigt die Auswahl mit dem **Select**-Button und kann nun deren Status ändern und einen Kommentar hinzufügen (Abbildung 5.4).

Abbildung 5.4:
Ansicht für den
Shop-Administrator
im Tracking-Modul

Die Status in der Combobox **Order** können Sie Ihren Anforderungen entsprechend anpassen und im Setup beispielsweise folgenden Code eintragen:

```
plugin.tt_products {
  statusCodes {
    1 = Eingegangene Bestellung

    # 2-49: Useable by the shop admin

    2 = Ihre Bestellung ist eingegangen
    3 = Der Shop erwartet Waren eines Drittanbieters
    4 = Der Shop erwartet Ihren Zahlungseingang
    5 = Der Shop erwartet Ihre Materiallieferung
    6 = Die Bestellung ist bezahlt
    7 = Ihre Bestellung wurde versandt
    8 = Geschenkgutscheine wurden versandt
    9 = Die Waren mussten für Sie bestellt werden.

    # 50-59: Generelle User-Nachrichten
    50 = Kunde möchte Bestellung stornieren
    51 = Nachricht des Kunden an den Shop
    52 = Email für Geschenkgutscheine versenden

    # 60-69: Spezielle User-Nachrichten
    60 = Versand des Gutschein-Zertifikat an den Gutscheinempfänger

    # 100-299 : Bestellung abgeschlossen
    100 = Die Bestellung ist versandt und abgeschlossen
    101 = Die Bestellung ist abgeschlossen
    102 = Die Bestellung wurde storniert
  }
  statusDate_stdWrap.strftime= %e-%m-%y %H:%M
  orderDate_stdWrap.strftime= %e-%m-%y
}
```

Die Nachrichten 2–49 sind für den Shop-Administrator reserviert, so dass Sie hier beliebige Texte einfügen können.

Das Tracking-Modul steht natürlich auch mehrsprachig zur Verfügung. Hinweise zur Anpassung finden Sie im Abschnitt 4.5 ab Seite 85.

Externe Tracking-Systeme

Viele Shopbetreiber nutzen in ihren Shops kein eigenes Tracking, sondern integrieren das ihrer Paketdienste oder Speditionen. Im Regelfall erhält eine Lieferung dort eine Nummer, die dann der Bestellung im Shop zugeordnet wird. Bei jeder Änderung im Lieferweg wird diese Nummer gescannt und im System des Paketdienstes verarbeitet. Hat das Paket im ersten Schritt den

Shop verlassen, können Sie es im Tracking-System über die Webseite des Versenders weiter verfolgen. Nicht jeder Lieferant bietet jedoch ein derart leistungsfähiges Tracking-System an.

Checkliste

- Haben Sie die PID gesetzt?
- Haben Sie die korrekte Domain angesprochen?

Ein kleines Problem taucht bei den E-Mail-Benachrichtungen recht häufig auf: Der Link verweist auf eine Seite, die aber nicht existiert. Betrachten Sie den Code im HTML-Template, erkennen Sie den Marker für die Domain Ihrer Website. Der Link sieht wie folgt aus:

```
http://###DOMAIN###/index.php?id=###PID_TRACKING###&tracking=###ORDER_TRACKING_NO###
```

In der E-Mail sieht der Link im Klartext dann etwa so aus:

```
http://mytypo3domain.de/index.php?id=262&tracking=00-25f85b
```

Es kann also sein, dass gar keine Domain eingetragen ist und darum der Link ins Leere zeigt. Legen Sie in diesem Falle über das Weltkugelsymbol (Webmodul Liste) einen neuen Datensatz vom Typ **Domain** an. Dann sollte der Link funktionieren.

5.1.8 Gutscheine

Über das Gutscheinsystem von tt_products können Kunden Geschenkgutscheine im Shop erwerben und anderen zur Verfügung stellen. Im Shop wird der Wert des Gutscheins zunächst in „Kontopunkte“ umgewandelt und später beim Kauf mit dem Rechnungsbetrag verrechnet.

Das System der Bonuspunkte ist aber auch ohne Gutscheine einsetzbar, denn der Shop-Administrator kann jederzeit Guthaben für jeden Frontendbenutzer eintragen. In dem entsprechenden Feld von Abbildung 5.5 sehen Sie außerdem bei allen Tests, wie die Gutscheinpunkte nach dem Einlösen automatisch reduziert werden.



The image shows a rectangular box with a light blue header containing the text 'Gutscheinpunkte'. Below the header is a white input field with a thin border, containing the number '10'.

Abbildung 5.5:
Gutscheinpunkte des
Frontendbenutzers

In diesem Beispielshop soll ein Geschenkgutschein über EUR 20,- zu erwerben sein. Legen Sie eine eigene Seite für Ihre Gutscheine an, fügen Sie

das Shop-Plugin ein und weisen Sie das Objekt **Produkte: Liste Gutscheine** zu. Ausgangspunkt ist ein SysOrdner **Gutscheine**, den Sie zunächst erstellen müssen. Legen Sie nun in diesem SysOrdner ein Produkt an – denn ein Gutschein ist grundsätzlich nichts anderes als ein Produkt mit besonderen Eigenschaften – und vergeben Sie eine Artikelnummer und den Preis.

Im TS-Setup des Shop-Template müssen Sie zudem eine SQL-Where-Bedingung einfügen, damit vorhandene Produkte als Gutscheine verwendet werden können:

```
### Geschenkgutscheine SQL WHERE
plugin.tt_products.whereGift = uid = 2
```

Das bedeutet, dass das Produkt mit der UID 2 (bzw. der für Ihr System passenden) ein Gutschein ist und als solcher behandelt werden soll. Wie Sie die UID Ihrer Datensätze ermitteln, ist auf Seite 55 beschrieben. Mehrere Gutscheine können Sie mit der SQL-Where-Klausel über IN zuordnen:

```
## Geschenkgutscheine SQL WHERE
plugin.tt_products.whereGift = uid IN (2, 33, 89, 47)
```

Das bedeutet, dass Gutscheine zwar von einem Redakteur angelegt, aber nur vom Administrator durch eben diese Bedingung – also die Zuweisung der UID – freigegeben werden können.

Der Gutschein greift auf einen anderen Subpart im HTML-Template zu, nämlich `ITEM_SINGLE_GIFTS_TEMPLATE`; darum muss, wie oben beschrieben, zwingend eine eigene Seite für die Listenansicht der Gutscheine existieren.

Damit auch die Links von der Listenansicht auf die Einzelansicht und wieder zurück führen, verwenden Sie die komplexe Variante des `PIDitemDisplay`-Setup. Diese wird auch wieder über eine SQL-Where-Abfrage formuliert, allerdings nur in vereinfachter Form mit dem Gleichheitszeichen. Der Grund liegt darin, dass die Daten in der Listenansicht schon über das PHP-Programm eingelesen wurden. Es wird auf eine weitere SQL-Abfrage verzichtet. Der Einfachheit halber und wegen der höheren Ausführungsgeschwindigkeit ist hier im Moment nur eine Where-Bedingung mit einem Gleichheitszeichen erlaubt.

```
PIDitemDisplay {
  10.type = sql
  10.where = uid = 2
  10.pid = 278
}
```

Hier wurde also definiert, dass das Produkt mit der UID 2 der Listenansicht mit der PID 278 zugewiesen ist. Diese PID ist die Seiten-ID der neu angelegten Gutscheinseite.

In der Listenansicht der Gutscheine findet der Kunde nun zusätzliche Eingabefelder für seinen Namen und seine E-Mail-Adresse sowie den Namen und die E-Mail-Adresse desjenigen, der den Gutschein erhalten soll. Sofern die notwendigen Eingaben vorgenommen und gespeichert wurden, kann der Kunde seine Bestellung wie gewohnt fortführen.

Bevor ein Gutschein aktiv wird, muss der Administrator die Bestellung über das Tracking annehmen und als bezahlt bestätigen. Dazu wählt er zunächst die entsprechende Bestellung aus der Liste aus und klickt den Bestellstatus-Code `Order has been paid` in der Auswahlbox.

Der Kunde erhält daraufhin eine E-Mail-Benachrichtigung mit dem Link zur Tracking-Seite. Folgt er dem Link, muss er in der Combobox die Meldung `Nachricht Geschenkgutschein verschicken` auswählen und kann jetzt eine Nachricht an den oder die Beschenkten versenden.

Bei der Bestellung des Gutscheins im Shop hat der Kunde bereits eine Nachricht an den Beschenkten ausgelöst. Im Tracking wird eine weitere Nachricht erstellt, so dass den Empfänger des Gutscheins zwei Nachrichten erreichen. So teilt der Käufer dem Beschenkten mit, dass dieser nach der Registrierung im Shop einen Geschenkgutschein einlösen kann. Die Mail, die auch den notwendigen Gutschein-Code enthält, editieren Sie im HTML-Template im Subpart `TRACKING_EMAIL_GIFTNOTIFY_TEMPLATE`.

Der Beschenkte, der sich registriert und damit als Empfänger des Gutscheins identifiziert hat, kann den Gutschein bei seinem Einkauf einlösen. Ist er bereits registrierter Frontendbenutzer, entfällt natürlich eine neuerliche Registrierung. Auf der Warenkorbseite muss er den Gutscheincode eingeben; ist er gültig, wird der Betrag in Kontopunkten dem Konto des Frontendbenutzers gutgeschrieben. In welcher Form die Gutscheinpunkte ausgegeben werden, definieren Sie im Setup. Mit dem folgenden Code definieren Sie den Wert der Gutscheinpunkte – der Einfachheit halber sei ein Gutscheinpunkt gleich einem Euro (Preisfaktor 1); setzen Sie den Preisfaktor auf 2, ist ein Gutscheinpunkt zwei Euro wert.

```
## Gutscheinpunkte

plugin.tt_products.creditpoints {
    pricefactor = 1
}
```

Im Warenkorb (`Punkte einlösen`) kann der Käufer eingeben, wie viele seiner Gutscheinpunkte er einsetzen möchte, denn er muss nicht den kompletten Gutschein einlösen.⁴

⁴ Den aktuellen Kontostand fragt der Kunde mit seinem Bestellstatus-Code unter `Bestellung Liste` ab. Der Backendadministrator findet den aktuellen Stand im Formular des Frontendbenutzers.

In der Bestellbestätigung wird der Rechnungsendbetrag entsprechend reduziert. Dazu ist es notwendig, im HTML-Template die Marker `PRICE_TOTAL_TAX` und `PRICE_TOTAL_NO_TAX` durch `PRICE_VOUCHERTOTAL_TAX` und `PRICE_VOUCHERTOTAL_NO_TAX` zu ersetzen. Diese weisen den neuen Endbetrag – reduziert um die eingesetzten Kontopunkte – aus. Sie müssen die Marker nicht an jeder Stelle im HTML-Template ersetzen, sondern lediglich im Subpart `BASKET_ORDERCONFIRMATION_TEMPLATE`.

Sofern ein Gutschein nicht über Kreditkarte bezahlt wird, funktioniert das System. Bei der Kreditkartenzahlung ist zu berücksichtigen, dass die Punkte vom Punktekonto des Kunden erst abgezogen werden, wenn die Bestellung abgeschlossen wurde. Das ist bei der Kreditkartenzahlung nicht realisierbar, bei der zuerst die Zahlung geleistet und erst danach die Order generiert wird. In diesem Fall werden die Gutscheinpunkte zwar vom Konto des Frontendbenutzers abgezogen, aber bei der Bezahlung nicht gutgeschrieben, so dass mit Generierung der Order zwar die Punkte abgezogen werden, aber zugleich der volle Betrag bezahlt wurde.

Falls Sie das Gutscheinsystem auch in Verbindung mit Kreditkartenzahlung nutzen möchten, bietet sich zum Beispiel die Implementierung über Formulare an, denn dort wird der Zahlvorgang erst initialisiert, sobald die Bestellung abgeschlossen ist. Weitere Informationen finden Sie im Kapitel 6 ab Seite 149.

Wie Sie sicher bemerkt haben, gibt es ein weiteres Problem im Zusammenhang mit den Gutscheinen: die Mehrwertsteuer. Im Beispielshop werden Bücher mit dem reduzierten MwSt.-Satz verkauft, aber auch Produkte mit dem in Deutschland regulären Satz von derzeit 19%. Der Gutschein selbst sollte also einen Brutto-Betrag, aber nicht die enthaltene MwSt. ausweisen.

`tt_products` erlaubt aber in unserem Beispielshop keine leere MwSt., denn die Steuer ist im TS-Setup als Default-Wert über `TAXpercentage` gesetzt. Bleibt das Feld `MwSt` im Backendformular des Produkts leer, wird der Wert aus dem Setup übernommen. Gilt für ein Produkt ein anderer Satz, wird dieser ins Formular eingetragen und überschreibt dann den Default-Wert aus dem Setup. Um aber dennoch in den Genuss unserer Geschenkgutscheine zu kommen, muss `tt_products` überlistet und die Datei `lib/class.tx_ttproducts_price.php` über den Extension Manager editiert werden.

Wechseln Sie dort in die Extension `tt_products` und wählen Sie in der Combobox `Edit files`. Öffnen Sie die Datei `lib/class.tx_ttproducts_price.php`. Am besten kopieren Sie deren Inhalt in einen Editor und legen vor der Änderung eine Sicherheitskopie an. Suchen Sie nun die Stelle `function getPrice` und ergänzen Sie dort den folgenden Code:

```
//          Buch "Der TYPO3 Webshop"
           if (doubleval($taxpercentage) == -1) {
               $taxpercentage = 0;
           }
```

In der Version 2.6.0 von `tt_products` ist das bereits enthalten, Sie müssen dann lediglich die Kommentarzeichen entfernen, wenn Sie diesen Code nutzen möchten. Die Zeilen bedeuten: „Wenn du den Wert `-1` im Feld `MwSt.` findest, setze die Mehrwertsteuer auf `0`.“

Beachten Sie bitte, dass der Code überschrieben wird, sofern Sie ein Update von `tt_products` einspielen. Sichern Sie die Datei lokal, damit Sie diesen Abschnitt gegebenenfalls wieder hineinkopieren können.

Nun müssen Sie dem oder den Gutscheinen im Backendformular des Produkts noch den Wert `-1` im Feld `MwSt.` zuweisen und erhalten ein Produkt ohne ausgewiesene `MwSt.` Der Wert des Gutscheins ist nun ein Bruttobetrag.

Diese Lösung ist sicher nicht die eleganteste, macht aber deutlich, dass Sie selbst Funktionalitäten implementieren können, die der Shop per se nicht bietet. Für die Extension `tt_products` wäre es natürlich wünschenswert, wenn es hier bereits eine praktikable Lösung gäbe: So wäre es sicher sinnvoll, wenn im Produkt eine Option enthält keine `MwSt` zur Verfügung stehen würde. Auch ein eigener Produkttyp „Gutschein“, dessen Eigenschaften über das `TS-Setup` definiert werden, wäre wünschenswert, damit auch Redakteure Gutscheine anlegen können und diese nicht erst durch den Administrator bestätigen lassen müssen.

Wenn Sie in Ihrem Shop ausschließlich Produkte ohne ausgewiesene `MwSt.` verkaufen, Sie also nicht der `MwSt.`-Pflicht unterliegen, entfernen Sie im `HTML-Template` alle Marker zur Ausgabe der `MwSt.` Im `Setup` setzen Sie natürlich keinen `Default-Wert` des `Steuersatzes`, sondern explizit den `Steuersatz` auf `0`, um den `Default` von `25` zu überschreiben. In diesem Fall erübrigt sich auch die Bearbeitung der `function getPrice`.

Eine Liste der möglichen `Preismarker` finden Sie im Kapitel 4 ab Seite 69.

5.1.9 Freundschaftswerbung

Analog zum Gutscheinsystem kann „Freundschaftswerbung“ ein Anreiz für Kunden sein, in Ihrem Webshop einzukaufen. Hier werden Gutscheinpunkte zugeteilt, wenn ein Kunde erstmals einen Kauf tätigt und dabei die `E-Mail-Adresse` desjenigen angibt, der ihn für Ihren Shop geworben hat. Erweitern Sie den Code für die Gutscheine im `Extension-Template` des Shops und definieren Sie eine `Rabattstaffel`. Je mehr der Kunde bei seinem ersten Besuch kauft, desto mehr `Bonuspunkte` erhält der Werber. Der Kunde selbst erhält den eben definierten `Rabatt`, bei dem einer bestimmten Anzahl erworbener Produkte jeweils ein `Rabatt` zugeordnet ist – beachten Sie aber auch hier eventuelle `Preisbindungsregeln`!

```
plugin.tt_products.creditpoints {  
    pricefactor = 1
```

```

10.type = price
10.prod.1   = 0.02
10.prod.101 = 0.04
10.prod.501 = 0.06
}

```

5.1.10 Liste Highlights

Einzelne Produkte lassen sich durch eine Checkbox kennzeichnen, der dafür sorgt, dass sie auf einer separaten Seite angezeigt werden. Erstellen Sie beispielsweise eine Shopseite namens **Besonderheit**, fügen Sie das Plugin ein und wählen Sie **Produkte: Liste Highlights**. Alle Produkte, bei denen der Haken im Feld **Besonderheit** gesetzt ist, werden auf dieser Seite angezeigt (siehe dazu auch S. 74).

5.1.11 Liste Aktionen

Legen Sie eine neue Seite mit dem Titel **Aktion** an, fügen Sie das Shop-Plugin ein und wählen Sie **Produkte: Aktionen** aus. Auf dieser Seite erscheinen dann alle Aktionsprodukte. Weitere Informationen finden Sie im Kapitel 4 ab Seite 73. Im HTML-Template ist der Subpart `ITEM_LIST_TEMPLATE` bzw. `ITEM_SINGLE_DISPLAY` zuständig, also die normale Listen- und Einzelansicht. Wünschen Sie für Ihre Aktionsartikel ein anderes Layout, ergänzen Sie einen Template-Suffix-Subpart, wie ab Seite 122 beschrieben.

5.1.12 Liste Neue Artikel

Der Begriff „Artikel“ mag nach den Ausführungen zur `tt_products`-spezifischen Unterscheidung von „Produkt“ und „Artikel“ zu Verwirrungen führen, denn tatsächlich sind hier „Produkte“ gemeint; diese Liste funktioniert ausschließlich für Produkte, nicht für deren Varianten. Der Grund liegt darin, dass nur Produkten ein Datum zugewiesen werden kann.

Erstellen Sie eine Seite namens **Neuheiten**, fügen Sie das Shop-Plugin ein und wählen Sie den Anzeigentyp/Objekte **Liste: neue Artikel**. Im HTML-Template ist der Subpart `ITEM_LIST_TEMPLATE` für die Darstellung zuständig.

In den TS-Constants setzen Sie:

```
plugin.tt_products.newItemDays = 5
```

Das bedeutet, dass neu angelegte Produkte fünf Tage lang als Neuheiten angezeigt werden, der Default-Wert liegt bei sieben Tagen. Als neu gilt ein

Produkt ab dem Erstellungsdatum, das Änderungsdatum wird nicht berücksichtigt. Und noch einmal: In diesem Sinne „neu“ kann nur ein Produkt sein, kein Artikel. Möchten Sie also einen neuen Artikel zu einem Produkt anlegen, weisen Sie diesen dem Produkt zu; soll dann das Produkt als neu gelten, müssen Sie im TS-Setup folgenden Code ergänzen:

```
## Erweitert die newItemDays für geänderte Datensätze
plugin.tt_products.conf.tt_products.ALL {
  controlFields {
    newItemDays = tstamp
  }
}
```

Achtung: Mit diesem Code im TS-Setup legen Sie fest, dass nicht nur neue, sondern auch geänderte Produkte als neu gelten, und zwar über die Zeitspanne in Tagen, die Sie in den `newItemDays` definieren; wenn Sie nichts definieren, greift der Default von 7 Tagen. Sie können über `controlFields` mit den verschiedenen Tabellenfeldern experimentieren, die einen Zeitstempel enthalten (`tstamp`, `crdate`, `starttime`, `endtime`, `sellstarttime`, `sellendtime`).

5.1.13 Produkte Suche

`tt_products` verfügt über eine eigene, einfache Suchfunktion innerhalb des Shops; Suchergebnisse lassen sich nur in einer Listenansicht (auf einer eigenen oder einer bereits vorhandenen Seite) anzeigen. Wenn Sie eine neue Seite erstellen, weisen Sie dieser das Plugin zu und wählen als Plugin-Objekt **Produkte: Suche**. Die PID dieser Seite weisen Sie über die Constants oder den Constant Editor zu, z. B.:

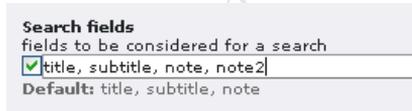
```
plugin.tt_products.PIDsearch = 259
```

Auf dieser Seite wird dann die Suchbox angezeigt. Die Ausgabe der Ergebnisse steuert der Subpart `ITEM_LIST_TEMPLATE`; sie erfolgt ebenfalls auf dieser Seite. Notwendiger Ausgangspunkt einer Suche sind der oder die Sys-Ordner mit den Produkten.

Sie können die Suche aber auch in die normale Listenansicht integrieren. Rufen Sie die Seite und das Plugin mit dem Objekt **Produkte: Liste** auf und fügen Sie das Objekt **Produkte: Suche** darunter ein; die Platzierung entscheidet darüber, wo in der Listenansicht die Suche erscheint. Soll das Suchfeld oberhalb der Listenansicht stehen, muss das Objekt entsprechend vor dem Objekt **Produkte: Liste** eingefügt werden. Nutzen Sie in der Listenansicht darüber hinaus den Miniwarenkorb (vgl. S. 110), wird dieser immer oben auf der Seite angezeigt, unabhängig davon, an welcher Position Sie das Objekt eingefügt haben.

Die Felder, die die Suche berücksichtigt, definieren Sie im Constant Editor unter **Search fields**. Im Beispiel von Abbildung 5.6 wurde zu den Standard-einstellungen das Feld `note2` hinzugefügt.

Abbildung 5.6:
Suchfelder zuweisen



Die Namen der möglichen Felder finden Sie im HTML-Template oder in der entsprechenden `tt_products`-Tabelle, die Sie über phpMyAdmin (vgl. Anhang C) ansehen können.

Falls Sie Produkte von der Suche ausnehmen möchten, legen Sie diese in einen SysOrdner, der nicht in die Suche eingebunden werden darf. Achtung: Wenn Sie mehrere Seiten für die Listenansicht verwenden, denen Sie einzelne Produkte-SysOrdner zugewiesen haben, und die Suche in diese Listenansicht integrieren, werden nur die Produkte gefunden, die auch im zugeordneten SysOrdner vorhanden sind.

5.1.14 Währung Auswahl

Leider steht diese Funktion noch nicht zur Verfügung. Geplant ist hier eine Auswahl der im Shop erlaubten Währungen.

5.1.15 Warenkorb Mini

Der Miniwarenkorb zeigt im oberen Bereich der gewünschten Shopseite an, welche Produkte sich bereits im Warenkorb befinden. Der Kunde erhält damit eine Übersicht, ohne eigens in die Warenkorbansicht wechseln zu müssen. Die Anzeige ist wohl nur auf Seiten mit einer Listenansicht sinnvoll, ist aber grundsätzlich überall einsetzbar. Da der Miniwarenkorb über ein Plugin eingefügt wird, kann er nur die Produkte der Listenansicht der entsprechenden Seite anzeigen. Wechselt der Kunde auf eine andere Listenansicht, zeigt der Miniwarenkorb nichts mehr an, obwohl Produkte im Warenkorb liegen – aber eben keine Produkte dieser Listenansicht bzw. dieses SysOrdners. Stellt dies ein Problem dar, verwenden Sie sog. *Kategorien*, um Produkte in verschiedenen Listenansichten anzuzeigen. Weitere Informationen dazu finden Sie im Abschnitt 5.4 ab Seite 117.

Nachteilig wirkt sich aus, dass bei Einsatz des Miniwarenkorb der Seitencache nicht mehr verwendet wird, denn da sich der Seiteninhalt ständig ändert, muss stets die gesamte Seite neu aufgebaut werden. Bei großen Seiten, die möglicherweise intensives Rendering erfordern und stark frequentiert sind, kann das zu Geschwindigkeitsproblemen führen.

Im HTML-Template ist der Subpart BASKET_OVERVIEW_TEMPLATE für die Darstellung verantwortlich. In der Listenansicht wählen Sie als Plugin-Objekt zusätzlich zu **Produkte: Liste den Warenkorb: Mini**. Die Reihenfolge, in der Sie die Objekte auswählen, spielt in diesem Fall keine Rolle.

Natürlich können Sie Ihren Miniwarenkorb selbst gestalten. Im HTML-Template `tt_products_css.html` ist, im Gegensatz zu den Default-Templates der Extension, die den Miniwarenkorb nutzen, zusätzlich der Produkttitel gesetzt. Damit das funktioniert, haben wir uns im Warenkorb-Template bedient, den Produkttitel eingefügt und die notwendigen Subparts für die Anzeige der verschiedenen Produktnamen um den Marker `PRODUCT_TITLE` ergänzt. Der Miniwarenkorb wird bei der Darstellung nun mit einem Warenkorb-Icon versehen, das gleichzeitig der Link zum „echten“ Warenkorb ist. Zudem wurde der Text „Ihre Produktauswahl“ in den Link einbezogen, der im HTML-Template selbstverständlich geändert werden kann.

```
<!-- ###BASKET_OVERVIEW_TEMPLATE### begin -->

<div class="shop_minibasket">

  <div class="floatbox_minibasket">
    <div class="shop_minibasket_image">
      <!--###LINK_BASKET###-->###IMAGE_BASKET###
    </div>
    <div class="text_mini">Ihre Produktauswahl</p><!--###LINK_BASKET###-->
    </div>
  </div>

  <div class="anzahl_mini">
    <p>###NUMBER_GOODSTOTAL###&nbsp;Produkt(e)</p>
  </div>
  <!-- ###ITEM_CATEGORY_AND_ITEMS### begin -->
  <!-- ###ITEM_CATEGORY### begin -->

  <!-- ###ITEM_CATEGORY### end -->
  <!-- ###ITEM_LIST### begin -->
  <!-- ###ITEM_SINGLE### begin-->

  <p> ###PRODUCT_TITLE###:

  <!-- ###ITEM_SINGLE### end -->
  <!-- ###ITEM_LIST### end -->
  <!-- ###ITEM_CATEGORY_AND_ITEMS### end -->
</div>

  (EUR ###PRICE_GOODSTOTAL_TAX###.)</p>
</div>
</div>

<!-- ###BASKET_OVERVIEW_TEMPLATE### end -->
```

Und dies ist das Ergebnis im Frontend:

Abbildung 5.7:
Der Miniwarenkorb



Möchten Sie ein anderes Icon setzen, definieren Sie das über die Constants oder den Constant Editor: Ändern Sie den dort angegebenen Pfad zum gewünschten Bild, zum Beispiel `fileadmin/images/basket1.gif`. Soll kein Bild erscheinen, lassen Sie das Feld einfach leer. Sie müssen den Haken im Constant Editor setzen, weisen aber einen leeren Pfad zu.

Abbildung 5.8:
Warenkorb-Icon:
Pfad zum Bild
zuweisen



5.1.16 Warenkorb Eingabe Kundendaten

Da in diesem Shop die Frontendbenutzerregistrierung – und zwar ausschließlich diese – zum Einsatz kommt, sei dieser Bereich nur kurz behandelt.

Über dieses Plugin-Objekt kann der Käufer (im Frontend) seine Adressdaten eingeben. Über `requiredInfoFields` definieren Sie die Pflichtfelder. Damit im Falle einer falschen Eingabe sinnvolle Fehlermeldungen erscheinen, muss die Extension `sr_feuser_register` zusätzlich installiert sein. Die Daten eines eingeloggten Frontendbenutzers können vom System übernommen werden: Setzen Sie dazu in den Constants des Extension-Templates `lockLoginUserInfo`. Zudem ermöglicht es `editLockedLoginInfo`, diese Daten zu editieren. Allerdings muss das HTML-Template dahingehend angepasst werden, dass die Edit-Felder der Benutzerregistrierung übernommen werden.

Sollen die Felder zwar vom angemeldeten Frontendbenutzer übernommen, aber für eine Bestellung überschreibbar sein, muss das HTML-Template angepasst werden. Statt statischem HTML-Ausgabertext wie hier

```
<td>###GW1B###<strong>Email:</strong>###GW1E###</td>
<td>###GW1B#####PERSON_EMAIL#####GW1E###</td>
```

muss ein Eingabetext, also ein Input-Feld definiert sein:

```
<td>###GW1B###<strong>Email:</strong>###GW1E###</td>
<td><input type="text" name="recs[personinfo][email]" size="60"
value="###PERSON_EMAIL###"></td>
```

Der Constant-Eintrag `loginUserInfoAddress` im Extension-Template des Shops bedeutet, dass ein Adressfeld in der Bestelltabelle und keine einzelnen Felder für Straße, PLZ, Ort und Land des Frontendbenutzers verwendet werden. Wenn Sie diese reduzierte Adressausgabe wünschen, können Sie trotzdem die Daten eines Frontendbenutzers übernehmen. Die Daten aus den einzelnen Feldern werden kombiniert und in das Adressfeld geschrieben. Für das Payment lassen sich außerdem `addRequiredInfoFields` definieren, über die eine bestimmte Bezahlmethode weitere Pflichtfelder öffnet, z. B. für Kreditkarten- und Kontodaten. Der Shopbetreiber möchte beim Versand ins Ausland möglicherweise auch eine Telefonnummer des Kunden, um im Zweifelsfalle Rückfragen stellen zu können.

```
10.addRequiredInfoFields = telephone
```

5.1.17 Bestellung Rechnung

Die Seite mit dem entsprechenden Plugin-Objekt zeigt nach Eingabe des Bestellcodes, der mit der Bestätigungs-E-Mail versendet wurde, eine Art „Pro-forma-Rechnung“ als HTML-Seite an, die jedoch nicht als offizieller Beleg im Geschäftsverkehr dienen sollte. Eine Anbindung an ein gängiges Faktura- oder Warenwirtschaftssystem bietet `tt_products` derzeit nicht.

Die vom System generierte HTML-Seite muss als HTML-Template im Verzeichnis `fileadmin` gespeichert sein; die notwendigen Verzeichnisse legt `tt_products` bei der Installation an.

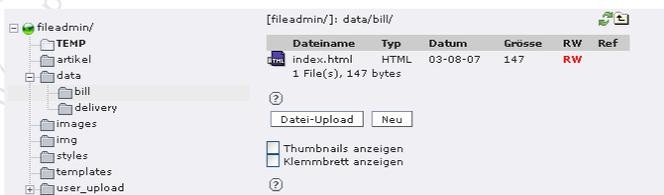


Abbildung 5.9:
Fileadmin: Ordner
für Rechnung und
Lieferschein

Erstellen Sie in Ihrem Shop eine Seite **Bestellung Rechnung**, fügen Sie das Shop-Plugin ein und weisen Sie das Objekt **Bestellung: Rechnung** zu. Über

die Constants oder den Constant Editor weisen Sie anschließend die PID dieser Seite zu.

```
## PID der Rechnungsseite  
plugin.tt_products.PIDbilling = 263
```

Auf der eben angelegten Seite erscheint nun der Link zur Rechnung. Folgen Sie diesem, erscheint im Browserfenster die Rechnung, nicht innerhalb der Webseite, jedoch im gleichen Fenster. Damit Sie nicht über den Browser-Button zurück zur Webseite gehen müssen, ist im HTML-Template ein JavaScript-Befehl (`history:back`) über den gesamten Seiteninhalt gelegt: Egal, was angeklickt wird, man gelangt zurück zur letzten Seite. Im Beispiel-Template finden Sie diesen Link im Subpart `BILL_TEMPLATE` und können ihn dort entfernen.

5.1.18 Bestellung Lieferschein

Die Seite mit dem entsprechenden Plugin-Objekt zeigt dem Kunden nach Eingabe des Bestellcodes einen Lieferschein an. Dieser muss als HTML-Template im Verzeichnis Fileadmin gespeichert sein. Dieses Modul funktioniert analog zum zuvor beschriebenen **Bestellung Rechnung**. Das Verfahren sei hier der Vollständigkeit halber beschrieben. Die PID muss auch hier wieder der Seite zugewiesen werden:

```
## PID der Lieferscheinseite  
plugin.tt_products.PIDdelivery = 264
```

Der entsprechende Subpart im HTML-Template des Shops ist `DELIVERY_TEMPLATE`. Es ist nicht unbedingt notwendig, eine eigene Seite für Rechnung und Lieferschein zu erstellen. Sie können die Plugin-Objekte auch auf Ihrer Tracking-Seite einfügen und dort anzeigen lassen. In diesem Fall ist wieder die Reihenfolge wichtig, in der Sie die Plugin-Objekte einfügen. Wenn Sie die Links zu Rechnung und Lieferschein oberhalb des Tracking anzeigen lassen möchten, setzen Sie diese Objekte zuerst ein.

Im Fileadmin-Verzeichnis für Rechnungen und Lieferscheine wird aus der `Index.html` in Verbindung mit dem jeweiligen Subpart im HTML-Template des Shops eine HTML-Datei erzeugt, deren Name dem Tracking-Code der Bestellung entspricht. Das bedeutet, dass für jede Bestellung, deren Rechnung und Lieferschein vom Kunden angeschaut werden, eine HTML-Datei generiert wird, und zwar genau in dem Moment, in dem die Seite angeklickt wird.

5.1.19 Bestellung Liste

Diese Seite bietet dem Kunden, sobald er sich im Frontend angemeldet und damit identifiziert hat, eine Übersicht über all seine Bestellungen.

Erstellen Sie eine Seite **Bestellung Liste**, fügen Sie das Shop-Plugin ein und wählen Sie das Plugin-Objekt **Bestellung; Liste**. Den SysOrdner mit den Produkten brauchen Sie hier nicht zuzuweisen. Der angemeldete Frontendbenutzer sieht, wenn er diese Seite aufruft, eine Liste seiner Bestellungen und – sofern vorhanden – seine Kontopunkte. Im HTML-Template wird das `ORDERS_LIST_TEMPLATE` verwendet. Hier kann er den Link zu einer bestimmten Bestellung anklicken und sie einsehen. Es finden sich auch die entsprechenden Links zu Rechnung und Lieferschein. Da in diesem Beispiel **Bestellung, Rechnung und Lieferschein** auf derselben Seite angezeigt werden, führen auch die Links alle auf diese eine Seite.

Wenn Sie weder Rechnung noch Lieferschein verwenden möchten, müssen Sie diese Links aus dem HTML-Template entfernen.

```
[<a href="index.php?id=###PID_BILLING###&tracking=###TRACKING_CODE#
##">Rechnung</a>]
[<a href="index.php?id=###PID_DELIVERY###&tracking=###TRACKING_CODE
###">Lieferschein</a>]
```

5.1.20 Allgemein Memo

Unter Memo bzw. „Merkzettel“ versteht tt_products eine Seite, auf der ein Shopbesucher Produkte vormerken kann, die er (noch) nicht in den Warenkorb legen möchte. Diese Seite steht ihm auch in späteren Sitzungen wieder zur Verfügung, sofern er die Inhalte nicht explizit löscht.

Erstellen Sie die entsprechende Seite mit dem Titel **Merkzettel**, fügen das Shop-Plugin ein und wählen unter **Anzeigentyp/Objekte** den Punkt **Allgemein: Memo**. Ausgangspunkt sind der oder die Produkte-SysOrdner. In den Constants muss die PID der Seite gesetzt werden:

```
## MEMO - Seite
plugin.tt_products.PIDmemo = 249
```

Bearbeiten Sie nun die Seiteneigenschaften und wählen Sie **Bei jedem Login anzeigen**. Das ist wichtig, weil der Merkzettel nur registrierten Benutzern zur Verfügung steht. Klickt der Besucher die Seite, d.h. den Menüpunkt **Merkzettel** an, erscheint die Meldung „Sie haben keine Produkte vormerkert“. Nachdem er jedoch per Klick auf das Merkzettel-Icon ein Produkt ausgewählt hat, erscheint dieses auf der Merkzettel-Seite, deren Inhalt vom Benutzer auch jederzeit wieder gelöscht werden kann (**Von der Merkliste streichen**). Der Shop-Administrator sollte allerdings darauf achten, dass die über dieses Verfahren im System gespeicherte Datenmenge nicht ausuferet.

Im HTML-Template befindet sich der HTML-Code (ITEM_LIST_TEMPLATE) für den Merktzettel sowie der Link zum zugehörigen Icon, die Sie nach Ihren Vorstellungen verändern können.

Da der Merktzettel nur für registrierte Benutzer einsetzbar ist, soll das Icon auch nicht angezeigt werden, wenn der Benutzer nicht angemeldet ist. Dazu müssen Sie den Subpart um den Merktzettel „herumwickeln“, wie im Codebeispiel zu sehen:

```
<!-- ###FE_GROUP_1_TEMPLATE### begin -->

<input type="image" name="addtomemo" class="addtomemo"
src="###PATH_FE_REL###/res/icons/fe/addmemo.png"
value="auf die Merkliste"
onClick="document.getElementsByName('###FORM_NAME###')[0].action='#
###FORM_MEMO###&tx_ttproducts_pil[addmemo]=###PRODUCT_UID###';"
title="Add to memo">

<!-- ###FE_GROUP_1_TEMPLATE### end -->
```

5.1.21 Allgemein Hilfe

Erstellen Sie eine Seite mit diesem Plugin-Objekt, erhalten sie im Frontend eine Hilfeseite für den Shop-Administrator. Sofern Sie diese Seite für Ihre Administratoren nutzen möchten, sollten Sie den Zugriff dazu auf Administratoren beschränken, damit Ihre Kunden sie nicht sehen.

5.1.22 Adressen Auswahl

Dieses Modul setzt den Einsatz von tt_address voraus. Die hier eingetragenen Adressen – z. B. die der Hersteller – kann tt_products auslesen, Ihren Produkten zuordnen und in der Einzelansicht der Produkte anzeigen. Für die Listenansicht steht dieses Feature nicht zur Verfügung, was darin begründet liegt, dass bei jedem Listenaufbau eine SQL-Abfrage notwendig und damit der Seitenaufbau stark verlangsamt wäre.

Erstellen Sie zunächst einen SysOrdner für Ihre Adressen – beispielsweise tt_address Inhalte – und darin einen Seiteninhalt mit dem Plugin Adressen. Ein Ausgangspunkt ist nicht zu setzen, da Seite und SysOrdner identisch sind.

Legen Sie nun einen neuen Datensatz in diesem SysOrdner an, wählen Sie aus der Liste Adresse und geben Sie eine Adresse ein. Wechseln Sie anschließend in den SysOrdner mit Ihren Produkten und weisen Sie dort einem Produkt über die Combobox die eben erstellte Adresse zu.

Im Frontend kann eine solche Adresse in der aktuellen Version von tt_products noch nicht sinnvoll angezeigt werden. Einige Shops nutzen das Verfahren jedoch für ein eigenständiges Menü; in einer eigenen Produktliste sind dann beispielsweise sämtliche Produkte eines Herstellers abrufbar.

5.2 SysOrdner

Wie bereits erläutert, können Sie nur einen oder aber verschiedene SysOrdner für Ihre Produkte verwenden, wobei Sie darauf achten müssen, sämtliche SysOrdner entsprechend in die Shopseiten einzubinden (vgl. Kapitel 1 ab Seite 24). In einem Buchshop könnte man z. B. für jede Sparte (Computerbücher, Belletristik, ...) eine eigene Listenansicht und entsprechende SysOrdner erstellen und diese dann einander zuordnen. In allen Ansichten, die für sämtliche Titel identisch sind (Einzelansicht, Warenkorb etc.) sind dann alle SysOrdner als Ausgangspunkte einzutragen. Achtung: Beachten Sie, dass der Miniwarenkorb in einer solchen Konfiguration nicht einsetzbar ist, da er als Seiteninhaltelement einer bestimmten Listenansicht und damit auch einem SysOrdner zugewiesen wurde. Eine Lösung bieten die in Abschnitt 5.4 beschriebenen Kategorien.

5.3 Shops auf unterschiedlichen Seiten

Wie bereits mehrfach erwähnt, können Sie Ihren Shop auf verschiedenen Seiten Ihres Internetauftritts platzieren. Sie sind nicht gezwungen, eine Shop-Hauptseite anzulegen und nur innerhalb dieser zu arbeiten. Auf jeder beliebigen Seite können Sie Shop-Plugins einfügen, SysOrdner zuweisen und anzeigen lassen. Zur Sicherheit sollten Sie beim Verweis auf den SysOrdner `recursive` auf unendlich setzen, damit sämtliche Unterseiten und SysOrdner einbezogen werden.

5.4 Kategorien

Unter *Kategorie* versteht man eine – wie auch immer motivierte – Gruppierung von Produkten, bei Computerbüchern also etwa eine nach Themen („Web-Design“, „Betriebssysteme“), nach dem Wissensstand der Leser („Einsteiger“, „Profi“) oder Ähnliches. Jedes Produkt müssen Sie dann einer solchen Kategorie explizit zuordnen. Auch Seiten im Seitenbaum können als Kategorien dienen, sofern Sie dies im Extension Manager definiert und die Produkte auch auf diesen Seiten gespeichert haben.

Ein mögliches Szenario für den Einsatz von Kategorien ist im Folgenden beschrieben. Im Beispielschop sollen Produkte einer Kategorie auf einer ei-

genen Seite angezeigt werden: Produkte der Kategorie 1 also auf der Seite **Kategorie 1**, Produkte der Kategorie 2 auf der Seite **Kategorie 2**.

Je mehr Produkte Sie in einem Shop haben, desto schwieriger wird es, diese unter vielen anderen Produkten herauszusuchen. Ein gut strukturierter Kategoriebaum erlaubt es dem Käufer, im Shop zu stöbern, ohne alle Produkte einzeln auflisten zu müssen. Eine Navigation in einem Shop ist nur über Hierarchien oder eine Produkt-Suche möglich.

Viele neue Attribute können über sog. *Mehrfachkategorien* indirekt angelegt werden, was bedeutet, dass ein Produkt mehr als einer Kategorie angehört.

Allerdings muss die Anzahl der Kategorien deutlich unter der Anzahl der Produkte liegen. Wer ruft schon gerne eine Kategorie auf, um dann eine Meldung der Art „Keine Produkte in dieser Kategorie gefunden“ angezeigt zu bekommen? Die Kategorien sollten möglichst sprechende Namen haben und den Benutzer damit schnell zum Ziel führen. Setzen Sie auch Alternativtexte bei der Anzeige der Kategorien ein, die Details über die Kategorien geben. Das ist umso wichtiger, je höher eine Kategorie im Hierarchiebaum steht, denn der Benutzer will schon zu Beginn seiner Suche den richtigen Weg einschlagen.

5.4.1 mbi_products_categories

In `tt_products` wird von Haus aus nur eine einfache Kategorie je Produkt mitgeliefert. Erst über die Zusatz-Extension `mbi_products_categories` können hierarchische Kategorien aufgebaut werden. Sie sollten also schon beim Aufbau eines Shops dessen mögliche Komplexität in Betracht ziehen und gegebenenfalls mit zwei oder mehr unabhängigen Kategoriebäumen arbeiten. Wenn Sie mit Seiten als Kategorien beginnen, müssten Sie beim Umstieg auf Produkt-Kategorien alle Kategorien neu eingeben.⁵

Abbildung 5.10:
PID des SysOrdnerns
für die Kategorien



⁵ Unter <http://ttproducts.de/> gibt es die `mbi_products_categories` in Version 0.2.0. Damit können jedem Produkt beliebig viele Kategorien zugeordnet werden.

Wechseln Sie zunächst in den Extension Manager und installieren Sie **Shop Categories**. Anschließend tragen Sie die PID des SysOrdnerns, in dem die Kategorien angelegt werden, in das Feld **Page Id List** ein (Abbildung 5.10) und bestätigen mit einem Klick auf den **Update**-Button.

In diesem Beispiel handelt es sich um den SysOrdner mit der PID 288, in dem bereits die Produkte verwaltet werden. Dort sollen auch die Kategorien angelegt werden.

Verlassen Sie den Extension Manager und wechseln Sie ins Webmodul **Liste** und den SysOrdner mit den Produkten, der im Beispielshop die PID 288 hat. Hier werden nun die Kategorien angelegt. Wählen Sie **Neuen Datensatz anlegen | Produkt Kategorie** aus und erstellen Sie eine Kategorie mit Namen **Kategorie 1** (Abbildung 5.11). Wiederholen Sie den Vorgang für Kategorie 2. Jeder Kategorie lassen sich übrigens weitere Unterkategorien zuordnen.

Eine Kategorie kann für mehrere andere Kategorien als Oberkategorie zugewiesen werden, nicht umgekehrt. Achten Sie allerdings darauf, dass eine Oberkategorie nicht eine unter ihr liegende Kategorie selbst als Oberkategorie auswählen darf, da im Programmcode alle Kategorien von unten nach oben durchlaufen werden und somit Endlosschleifen entstehen würden.

The screenshot shows a web form for creating a product category. The form is titled "Produkt Kategorie [28] - Kategorie 1". It has several sections:

- Titel:** A text input field containing "Kategorie 1".
- Untertitel:** An empty text input field.
- Bestätigungsemail:** A dropdown menu with a blue arrow icon.
- Bild:** A file selection area with a folder icon, a text input field, and a "Durchsuchen..." button. Below it, a list of supported image formats is shown: GIF, JPG, JPEG, TIF, BMP, PCX, TGA, PNG, PDF, AI.
- Oberkategorie:** A section with two sub-sections:
 - Ausgewählt:** An empty text input field.
 - Objekte:** A list box containing one item: "Produkt Kategorie" with a small globe icon.

Abbildung 5.11:
Kategorie erstellen

Diese Kategorien weisen Sie nun je einem Produkt zu: Wählen Sie ein passendes Produkt aus und weisen Sie über die Auswahlbox die **Kategorie 1** (Abbildung 5.12) zu.

Abbildung 5.12:
Kategorie zuweisen

The screenshot shows a product configuration form with the following fields and values:

- Name:** Ein Testprodukt
- Untertitel:** (empty)
- WWW:** mytypo3domain.de
- Artikel Nr.:** 123456
- Am Lager (St.):** 1
- Kategorie:**
 - Ausgewählt: Kategorie 1
 - Objekte: Produkt Kategorie, Kategorie 1, Subkategorie 1, Kategorie 2

Auch hier wiederholen Sie den Vorgang für andere Produkte und Kategorien. Ein Produkt mehreren Kategorien zuzuordnen ist in dieser Konstellation leider nicht möglich.

Erstellen Sie eine Seite **Kategorie 1**, auf der nur Produkte angezeigt werden, denen die Kategorie 1 auch zugewiesen wurde (entsprechend eine Seite für Kategorie 2). Fügen Sie jeweils das Shop-Plugin in die Seite ein und weisen Sie als Anzeigeobjekt **Produkte: Liste** zu. Ausgangspunkt ist der SysOrdner mit Produkten und Kategorien.

Damit die Seite auch „weiß“, welche Kategorie hier zu sehen sein soll, muss ein wenig Code ergänzt werden, und zwar erhält jede Kategorie-Seite ein eigenes Extension Template Setup. Dort muss nur eine einzige Zeile stehen.

Erstellen Sie also ein solches Extension-Template über **Neuen Datensatz anlegen** auf der Seite **Kategorie 1**, vergeben Sie einen Namen und tragen Sie folgenden Code in das Setup der Seite ein:

```
plugin.tt_products.defaultCategoryID = 28
```

So weisen Sie der Seite die Kategorie mit der UID 28 zu, so dass nur Produkte dieser Kategorie aufgelistet werden. Die Produktlisten werden somit über die Kategorie gefiltert. Sofern Sie auf dieser Seite weitere Kategorien anzeigen möchten, tragen Sie die UIDs durch Komma getrennt ein.

5.4.2 Seiten als Kategorien

Legen Sie die Produkte auf den Seiten oder SysOrdnern im Seitenbaum an. Im Extension Manager müssen Sie `Use Page as Category [pageAsCategory]` auf 1 setzen. Das bewirkt, dass die Seite, auf der das Produkt abgespeichert wird, die Oberkategorie ist. Die Produktkategorie ist die Hauptkategorie. Sie erhalten damit hierarchisch geordnete Kategorien, die sich einfach im Seitenbaum des TYPO3-Backends aufbauen lassen. Der Seitenbaum entspricht dabei dem Kategoriebaum. Diese Seiten werden normalerweise nie im Frontend angezeigt, was grundsätzlich aber möglich wäre. Setzen Sie `pageAsCategory` gleich 2, wird die Produktkategorie ignoriert und es werden nur noch die Seiten als Kategorien verwendet. Das ist etwas übersichtlicher und einfacher. Sie müssen die Produkte nicht mehr neben dem Abspeichern im Seitenbaum auch noch eigens einer Kategorie zuordnen. Nur diese Methode wird im Programmcode für die Kategorielisten und -selectboxen derzeit unterstützt.



Abbildung 5.13:
Seiten als Kategorien
definieren

Wechseln Sie in die Extension `tt_products` und tragen Sie in das Feld `Use Page as Category` den Wert 1 ein. Das bedeutet, dass die Hauptkategorie eine normale Kategorie ist (vgl. den vorherigen Abschnitt). Auch diese Eingabe bestätigen Sie mit einem Klick auf den `Update`-Button (Abbildung 5.14).

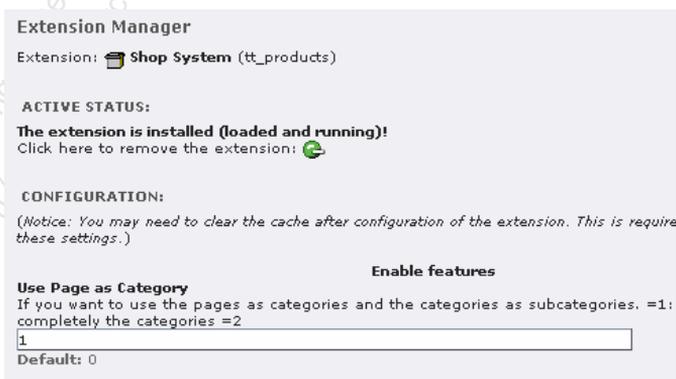


Abbildung 5.14:
Seiten als Kategorien
definieren

5.5 Benutzergruppen und Template-Suffixe

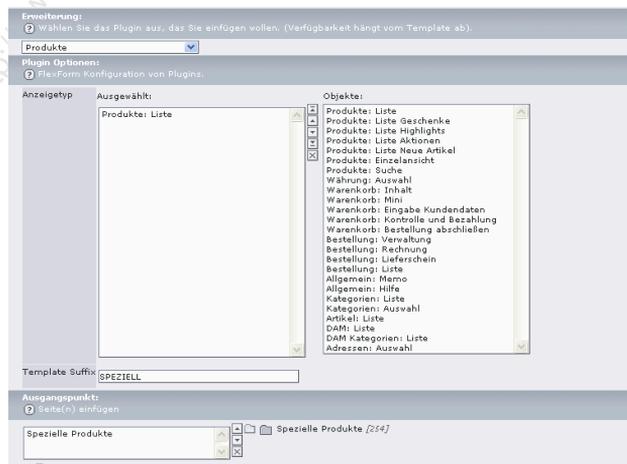
Mit `tt_products` können Sie den Zugriff und damit die Bestellmöglichkeit von Produkten für bestimmte Kundengruppen definieren. Kapitel 6 zu Versand- und Zahlungsmethoden beschreibt beispielsweise, wie man ausgewählten Gruppen oder Personen Sonderkonditionen einräumt.

Folgende Situation: Sie möchten Produkte, die nicht an Personen unter 18 Jahren ausgegeben werden dürfen, über Ihren Shop verkaufen, z. B. alkoholische Getränke, Schneidwaren, Medikamente etc. Sie *müssen* also sicherstellen, dass Ihr Kunde volljährig ist, bzw. dürfen andernfalls die Bestellung nicht erlauben; die Ansicht der Produkte ist aber sehr wohl erlaubt.

Umgesetzt wird dieses Szenario über Benutzerrechte in Kombination mit Template-Suffixen: Legen Sie zunächst eine Benutzergruppe **Speziell** an und weisen Sie einen Ihrer Testbenutzer dieser Benutzergruppe zu. Diesem Benutzer soll später der Zugriff auf die Produkte, die bestimmten Auflagen unterliegen, möglich sein.

Legen Sie eine Seite **spezielle Produkte** an, fügen Sie das Shop-Plugin ein und wählen Sie den Anzeigentyp/Objekte **Produkte: Liste**. In das Feld **Template Suffix** tragen Sie **SPEZIELL** ein. Wichtig ist, dass Sie hier ausschließlich Versalien benutzen. Weisen Sie diesem Plugin die Benutzergruppe **Speziell** über **Zugriff** zu und speichern Sie den Datensatz (Abbildung 5.15 und Abbildung 5.18).

Abbildung 5.15:
Template-Suffix
einfügen



Legen Sie einen **Produkte-SysOrdner** und darin ein **Testprodukt** an und weisen Sie der eben erstellten Seite diesen SysOrdner als **Ausgangspunkt** zu. Für einen „Echtbetrieb“ muss dieser SysOrdner natürlich noch den Shopseiten, die für die Bestellung relevant sind, zugewiesen werden.

Anschließend muss das HTML-Template um einen weiteren Subpart für die Listenansicht erweitert werden: eben das sog. *Template-Suffix*. Ein Template-Suffix ist nichts anderes als ein spezieller Subpart. Falls Sie das HTML-Template `tt_products_css.html` verwenden, entfällt der nächste Schritt, denn dort ist dieser Subpart bereits eingefügt.

Wechseln Sie in das HTML-Template Ihres Shops und kopieren Sie den Subpart `###ITEM_LIST_TEMPLATE###`. Danach fügen Sie ihn direkt hinter dem Ende dieses Subparts im selben Template wieder ein. Benennen Sie diesen neuen Subpart in `ITEM_LIST_TEMPLATE_SPEZIELL` um, und zwar sowohl zu Beginn wie auch an dessen Ende.

Entfernen Sie innerhalb des neuen Subparts den Bereich `<!-- ###FE_GROUP_1_TEMPLATE### begin -->` bis `<!-- ###FE_GROUP_1_TEMPLATE### end-->` und fügen ihn um den Div-Tag `<div id="anzahl_LIST">...</div>` wieder ein.

Der Code für dieses Template-Suffix sollte damit wie folgt aussehen.

```
<!-- ###ITEM_LIST_TEMPLATE### end -->

<h3>ITEM_LIST_TEMPLATE_SPEZIELL</h3>
<em>This subpart is used to display the regular list of products.
  It's also used by search-results.</em>

<!-- ###ITEM_LIST_TEMPLATE_SPEZIELL### begin
  This subpart is used to display the regular list of products.
  It's also used by search-results.-->

  <!--###LINK_PREV###-->###GW2B###<div
  align="left"><b>vorherige Seite</b>###GW2E###
  </div><!--###LINK_PREV###-->
  <!-- ###LINK_BROWSE### begin -->[ ###BROWSE_LINKS### ]<!--
  ###LINK_BROWSE### end-->
  <!--###LINK_NEXT###-->###GW2B###<div
  align="right"><b>nächste Seite</b>###GW2E###</div><!--
  ###LINK_NEXT###-->

  <!--###GW2B###Produktauswahl:###GW2E### -->
  <!-- ###ITEM_CATEGORY_AND_ITEMS### begin -->
    <!-- ###ITEM_CATEGORY### begin -->

    <div id="CATEGORY_TITLE_LIST" align="left">
    ###GW1B#####CATEGORY_TITLE### ###GW1E###
  </div>

  <!-- ###ITEM_CATEGORY### end -->

  <!-- ###ITEM_LIST### begin -->
  <!-- ###ITEM_SINGLE### begin-->
```

```

<!-- ###ITEM_SINGLE_PRE_HTML###-->

<form method="post" action="###FORM_URL###" name="###FORM_NAME###">

<div id="PRODUCT_TITLE_LIST">
  <!--###LINK_ITEM###-->###GW1B###<b>###PRODUCT_TITLE###</b>
  ###GW1E###<!--###LINK_ITEM###-->
  <p>###PRODUCT_SUBTITLE###</p>
  <!--###LINK_ITEM###-->mehr Infos<!--###LINK_ITEM###-->
</div>
<div class="floatbox_LIST">
  <div id="PRODUCT_IMAGE_LIST">
    ###PRODUCT_IMAGE1###
  </div>
  <div id="display_variant2_LIST">
    <!-- ###display_variant2### -->
    ###GW2B### Gre: ###GW2E###
    <SELECT style="font-size: 10px"
    name="###FIELD_SIZE_NAME###">###PRODUCT_SIZE###</SELECT>
    <!-- ###display_variant2### -->
  </div>
  <div id="display_variant1_LIST">
    <!--###display_variant1### -->
    ###GW2B### Art: ###GW2E###
    <SELECT style="font-size: 10px"
    name="###FIELD_COLOR_NAME###">###PRODUCT_COLOR###</SELECT>
    <!--###display_variant1### -->
  </div>
  <div id="display_variant5_isSingle_LIST">
    <!-- ###display_variant5_isSingle### -->
    Einzelprodukt, das nur einmal gewählt werden kann.
    ###PRODUCT_ADDITIONAL_SINGLE###
    <!-- ###display_variant5_isSingle### -->
    <!-- ###display_variant5_isNotSingle### -->
    <!-- ###display_variant5_isNotSingle### -->
    <!-- ###ITEM_SINGLE### end -->
  </div>
  <div id="web_price_LIST">
    ###GW1B###Web-Preis: ###PRICE_TAX### &euro;###GW1E###
    ###GW2B###(enth. MwSt.: ###PRICE_ONLY_TAX### &euro;)###GW2E###
    ###PRODUCT_DELIVERY###
  </div>

  <!-- ###FE_GROUP_1_TEMPLATE### begin -->
  <div id="anzahl_LIST">
    ###GW1B### Anzahl: ###GW1E### <INPUT id="input_qty" size="4"
    maxlength="6" type="text" name="###FIELD_NAME###"
    value="###FIELD_QTY###">
    <input type="image" src="fileadmin/osp/images/icons/basket1.gif"
    name="basket" id="inputimage_list" alt="In den Warenkorb" />
    <input type="image" name="addtomemo"
    src="###PATH_FE_REL###/res/icons/fe/addmemo.png"

```

```

        value="auf die Merkliste"
        onClick="document.getElementsByName('###FORM_NAME###')[0].action='##
#FORM_MEMO###&tx_ttproducts_pil[addmemo]=##$#PRODUCT_UID###';"
        title="Add to memo">
</div>

<input type="button" id="button2_LIST" name="warenkorb"
value="zum Warenkorb"
onClick="self.location.href='index.php?id=###PID_BASKET###' " />

<!-- ###FE_GROUP_1_TEMPLATE### end -->
</div>
</form>

        <!-- ###ITEM_LIST### end -->
        <!-- ###ITEM_CATEGORY_AND_ITEMS### end -->

        <!--###LINK_PREV###-->###GW2B###<div
align="left"><b>vorherige Seite</b>###GW2E###</div><!--
###LINK_PREV###-->
        <!--###LINK_NEXT###-->###GW2B###<div
align="right"><b>nächste Seite</b>###GW2E###</div><!--
###LINK_NEXT###-->

<!-- ###ITEM_LIST_TEMPLATE_SPEZIELL### end -->

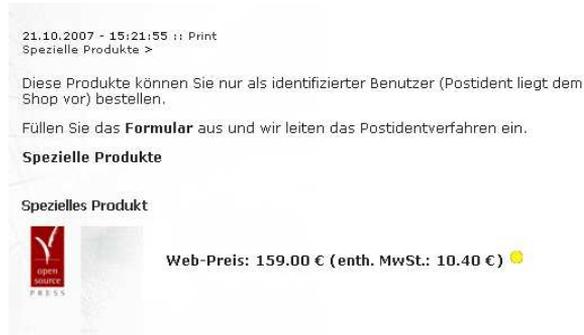
```

Ruft nun ein nicht angemeldeter Besucher die Seite **Spezielle Produkte** auf, kann er diese zwar sehen, aber nicht bestellen. Zuständig ist dafür der Subpart `FE_GROUP_1_TEMPLATE` (im Codebeispiel fett gedruckt). Nur registrierte Benutzer der Gruppe **Speziell** haben Zugriff auf die Felder zur Bestellung, die vom Subpart `FE_GROUP_1_TEMPLATE` eingeschlossen werden.

Derzeit funktioniert der Subpart `FE_GROUP_1_TEMPLATE` allerdings nur pauschal für alle angemeldeten Benutzer und kann nicht auf einzelne Benutzergruppen angewendet werden. Für zukünftige Versionen ist eine Erweiterung dieses Features geplant. Dann steht die Zahl 1 für die Benutzergruppe mit der UID 1, die 2 für die Benutzergruppe 2 usw. Dann könnte es zum Beispiel eine privilegierte Benutzergruppe für jene Personen geben, die sich bereits über das Postidentverfahren identifiziert haben (siehe dazu den folgenden Abschnitt).

Damit nicht angemeldete Kunden auch wissen, dass nicht etwa ein Fehler im Shop vorliegt, sondern die Bestellmöglichkeit mit Absicht eingeschränkt ist, erstellen Sie ein Text-Inhaltselement (**Beim Login verstecken**) mit einem entsprechenden Hinweis. Das Ergebnis im Frontend sehen Sie in Abbildung 5.16.

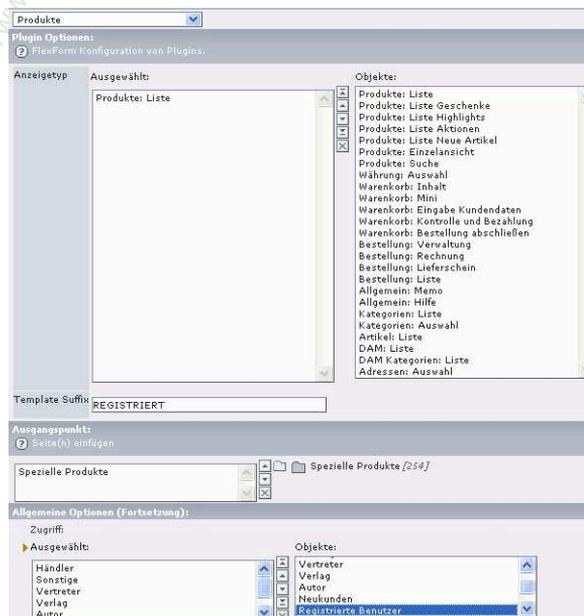
Abbildung 5.16:
Nicht angemeldete
Benutzer können
nicht bestellen



Problematisch sind damit jedoch noch alle Benutzer, die zwar registriert, aber eben nicht befugt sind, hier zu bestellen. Aber da sowohl TYPO3 als auch tt_products flexibel sind, gibt es auch hier eine – nicht sonderlich elegante, aber funktionierende – Lösung.

Informieren Sie die Besucher Ihrer Website, dass hier ausschließlich autorisierte Benutzer bestellen können, da die Produkte bestimmten Einschränkungen unterliegen. „Klonen“ Sie dazu das Textelement dieser Seite, fügen Sie es unterhalb des ersten wieder ein und weisen Sie es allen Benutzergruppen mit Ausnahme der Gruppe **Speziell** zu.

Abbildung 5.17:
Plugin mit
Templatesuffix
REGISTRIERT für
registrierte Benutzer



Fügen Sie ein weiteres Plugin des Shops ein, auf dem Sie den Zugriff für alle Frontendbenutzergruppen erlauben – mit Ausnahme der Gruppe **Speziell**. Die Sache hat allerdings einen Haken: Sie benötigen ein weiteres Template-Suffix, nämlich eines, in dem der Teil mit dem Subpart `FE_GROUP_1_TEMPLATE` nicht enthalten ist. Damit hätte dieses Template-Suffix keine Eingabefelder für die Artikelanzahl und keine Buttons, um Produkte in den Warenkorb zu legen. Im HTML-Template `tt_products_css.html` ist dieses Template-Suffix **REGISTRIERT** bereits eingepflegt. Sie müssen lediglich dem Plugin dieses Template-Suffix zuordnen. Über **Allgemeine Optionen** weisen Sie den Datensatz allen Benutzergruppen außer der Gruppe **Speziell** zu (Abbildung 5.17).

Sobald der Benutzer angemeldet ist, sieht er den „geklonten“ Text und den Subpart mit dem Suffix **REGISTRIERT**, der keine Bestellmöglichkeit bietet. Wir nutzen hier also die Sichtbarkeiteinstellungen von **TYPO3**.

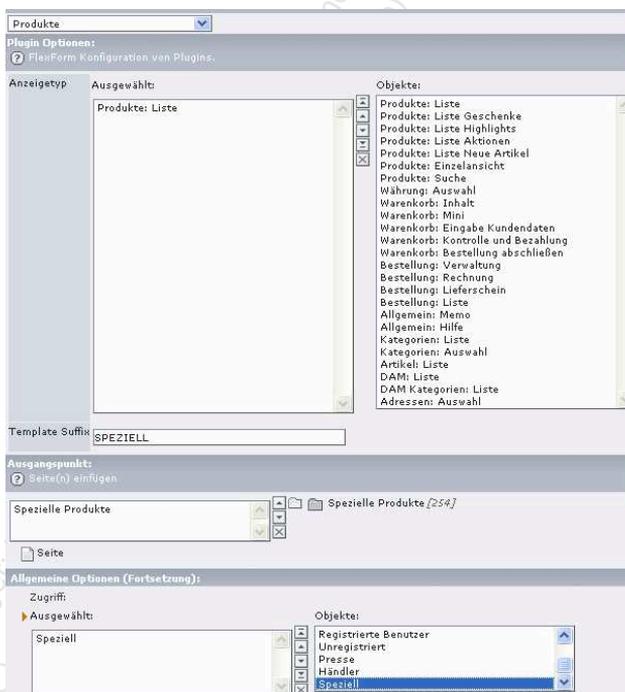


Abbildung 5.18:
Plugin mit
Template-Suffix
SPEZIELL und der
Gruppe **Speziell**

Das Ergebnis im Frontend: Benutzer, die nicht angemeldet sind, sehen den Text und die Produktliste der speziellen Produkte, ohne bestellen zu können, wie in Abbildung 5.16 zu sehen. Angemeldete Benutzer, die nicht der Gruppe **Speziell** angehören, können ebenfalls nicht bestellen. Das Ergebnis im Frontend ist identisch. Angemeldete Benutzer, die der Gruppe **Speziell** angehören, sehen den Text nicht und können einkaufen.

Sicher ist dies keine dauerhafte Lösung. Es soll Ihnen lediglich als Beispiel dienen, wie bestimmte Ansichten verschiedenen Frontendbenutzern oder Gruppen zugewiesen werden können.

5.5.1 Postidentverfahren

Das Postidentverfahren ist eine Identitätsüberprüfung der Deutschen Post AG.⁶ Es unterscheidet drei Verfahren:

Postident Basic

eindeutige Identitätsüberprüfung aufgrund persönlicher Vorstellung und Vorlage des Personalausweises

Postident Spezial

Identifizierung aufgrund identischer Unterschriften auf Originaldokumenten

Postident Comfort

Identifikation durch den Zusteller

Mit dem Verfahren *Postident Comfort* könnten Sie eigentlich darauf verzichten, Ihre Produkte nur speziellen Gruppen zuzuordnen oder als Spezialanfertigung (vgl. Seite 75) zu deklarieren, denn Sie haben ja die Möglichkeit, die Identifizierung über den Zusteller zu veranlassen. Nachteilig ist, dass Sie Ihre Produkte bereits versendet haben, wenn sich vielleicht herausstellt, dass der Empfänger nicht berechtigt ist, die Ware entgegenzunehmen.

5.6 Listenansicht in unterschiedlichen Spalten

Verwenden Sie eine Seite mit einem zwei- oder dreispaltigen Layout, kann das Shop-Plugin natürlich auch zugleich in den verschiedenen Spalten eingesetzt werden. In diesem Beispiel wurde das Plugin auf der Seite **Shop** nochmals in der rechten Spalte eingebunden. Da diese jedoch viel schmaler ist, muss die Listenansicht angepasst werden. Erstellen Sie darum einen neuen Template-Subpart mit dem Template-Suffix **MINI**, indem Sie den bereits erstellten Subpart **SPEZIELL** in Ihrem HTML-Template kopieren. Entfernen Sie die nicht benötigten Bereiche und lassen Sie nur das Bild und die beiden Überschriften sowie den Link zur Einzelansicht im Template stehen.

Möchten Sie die Schriften oder das Layout in dieser Spalte bzw. dem Subpart ändern, passen Sie die Definitionen in den **DIV**-Tags des HTML-Templates und in Ihrem Shop-CSS an.

⁶ Weitere Informationen und Formulare für das Postidentverfahren finden Sie unter <http://www.deutschepost.de/> (Suche: „Postident“).

Damit das Bild in einer passenden Größe angezeigt wird – denn auch das Produktbild in der Listenansicht sollte kleiner sein –, ändern Sie im HTML-Template zunächst den Bildmarker in `PRODUCT_IMAGE2:MINI`. Wechseln Sie in das Produkt und tragen Sie das entsprechende Bild zusätzlich ins Feld **Image** ein. Damit weisen Sie dem Produkt ein zweites Bild zu, wie bereits ab Seite 78 beschrieben. Vergessen Sie nicht, in den Constants die Anzeige eines weiteren Bildes zu aktivieren.

Kopieren Sie folgenden Code in das TS-Setup Ihres Shop-Template:

```
# Minibilder definieren

plugin.tt_products.conf.tt_products.LIST.image.mini {
  file.maxW = 40
  altText = TEXT
  altText.data = field:title
  titleText = TEXT
  titleText.data = field:title
  imageLinkWrap <
  imageLinkWrap = 0
}
```

Dieser Code definiert, dass die Bilder, die als **Bild2** eingefügt wurden und die Endung `MINI` haben, nur 40 Pixel groß angezeigt werden. Weiterhin wird der Bildtitel als Alternativtext angezeigt, sofern Sie die Anzeige von Bildern im Browser deaktivieren. Wenn Sie `TYPO3` in Version 4.2 einsetzen, sollte beim Überfahren mit der Maus ein Bildtitel, der ebenfalls aus dem Produkttitel besteht, angezeigt werden. Mit `altText.data = field:title` wird festgelegt, dass der Titel aus dem Datensatz angezeigt wird, wenn der Browser kein Bild darstellen kann. Mit `imageLinkWrap` können Sie definieren, dass ein Klick auf das Bild dieses Bild in Originalgröße darstellt. Normalerweise wird ein JavaScript-Code aktiviert, damit dieses Bild in der entsprechenden Größe dargestellt wird. In diesem Fall ist der Wert allerdings auf Null gesetzt.

Diese speziellen Anzeigeeoptionen eignen sich beispielsweise auch für das Hervorheben von Aktionen, Besonderheiten und zur Bewerbung der neuesten Produkte in Ihrem Shop.

6

Kapitel

Versand- und Zahlungsmethoden

Mit der Ausgestaltung von Versandoptionen und Zahlungsmethoden widmen wir uns nun dem sicher heikelsten Teil eines Online-Shops, denn einerseits sind in diesem Bereich die Anforderungen erfahrungsgemäß sehr vielfältig, andererseits ist beim Umgang mit Kundendaten und Finanzen natürlich besondere Sorgfalt geboten.

Auch hier werden wir uns Schritt für Schritt einer komplexen Lösung nähern und auf dem Weg grundlegende Konzepte kennenlernen.

In Kapitel 2 auf Seite 35 wurde in den Constants von `tt_products` definiert, dass nur jeweils 1 Exemplar eines Produkts im Rahmen eines Bestellvorgangs gekauft werden darf. Diese Beschränkung heben Sie nun auf, indem Sie die Anweisung auskommentieren oder löschen.

```
#plugin.tt_products.basketMaxQuantity = 1
```

Damit wird jedoch der Default-Wert 100.000 aktiv, so dass ein niedrigeres Limit hier wohl grundsätzlich sinnvoll ist.

Durch die Definition bestimmter Methoden können Sie die Versandkosten, die Sie Ihren Kunden in Rechnung stellen, an unterschiedliche Bedingungen (Stückzahl, Warenwert, Kundengruppe etc.) knüpfen und das System mehr oder weniger ausgefeilt implementieren. Um eben diese Bedingungen wird es in den folgenden Szenarien gehen, so dass Sie zuletzt die verschiedenen Konzepte umsetzen und – was noch interessanter sein dürfte – miteinander kombinieren können.

Bevor es nun an den Code geht, sei noch einmal die wichtige Unterscheidung von Versand (*Shipping*) und Zahlung (*Payment*) ins Gedächtnis gerufen, die ja auch in zwei separaten Bereichen des TS-Setup und den Auswahlmöglichkeiten für den Kunden im Frontend ihren Niederschlag findet.

Entscheidend für die gesamte Abwicklung und Berechnung eines Kaufvorgangs ist in diesem System die Auswahl des Kunden bei den *Versandoptionen*: Handelt es sich um eine Lieferung innerhalb Deutschlands, holt der Kunde die Ware selbst ab oder geht die Ware ins (außereuropäische) Ausland? – Diese Auswahl bestimmt einerseits die daraufhin möglichen Zahlungsmethoden (Barzahlung bei Selbstabholung, Vorauskasse bei Auslandslieferung o. Ä.) und natürlich die mit der Lieferung verbundenen Kosten. Auch die Behandlung der Mehrwertsteuer ergibt sich daraus, so dass beispielsweise ein Kunde mit VAT-ID in der EU die Waren umsatzsteuerfrei beziehen könnte. Mit diesem Prinzip im Hinterkopf sind die im folgenden beschriebenen Lösungen unmittelbar verständlich: Stets werden zunächst Versandoptionen der erwarteten Kunden(gruppen) und die entsprechenden Zahlungsoptionen aufeinander abgestimmt.

Damit zurück zur konkreten Umsetzung: Nehmen Sie sich zunächst den **Versand innerhalb Deutschlands** vor und verändern Sie das System im Hinblick auf eine an die bestellte Stückzahl gebundene Kostenberechnung.

6.1 Versandkosten nach Stückzahl

Wechseln Sie in das Extension-Template Ihres Shops (`++tt_products`) und kopieren Sie das Setup in einen Editor. Es empfiehlt sich, den Code zunächst lokal zu bearbeiten und eine Sicherungskopie anzulegen. Testen Sie, wenn möglich, Ihre Implementierung in einer lokalen TYPO3-Umgebung. Falls Ihnen ein Fehler unterläuft, würden Sie nicht den gesamten Online-Shop lahmlegen.

6.1.1 Versand innerhalb Deutschlands

In Bezug auf den Versand wollen wir im ersten Schritt Folgendes erreichen: Die Lieferung erfolgt nur innerhalb Deutschlands, und die Kosten berechnen sich nach der Anzahl der bestellten Artikel.

Mit `count` nehmen Sie im Shipping-Bereich des Setup die Staffelung nach der Stückzahl vor. Hier sollen z. B. EUR 4,50 Versandkosten für einen bestellten Artikel berechnet werden, EUR 8,- für 2 und weitere Festbeträge für andere Stückzahlen (die Berechnung soll natürlich nur ein Beispiel sein und keine praktikable Preisgestaltung vorgeben!). Wie im folgenden Code zu sehen, ließe sich die Staffelung beliebig weiterführen.

Bei „Vielbestellern“ wollen Sie sich darüber hinaus mit einer versandkostenfreien Lieferung ab einem Gesamtwarenwert von EUR 500,- bedanken, was die Anweisung `price.noCostsAmount` in der letzten Zeile des Codebeispiels umsetzt.

Wählt der Kunde also den **Versand innerhalb Deutschlands**, der als erste Option (10) über `title` definiert wird, gilt (vorerst):

```
shipping {
  10.title = Versand innerhalb Deutschlands
  10.price = 4.50
  # exclude schließt Zahlungsmethoden aus: Wer z.B. "Versand"
  # auswählt, kann nicht bar zahlen.
  10.excludePayment = 20
  10.price.type = count
  10.price.1 = 4.50
  10.price.2 = 8.00
  10.price.3 = 12
  10.price.4 = 16
  10.price.5 = 20
  10.price.6 = 25
  10.price.10 = 30
  10.price.noCostsAmount = 500
}
```

Widmen Sie sich nun den sich daraus ergebenden Zahlungsmöglichkeiten, die im Payment-Bereich des Setup definiert werden. Bis auf die bei Versand unmögliche **Barzahlung** (`10.excludePayment = 20` – die Nummern beziehen sich auf die Payment-Definitionen im Setup) sollen dem Kunden alle Optionen zur Verfügung stehen, also **Rechnung** und **Nachnahme**.

Bei **Nachnahme** entstehen zusätzliche Kosten, die an den Kunden weitergegeben werden, und zwar über eine separate Definition im Payment-Bereich:¹

```
...
30.title = Nachnahme
30.price = 4.00
...
```

¹ Der Betrag ist wiederum nur ein Beispiel. Erkundigen Sie sich bei der Deutschen Post oder anderen Anbietern nach den tatsächlich für Sie entstehenden Kosten.

Der Betrag von EUR 4,- wird bei Klick des Kunden auf **Nachnahme** auf diese Weise automatisch ausgewiesen und in Rechnung gestellt; da es sich hier um eine von der Anzahl der bestellten Produkte unabhängige Gebühr handelt, ist keine weitere Berechnung oder Staffelung notwendig.

6.1.2 Versand ins Ausland

Aktuell bietet der Shop ausschließlich die Lieferung innerhalb Deutschlands. Dieses Angebot können Sie nun in einem ersten Schritt für Bestellungen ausländischer Kunden vorbereiten und die Länderauswahl nicht durch eine feste Zuordnung, sondern die flexible Abfrage aus einer umfangreichen Liste steuern.

Diese Liste stellt die Extension `static_info_tables` bereit; über das TypoScript-Setup `where.static_countries`, welches eine SQL-where-Bedingung definiert, erfolgt dann der Zugriff auf einzelne Länder(gruppen).²

Erweitern Sie darum zunächst lediglich Ihre Konfiguration, indem Sie die Versandmethoden (**Versand innerhalb Deutschlands** und **Selbstabholer**) an die entsprechende Länderauswahl binden:

```
shipping {
  ...
  10.title = Versand innerhalb Deutschlands
  10.where.static_countries = cn_short_local = 'Deutschland'
  10.price = 4.50
  10.excludePayment = 20
  ...
  20.title = Selbstabholer
  20.where.static_countries = cn_short_local = 'Deutschland'
  20.price =
  20.excludePayment = 10
  ...
}
```

Dieses Verfahren gilt es nun auszubauen, denn in einem Shop, dessen Angebot nicht auf Deutschland beschränkt ist, stoßen Sie bei der Vielzahl an Ländern und der noch größeren Vielzahl an Versandbedingungen mit statischen Zuweisungen rasch an die Grenzen des Überschaubaren.

Beginnen Sie mit der Unterscheidung folgender, zusätzlicher Versandgruppen, die mit Hilfe der `static_info_tables` möglichst flexibel gestaltet und im Rahmen einer „Mischkalkulation“ umgesetzt werden sollen:

² Das darf nicht in die Constants! In `sr_feuser_register` 2.5.8+ steht Ihnen dasselbe Setup zur Begrenzung der Länder in der Select Box zur Verfügung, `static_info_tables` 2.0.5+ ist hierfür erforderlich.

- EU-Ausland
- Nicht-EU-Ausland
- Schweiz

Widmen Sie sich zunächst den EU-Ländern außer Deutschland. Glücklicherweise bieten die `static_info_tables` bereits eine fertige Liste, die in einer neuen Versandoption zusammengefasst und für die zusätzlich eine neue Bezahlmethode definiert wird, nämlich **Vorkasse**.

Bei der Vorkasse fallen keine zusätzlichen Gebühren an (weder im In- noch im Ausland), so dass im Payment-Abschnitt zu ergänzen ist:

```
40.title = Vorkasse
```

Als neue Versandoption definieren Sie im Shipping-Bereich:

```
40.title = Versand innerhalb der EU ###STATICCOUNTRIES_CN_SHORT_EN###
40.where.static_countries = cn_eu_member = 1 AND cn_short_local !=
  'Deutschland'
40.price = 8.50
40.excludePayment = 10, 20, 30
```

Übersetzt bedeutet dies: Kunden aus allen EU-Ländern (`cn_eu_member`) außer Deutschland (ausgeschlossen über den Ausdruck `cn_short_local != 'Deutschland'`) können ausschließlich gegen Vorkasse (`excludePayment = 10, 20, 30`) einkaufen. Über `excludePayment` werden die Setup-Zeilen übergeben, die aus den Bezahlmethoden gestrichen werden. Zudem sind mit `price = 8.50` die Versandkosten pauschal auf EUR 8,50 festgelegt. Das muss nun dringend geändert werden, denn bei der Bestellung mehrerer Artikel erhöhen sich diese Kosten, so dass sich auch hier die Staffelung per `count` anbietet. Darum ergänzen Sie:

```
40.price.type = count
40.price.1 = 8.50
40.price.2 = 16
40.price.3 = 22
40.price.4 = 30
40.price.5 = 40
40.price.6 = 48
40.price.10 = 55
```

Erklärungsbedürftig ist der Marker `STATICCOUNTRIES_CN_SHORT_EN`. Dieser bewirkt, dass in der Combobox, die dem Kunden alle möglichen Versandoptionen anzeigt, die Ländernamen aus der angegebenen Gruppe (`cn_eu_member`) einzeln aufgelistet werden (Abbildung 6.1).

Abbildung 6.1:
Länderliste in der
Auswahlbox

| | | | | |
|---|-----------|--------------|---------|-------------------|
| Produkte: | Variante: | Einzelpreis: | Anzahl: | Gesamt in Euro: |
| Ein | Paperback | 29.90 | 1 | inkl. MWSt. 29.90 |
| Testprodukt: Paperback | | | | |
| Versand und Verpackung: | | | | |
| Pauschale: Versand nach Gewicht | | | | |
| Versand nach Gewicht | | | | inkl. MWSt. 2.50 |
| Selbstabholer | | | | |
| Nachnahme innerhalb Deutschlands | | | | WSt. 0.00 |
| Versand innerhalb der EU Austria | | | | |
| Versand innerhalb der EU Belgium | | | | |
| Versand innerhalb der EU Cyprus | | | | |
| Versand innerhalb der EU Czech Republic | | | | |
| Versand innerhalb der EU Denmark | | | | |
| Versand innerhalb der EU Estonia | | | | |
| Versand innerhalb der EU Spain | | | | WSt. 32.40 |
| Versand innerhalb der EU Finland | | | | (WSt. 30.44) |
| Versand innerhalb der EU France | | | | 1.96) |
| Versand innerhalb der EU United Kingdom | | | | |
| Versand innerhalb der EU Greece | | | | |
| Versand innerhalb der EU Hungary | | | | |
| Versand innerhalb der EU Ireland | | | | |
| Versand innerhalb der EU Italy | | | | |
| Versand innerhalb der EU Lithuania | | | | |
| Versand innerhalb der EU Luxembourg | | | | |
| Versand innerhalb der EU Latvia | | | | |

Kosmetische Variante: Wenn Sie die Ländernamen auf Englisch auflisten möchten, nicht aber die Option **Versand innerhalb der EU**, setzen Sie nur den Marker in Verbindung mit der SQL-Abfrage.

```
40.title = ###STATICCOUNTRIES_CN_SHORT_EN###
...
40.where.static_countries = cn_eu_member = 1 AND cn_short_local !=
'Deutschland'
```

Möchten Sie – z. B. aus Gründen der Übersicht – auf diese Liste komplett verzichten und **Versand innerhalb der EU** lediglich als *eine* Option anbieten, lassen Sie einfach den Marker weg:

```
40.title = Versand innerhalb der EU
```

Auch die gesamte Abfrage über `where.static_countries` ist in diesem Falle überflüssig, da keine Verwendung für die darüber ermittelten Daten besteht.

Kommen Sie nun zu einer weiteren, länderspezifischen Versandgruppe und räumen der Schweiz einen Sonderstatus ein, wobei Sie die dazu notwendigen Elemente bereits kennen:

```
50.title = Vorkasse (Schweiz)
50.where.static_countries = cn_short_de = 'Schweiz'
```

Im Shipping definieren Sie die Versandgebühren, schließen die nicht gewünschten Bezahlmethoden aus, erstellen ggf. wieder eine Preisstaffel mit count und erheben keine Versandkosten, wenn die Bestellung einen Wert von EUR 500,- erreicht.

```
50.title = Lieferung in die Schweiz
50.show = 1
50.price = 9.50
50.where.static_countries = cn_short_local = 'Schweiz'
50.excludePayment = 10, 20, 30, 40, 60
50.price.type = count
50.price.1 = 9.50
50.price.2 = 17
50.price.3 = 23
50.price.4 = 31
50.price.5 = 41
50.price.6 = 49
50.price.10 = 56
50.price.noCostsAmount = 500
```

show = 1 definiert, dass der Eintrag in der Auswahlliste erscheint. Im Gegensatz dazu stünde mit showLimit = 2 die Option nur dann zur Verfügung, wenn mindestens zwei Produkte im Warenkorb liegen. showLimit = 0 hat dieselbe Wirkung wie show = 1: Der Eintrag ist permanent, also unabhängig von der Anzahl der im Warenkorb befindlichen Produkte sichtbar.

Die Staffelpreise für den Versand bedeuten hier: 1 Stück für EUR 9,50, 2 Stück für EUR 17,-, ..., 6-9 Stück für EUR 49,-, 10 Stück und mehr für EUR 56,-. Ab EUR 500,- erfolgt die Lieferung versandkostenfrei. Die Art der Ware spielt hier keine Rolle, entscheidend ist die Stückzahl.

Im letzten Schritt für dieses Szenario bilden Sie eine neue Gruppe für die Lieferung ins nichteuropäische Ausland (Option **Lieferung weltweit (außer EU und Schweiz)**). Um Schreiarbeit und Fehlerquellen zu vermeiden, erstellen Sie die SQL-Abfrage so, dass alle Länder außer Deutschland, den übrigen EU-Ländern und der Schweiz zurückgegeben werden. Der Payment-Block lautet dann:

```
60.title = Vorkasse (weltweit außer EU und Schweiz)
60.where.static_countries = cn_short_de != 'Schweiz' AND
cn_eu_member != 1
```

Im Shipping-Bereich ergänzen Sie die Versandmethode und schließen auch hier die unerwünschten Bezahlmethoden aus. Die oben beschriebene SQL-Abfrage verwendet den logischen Ausdruck NOT IN:

```
60.title = Lieferung weltweit (außer EU und Schweiz)
###STATICCOUNTRIES_CN_SHORT_EN###
60.show = 1
```

```
60.price = 20.00
60.where.static_countries (
  cn_short_en NOT IN (
    'Switzerland') AND cn_eu_member != 1
)
60.excludePayment = 10, 20, 30, 40, 50
60.price.type = count
60.price.1 = 20
60.price.2 = 38
60.price.3 = 58
60.price.4 = 78
60.price.5 = 98
60.price.6 = 110
60.price.10 = 120
60.price.noCostsAmount = 500
```

Auch hier können Sie natürlich auf den Marker (und folglich auf das SQL-Statement) verzichten, um die Optionsliste übersichtlich zu halten; mit dem oben genannten Code werden konsequent die „übrig gebliebenen“ gut 150 Länder einzeln ausgegeben ...

Bei allen SQL-where-Abfragen müssen Sie auf die korrekte Klammerung achten. Wichtig: Öffnende runde Klammern dürfen *nicht* am Beginn einer Zeile stehen – das TypoScript der Seite gerät sonst durcheinander! Es darf auch keine schließende runde Klammer am Beginn einer Zeile stehen – es sei denn, sie definiert das Ende des gesamten Ausdrucks.

6.1.3 Ländergruppen definieren mit `static_info_tables`

Das Shopsystem hat nun bereits ein gewisses Maß an Flexibilität erreicht, indem es verschiedene Versandoptionen definiert und dazu passende Zahlungsoptionen anbietet sowie Versandkosten berechnet.

Bei einer Lieferung in EU-Länder haben wir aber beispielsweise die Rechnung ohne die Deutsche Post mit ihrem Paketdienst DHL gemacht; exemplarisch sollen dessen Konditionen im System implementiert werden.³

Um also die tatsächlichen Versandkosten von DHL an ausländische Kunden weiterzureichen, müssen Sie Ihr System erneut ändern und neue Ländergruppen entsprechend den vier DHL-Versandzonen definieren.

Angesichts der sehr zahlreichen Versandmöglichkeiten, die auch warenspezifisch sind, und der ebenso zahlreichen alternativen Paketdienste, kann das folgende System nur ein Beispiel sein. Für Ihren „echten“ Shop müssen Sie den oder die zu Ihren Kunden und Ihrem Warenangebot passenden Dienstleister selbst ermitteln und im Versand- und Bezahlssystem berücksichtigen. Oder Sie arbeiten mit Pauschalbeträgen.

³ http://www.dhl.de/dhl?xmlFile=3001057&lang=de_DE

DHL z. B. unterscheidet vier sog. „Zonen“:

Länder der Zone 1 (EU)

Belgien, Bulgarien, Dänemark (außer Färöer, Grönland), Estland, Finnland (außer Ålandinseln), Frankreich (außer überseeische Gebiete und Departements), Griechenland (außer Berg Athos), Großbritannien (außer Kanalinseln), Irland, Italien (außer Livigno und Campione d'Italia), Lettland, Litauen, Luxemburg, Malta, Niederlande (außer außereuropäische Gebiete), Österreich, Polen, Portugal, Rumänien, Schweden, Slowakei, Slowenien, Spanien (außer Kanarische Inseln, Ceuta + Melilla), Tschechische Republik, Ungarn, Zypern/Republik (außer Nordteil)

Erstellen Sie also die Bezahl- und Versandmethoden zunächst für diese Zone und nutzen dazu die dreistelligen (A3) Ländercodes nach ISO 3166. Sie finden die Ländertabelle im Webmodul Liste per Klick auf den Namen neben dem Weltkugelsymbol. Zur oben aufgeführten Liste von DHL gibt es allerdings ein paar Unterschiede, z. B. Färöer und Grönland, die zwar zu Dänemark gehören, aber beim Versand einer anderen Zone zugeordnet werden müssen. Bei Griechenland gibt es gar ein Problem, denn der Berg Athos ist in den `static_info_tables` nicht eigens aufgeführt. Wenn Sie also dorthin liefern möchten, erstellen Sie eine eigene Bezahl- und Liefermethode, wie Sie es beispielsweise für die Schweiz getan haben.

Ihre Länderliste für Zone 1 sähe demnach im Payment wie folgt aus:

```
#           Ländercodes nach ISO 3166 A3
40.where.static_countries (
cn_iso_3 IN (
'BEL',
'BGR',
'DNK',
'EST',
'FIN',
'FRA',
'GRC',
'GBR',
'IRL',
'ITA',
'LVA',
'LTU',
'LUX',
'MLT',
'NLD',
'AUT',
'POL',
'PRT',
'ROU',
'SWE',
```

```
'SVK',
'SVN',
'ESP',
'CZE',
'CYP',
'HUN' )
)
```

Beachten Sie: Die erste runde Klammer ist Teil von TypoScript. Damit wird der folgende Code als eine zusammengehörende Anweisung definiert. Die zweite runde Klammer ist Teil der SQL-where-Bedingung. Sie wird 1:1 zur Generierung der SQL-Abfrage übernommen. Das bedeutet, dass der Teil, der innerhalb der ersten Klammern steht, auch SQL-syntaktisch korrekt sein muss.

Alternativ bieten sich auch die englischen Kurzschreibweisen an:

```
#           Ländercode nach Short_en
40.where.static_countries (
  cn_short_en IN (
    'Belgium',
    'Bulgaria',
    'Denmark',
    'Estonia',
    'Finland',
    'France',
    'Greece',
    'United Kingdom',
    'Ireland',
    'Italy',
    'Latvia',
    'Lithuania',
    'Luxembourg',
    'Malta',
    'Netherlands',
    'Austria',
    'Poland',
    'Portugal',
    'Romania',
    'Sweden',
    'Slovakia',
    'Slovenia',
    'Spain',
    'Czech Republic',
    'Cyprus',
    'Hungary' )
)
```

In gleicher Weise verfahren Sie für die Zonen 2 („Europa ohne EU“) und 3 („Welt“), wie im Codebeispiel zu sehen.

Länder der Zone 2 (Europa ohne EU)

Ålandinseln (Finnland), Andorra, Albanien, Armenien, Aserbaidschan, Berg Athos (GR), Bosnien-Herzegowina, Campione d'Italia (IT), Ceuta (E), Färöer-Inseln (DK), Georgien, Gibraltar (GB), Grönland (DK), Guernsey (GB), Island, Jersey (GB), Kanarische Inseln (E), Kasachstan, Kosovo (serbische Provinz), Kroatien, Liechtenstein, Lignino (IT), Mazedonien, Melilla (E), Moldau (Republik), Montenegro (Republik), Norwegen, Russische Föderation, San Marino, Schweiz, Serbien (Republik), Türkei, Ukraine, Vatikanstadt, Weißrussland, Zypern/Republik (Nordteil)

Entsprechend lautet der Code (Auszug):

```
50.where.static_countries ( cn_iso_3 IN (
'ALA' ,
'AND' ,
'FRO' ,
'GBR' )
...
)
```

Länder der Zone 3 (Welt)

Ägypten, Algerien, Bahrain, Iran, Irak, Israel, Jemen, Jordanien, Kanada, Katar, Kuwait, Libanon, Libyen, Marokko, Oman, Saudi-Arabien, Syrien Tunesien, USA, Vereinigte Arabische Emirate

```
60.where.static_countries = cn_iso_3 IN (
'EGY' ,
'DZA' ,
'BHR' )
...
)
```

Neben der „Welt“ kennt DHL auch noch eine „Rest-Welt“:

Zone 4 (Rest-Welt)

Alle Länder und Gebiete, die nicht den Zonen 1, 2 oder 3 zugeordnet sind.

Um hier nicht alle der in den Zonen 1, 2 und 3 nicht erfassten Länder mühsam herausuchen zu müssen, kopieren Sie die Listen dieser Zonen hierher und schließen sie dann mit dem bereits bekannten Ausdruck NOT IN von der Anzeige aus:

```
40.where.static_countries = cn_iso_3 NOT IN (
'AND' ,
'BEL' ,
'BGR' )
)
```

Wie Sie vielleicht bemerkt haben, werden in den Codebeispielen unterschiedliche Felder aus den `static_info_tables` angesprochen: Einmal der ISO-Code (`cn_iso_3`), einmal die englische (`cn_short_en`) und dann wieder die deutsche (`cn_short_local`) Kurzschreibweise. In der Tabelle, die Sie über das Webmodul Liste oder über phpMyAdmin erreichen, finden Sie die Feldnamen, die Sie nutzen können. Eventuell müssen Sie zunächst die Anzeige der Felder über den Befehl Felder setzen aktivieren.

Abbildung 6.2:
Felder der Tabelle
`static_countries`

| | Field | Type | Collation | Attributes | Null | Default |
|--------------------------|------------------------|--------------|-------------------|------------|------|---------|
| <input type="checkbox"/> | uid | int(11) | | UNSIGNED | No | auto |
| <input type="checkbox"/> | pid | int(11) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_iso_2 | char(2) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_iso_3 | char(3) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_iso_nr | int(11) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_parent_tr_iso_nr | int(11) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_official_name_local | varchar(128) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_official_name_en | varchar(128) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_capital | varchar(45) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_tldomain | char(2) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_currency_iso_3 | char(3) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_currency_iso_nr | int(10) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_phone | int(10) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_eu_member | tinyint(3) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_address_format | tinyint(3) | | UNSIGNED | Yes | 0 |
| <input type="checkbox"/> | cn_zone_flag | tinyint(4) | | | Yes | 0 |
| <input type="checkbox"/> | cn_short_local | varchar(70) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_short_en | varchar(50) | latin1_swedish_ci | | Yes | |
| <input type="checkbox"/> | cn_uno_member | tinyint(3) | | UNSIGNED | Yes | 0 |

6.2 Versandkosten nach Gewicht

Eine weitere, noch flexiblere Möglichkeit zur Berechnung der Versandkosten ist die Ermittlung des tatsächlichen Portos über das Gewicht. Voraussetzung dafür ist das Hinterlegen des Produktgewichts im System. Sofern Sie mit Varianten arbeiten (vgl. Abschnitt 4.2 ab Seite 79), muss das Gewicht beim Artikel eingetragen werden, nicht beim Elternprodukt – dies ist eine sehr häufige Fehlerquelle. Die Angabe des Gewichts im Produkt bzw. im Artikel selbst erfolgt in Kilogramm. Im folgenden Beispiel beträgt es der Einfachheit halber 1 kg. Im hier zu bearbeitenden TS-Setup sind hingegen Angaben in Gramm notwendig.

Im Codebeispiel ist testweise die Versandmethode **Versand nach Gewicht** definiert. Alle anderen Bezahlmethoden sind hier zunächst einmal ausgenommen.

```
10.title = Versand nach Gewicht
10.price.type = weight
10.price.1000 = 4
10.price.5000 = 20
10.price.10000 = 40
10.price.20000 = 80
10.price.30000 = 120
10.excludePayment = 20, 30, 40, 50, 60
```

Auch diese Methode ließe sich natürlich für einzelne Länder oder Ländergruppen, wie oben beschrieben, definieren. Das aufgeführte Codebeispiel weist aus, dass bei einem Versandgewicht ab 1.000 g Versandkosten in Höhe von EUR 4,-, ab 5 kg EUR 20,- usw. erhoben werden. Wenn Sie allerdings `10.price.noCostsAmount = 500` setzen, werden die ermittelten Versandkosten zwischen EUR 4,- und 120,- ignoriert, wenn der Warenwert über EUR 500,- liegt. Im Beispielshop tritt das bei einem Gewicht ab 20 kg ein.

Die Preisstaffelung nach festen Gewichtsklassen wird in zwei Schritten dem flexiblen Berechnungssystem nach DHL-Art anpasst. Ausgehend von einer `tt_products`-weiten Maßeinheit (`FactWeight` genannt) – in diesem Falle Kilogramm – weisen Sie zunächst einer Einheit bestimmte Versandkosten zu. Aus dem bei den Produkten jeweils hinterlegten Gewichten kann das System dann den Preis berechnen.

```
10.title = Versand nach Gewicht
10.priceFactWeight = 4
10.price.noCostsAmount = 500
```

Mit diesem Code kostet der Versand pro Kilogramm also EUR 4,-. Um nun das DHL-System vollständig zu implementieren, müssen Sie zu diesem gewichtsabhängigen noch einen stückzahlabhängigen Versandkostenanteil ergänzen.

6.3 Versandkosten nach Gewicht und Stückzahl

Hier – wie bei vielen Paketdiensten – setzen sich die Versandkosten aus einer festen Gebühr von EUR 0,50 pro bestelltem Artikel *plus* einem oben beschriebenen gewichtsabhängigen Betrag zusammen.

Dazu ergänzen wir `priceFactWeight` um eine Zähler-Konfiguration namens `priceFactCount`:

```
10.title = Versand nach Gewicht
10.priceFactWeight = 4
10.price.type = count
10.priceFactCount = 0.50
10.excludePayment = 20, 30, 40, 50, 60
10.price.noCostsAmount = 500
```

Auch hier gibt es wie oben keine Preisstaffel. Damit aber `noCostsAmount` verwendet werden kann, muss ein `price-Setup` definiert werden. Ab einem Einkaufswert von EUR 500,- werden keine Versandkosten berechnet; die über `10.priceFactWeight` und `10.priceFactCount = 0.50` ermittelten Versandkosten werden in diesem Falle nicht erhoben und im Warenkorb mit EUR 0,- ausgewiesen. Unterhalb dieses Bestellwerts werden die Versandkosten in Rechnung gestellt.

6.4 Preisnachlässe

Auch Rabatte sind beim Online-Kauf möglich; für die Umsetzung gibt es verschiedene Wege, die wir uns nun anschauen werden. Eine Randbemerkung: In diesem Beispielshop, über den hauptsächlich Bücher verkauft werden, wäre das aus Gründen Buchpreisbindung bei der Produktgruppe Bücher rechtlich nicht zulässig. Bei Büchern könnten Sie allerdings ab einem bestimmten Preis ohne Lieferkosten versenden.

6.4.1 Benutzer

Beim Editieren der Benutzerdatensätze im Seiten- oder Listen-Modul im TYPO3-Backend können Sie einen Rabatt je Benutzer festlegen, indem Sie ein Häkchen im Feld **Rabatt** setzen und den gewünschten Prozentsatz eingeben. Sobald der Benutzer angemeldet ist, erhält er auf jedes Produkt und jeden Artikel diesen Rabatt.

Wenn Sie die Frontendbenutzerregistrierung nutzen (vgl. Abschnitt 3.2 ab Seite 54), setzen Sie das Häkchen im Feld **Rabatt** der Frontendbenutzerregistrierung und tragen einen Prozentwert ein.

6.4.2 Skonto

Wenn Sie den folgenden Code im Payment bei Barzahlung einfügen, erhalten die Kunden 3% Skonto.

```
plugin.tt_products.payment {
    radio=0
    TAXpercentage = 7
```

```

10.title = Rechnung
10.where.static_countries = cn_short_local = 'Deutschland'
20.title = Barzahlung
## Skonto bei Barzahlung
20.percentOfGoodstotal = -3
20.where.static_countries = cn_short_local = 'Deutschland'
...
}

```

In Version 2.6.0 von `tt_products` ist es noch nicht möglich die erhaltenen Prozente zusätzlich durch Text und Marker auszugeben. Der Preisnachlass wird jedoch korrekt berechnet.

6.4.3 Bestellmenge und Produktpreis

Rabatte können sich darüber hinaus auch nach der Bestellmenge richten, indem z. B. der Stückpreis im Rahmen einer Mengenschaffelung (`count`) sinkt.

```

plugin.tt_products.getDiscountPrice = 1
## Rabatte
plugin.tt_products.discountprice {

    10.type = count
    10.field = price
    10.additive = 1
    10.prod.1 = 39.90
    10.prod.10 = 38.90
    10.prod.100 = 35.90

    20.type = count
    20.field = price
    20.additive = 1
    20.prod.1 = 19
    20.prod.10 = 18
    20.prod.100 = 15
}

```

Zunächst muss die Rabattdfunktion durch `getDiscountPrice = 1` aktiviert werden. Damit werden die Rabatte über `discountprice` für alle Käufer aktiviert. Auch eine Aktivierung nur für angemeldete Benutzer einer bestimmten Benutzergruppe mit `discountGroupName` ist möglich und wird im weiteren Verlauf beschrieben. Anders wäre die Umsetzung produktbezogener Rabatte über `discountprice` nicht möglich.

In diesem Beispiel richten sich die Nachlässe nach dem Produktpreis und der bestellten Stückzahl. Im ersten Fall (10) greift die Staffel für Artikel mit einem Preis von EUR 39,90; für alle Artikel dieses Preises gilt dann, dass bei

10 bestellten Exemplaren der Stückpreis auf EUR 38,90 sinkt, bei 100 auf EUR 35,90.

Der zweite Block (20) definiert eine solche Reihe für Produkte mit einem regulären Preis von EUR 19,-.

`additive = 1` sorgt dafür, dass zur Bestimmung der Rabattgrenze *alle* Produkte im Warenkorb herangezogen werden. Ist `additive` nicht gesetzt, werden nur jeweils die Produkte einer Preisgruppe (10, 20, ...) addiert und ggf. rabattiert. Beispiel: Es liegen 10 (oder mehr) Produkte mit unterschiedlichen Stückpreisen im Warenkorb; ist `additive = 1` gesetzt, wird auf jeden Artikel der in dessen Staffelung eingeräumte Rabatt gewährt; ist `additive = 0`, wird kein Rabatt gewährt, da sich nicht 10 Artikel derselben Preisgruppe im Warenkorb befinden.

Wünschenswert, aber noch nicht implementiert, ist ein *gruppenspezifischer* Preisnachlass. Finden die Entwickler einmal Muße oder einen Sponsor für dieses Feature, ähnelt der Code vermutlich dem folgenden.

Mehrere Gruppen würden durch Komma separiert. `plugin.tt_products.getDiscountPrice = 1` sollte in diesem Fall nicht gesetzt werden, weil es eine Gruppenbeschränkung überschreiben würde.

```
plugin.tt_products.discountGroupName = Discountgruppe1, Discountgruppe2
  ## Rabatte
  plugin.tt_products.discountprice {
    10.type = count
    10.field = price
    10.additive = 1

    10.prod.1 = 39.90
    10.prod.10 = 38.90
    10.prod.100 = 35.90

    20.type = count
    20.field = price
    20.additive = 1

    20.prod.1 = 29.90
    20.prod.10 = 28.90
    20.prod.100 = 25.90
  }
```

Aber, wie gesagt, diese Möglichkeit steht zum Zeitpunkt der Erstellung dieses Buches (noch) nicht zur Verfügung.

Sie können das über TypoScript-Bedingungen mit Angabe der Gruppen-IDs umgehen:

```
plugin.tt_products.getDiscountPrice = 0
[usergroup = 1, 2]
```

```

plugin.tt_products.getDiscountPrice = 1
[global]

## Rabatte
plugin.tt_products.discountprice {
    10.type = count
    10.field = price
    10.additive = 1
}

```

6.4.4 Händler-Rabatte mit Preis2

Verschiedene Arten, Sonderkonditionen zu gewähren, sind damit durchgespielt. Eine weitere soll nun im Zusammenhang mit besonderen Einkaufsbedingungen für Händler erläutert werden, dem sog. „Preis2“; im Kapitel 4 wurde ab Seite 71 bereits auf dieses Feld eingegangen. Es handelt sich weder um eine eigene Bezahlmethode noch um eine im oben angesprochenen Sinne gruppenspezifische Rabattregelung. Es handelt sich vielmehr um den Kauf zu einem anderen, fest definierten Preis. Dazu reichen die Informationen aus Kapitel 4 im Grunde aus, allerdings können Sie auch diesen „Preis2“ wiederum staffeln.

Folgendes Szenario: Kleineren Händlern Ihrer Produkte räumen Sie bei der Abnahme einer höheren Stückzahl auch einen höheren Rabatt ein; eben dies geschieht durch Staffelung von „Preis2“.

Definieren Sie zunächst eine Benutzergruppe aus der Benutzerregistrierung, der Ihre Fachhändler (Reseller) angehören. Setzen Sie folgenden Code ins Setup:

```

## Händlerpreis für Usergroup 11 = Händler

[usergroup=11]
priceNoReseller = 2
[global]

```

Damit erhält die Benutzergruppe mit der UID 11 den reduzierten, in Preis2 eingetragenen Preis. Um diesen nach Anzahl der bestellten Produkte staffeln zu können, verwenden Sie wieder count.

```

plugin.tt_products.discountprice {
    10.type = count
    10.field = price2
    10.additive = 1
    10.prod.1 = 34.90
    10.prod.10 = 33.90
    10.prod.100 = 30.90
}

```

Bei all diesen Möglichkeiten sollten Sie darauf achten, den Überblick zu behalten. Denn die Arbeit mit verschiedenen Rabattarten und -gruppen kann natürlich dazu führen, dass sich die gewährten Preisnachlässe bei Überschneidung der jeweiligen Kriterien addieren und Kunden plötzlich mehrere (und vielleicht ein wenig zu üppige) Rabatte erhalten.

6.5 Mindestbestellwert

Bei Shops, in denen Artikel mit sehr geringem Warenwert angeboten werden, empfiehlt sich unter Umständen die Festlegung eines Mindestbestellwerts, um einen unverhältnismäßig hohen Bearbeitungsaufwand zu vermeiden.

Soll eine Bestellung nur zustande kommen, wenn der Gesamtbestellwert EUR 100,- oder mehr umfasst, ergänzen Sie im TS-Setup folgenden Code:

```
basket.minPrice{
  type=price
  collect = goodstotal
  value = 100
}
```

Im HTML-Template müssen Sie nun den Subpart definieren bzw. editieren:

```
<!--###MESSAGE_MINPRICE_ERROR###-->

<p Sie haben den Mindestbestellwert von EUR 100.00 nicht erreicht.
Die Bestellung kann nicht ausgeführt werden.</p>

<!--###MESSAGE_MINPRICE_ERROR###-->
```

Dieser Subpart muss in einem Template-Subpart eingefügt werden, der zu einem Warenkorb-Plugin gehört, z. B. BASKET_TEMPLATE.

```
<div id="SUBMIT_BASKET5">
  <input type="hidden" name="mode_update" value="1">
  <input type="submit" id="button3" name="products_update"
  value="Warenkorb aktualisieren">
  <input type="submit" id="button3" name="products_clear_basket"
  value="Warenkorb löschen">

<!-- ###MESSAGE_MINPRICE### begin -->
  <input type="submit" id="button3" name="products_info"
  value="Daten eingeben"
  onclick="this.form.action='###FORM_URL_INFO###';">
  <input type="hidden" name="creditpoints_saved"
  value="###CREDITPOINTS_SAVED###">
```

```



```

Der Subpart MESSAGE_MINPRICE wird nur angezeigt, wenn der Mindestbestellwert erreicht ist. Wenn Sie, wie im Codebeispiel, den Button **Warenkorb aktualisieren** immer angezeigt lassen, kann der Benutzer bei einem zu geringen Bestellwert die Anzahl der bestellten Produkte noch erhöhen.

Darüber hinaus muss auf oberster Ebene ein neuer Subpart eingefügt werden; er darf also nicht in einen anderen Subpart verschachtelt werden.

```

<!-- ###BASKET_TEMPLATE_MINPRICE_ERROR### begin
If the basket is empty this subpart is displayed
-->
<br>
###GW1B###<B>Es müssen mindestens Waren für 100 Euro
gekauft werden.</B>###GW1E###
<!-- ###BASKET_TEMPLATE_MINPRICE_ERROR### end -->

```

Das wird benötigt, damit ein Kunde, der für weniger als EUR 100,- einkauft, keine Bestellung durchführen kann. Sollte er dies dennoch versuchen, erscheint der oben angegebene Text. Dieser Subpart erscheint auf allen Seiten im Anschluss an den Warenkorb. Auf diese hätte der Käufer im Shop allerdings nie kommen dürfen, solange im Warenkorb nicht ausreichend Waren liegen. Dagegen wird der Subpart-Marker MESSAGE_MINPRICE_ERROR innerhalb des Warenkorbs verwendet. Auch wenn jemand zu wenige Produkte in den Warenkorb gelegt hat, muss es ihm möglich sein, einen Blick in den Warenkorb zu werfen, um daraufhin gegebenenfalls weiter einzukaufen.

6.6 Kreditkartenzahlung

Die Zahlung mit Kreditkarte in Onlineshops ist nicht unproblematisch. Der Server muss SSH-verschlüsselt und sollte auch zertifiziert sein. Darüber hinaus unterliegen Nutzung, Speicherung und Weitergabe von Kreditkartendaten strengen gesetzlichen Regelungen, die für Betreiber kleiner und mittlerer Shops nicht ohne Weiteres zu erfüllen sind. Die Alternative besteht darin, den eigentlichen Datentransfer – und damit den eigentlichen Zahlungsvorgang – auszulagern und über ein sog. *Payment-Gateway* abzuwickeln; es handelt sich dabei in der Regel um fertige Software-Lösungen, die Sie „irgendwie“ in Ihren TYPO3-Shop integrieren müssen. Dieses „Irrendwie“ ist Thema der folgenden Abschnitte, denn obwohl TYPO3 als CMS

weit verbreitet ist, fehlte es bislang an einer Standardimplementierung dieses Verfahrens und natürlich auch an dessen Dokumentation.

Es geht hier also um Kreditkartenzahlung, bei der die sensiblen Daten *nicht* auf dem System des Shopbetreibers, sondern über einen Link zu einem externen Bezahlssystem auf diesem eingegeben und verarbeitet werden. Nach erfolgter Zahlung kehrt der Kunde via Redirect zum Shop-Server zurück und schließt dort die Bestellung ab.

Bereiten Sie zunächst im Setup die Zahlungsabwicklung über Kreditkarte vor; folgender Beispielcode soll dabei helfen. Vergessen Sie nicht, das Original vorher zu sichern. Im Weiteren wird es dann um die konkrete Einbindung zweier externer Systeme (PayPal und ipayment) gehen.

```
plugin.tt_products.payment >
plugin.tt_products.payment {
    #radio=0
    TAXpercentage = 7
    10.title = Rechnung
    10.where.static_countries = cn_short_local = 'Deutschland'
    20.title = Barzahlung
    20.percentOfGoodstotal = -10
    20.where.static_countries = cn_short_local = 'Deutschland'
    30.title = Nachnahme
    30.where.static_countries = cn_short_local = 'Deutschland'
    30.price = 4.00
    40.title = Vorkasse EU
    40.where.static_countries = cn_eu_member = 1 AND cn_short_en !=
        'Germany'
    50.title = Vorkasse Schweiz
    50.where.static_countries = cn_short_de = 'Schweiz'
    60.title = Vorkasse außereurop. Ausland
    60.where.static_countries = cn_short_de != 'Schweiz' AND cn_short_en !=
        'Germany' AND cn_eu_member != 1
    70.title = PayPal
    80.title = IPayment
}
```

6.6.1 PayPal über Formular

Die Kreditkartenzahlung via PayPal setzt voraus, dass sowohl der Shopbetreiber als auch der Kunde über ein PayPal-Konto verfügen. Um Ihr Formular ausgiebig testen zu können, sollten Sie sich auf der Seite des Anbieters zwei solche Testkonten mit Zugangs- und fiktiven Bankdaten in einer sog. „Sandbox“ einrichten.⁴ Dafür fallen keine Kosten an. Wichtig ist, dass Sie mit dem gleichen Browser sowohl auf die Sandbox-Seite als auch auf Ihrer Webseite arbeiten.

Im HTML-Template von `tt_products` finden Sie im Subpart `BASKET_ORDER_CONFIRMATION` folgendes Formular:

```
<form target="paypal" name="paypal"
  action="https://www.sandbox.paypal.com/de/webscr" method="post">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="business" value="paypal@mytypo3domain.de">
<input type="hidden" name="item_name" value="###ORDER_UID###">
<input type="hidden" name="amount" value="###PRICE_GOODSTOTAL_TAX###">
<input type="hidden" name="no_note" value="1">
<input type="hidden" name="currency_code" value="EUR">
<input type="hidden" name="lc" value="DE">
<input type="hidden" name="tax" value="###PRICE_PAYMENT_TAX###">
<input type="hidden" name="shipping" value="###PRICE_SHIPPING_TAX###">
<input type="hidden" name="cancel_return"
  value="http://mytypo3domain.de/index.php?id=329" />
<input type="hidden" name="return"
  value="http://mytypo3domain.de/index.php?id=320" />
<input type="submit" name="submit" id="button2"
  value="Bezahlung mit Kreditkarte via PayPal"
  alt="Bezahlen mit Kreditkarte via PayPal">
<input type="hidden" name="cancel_return"
  value="http://www.mytypo3domain.de/zahlungsstornierung.html"
<input type="hidden" name="return"
  value="http://www.mytypo3domain.de/danke.html"
</form>
```

Es empfiehlt sich, als `name` für das Formular `paypal` zuzuweisen, um bei weiteren Bezahlmethoden über Formulare die Übersicht zu behalten.

Es müssen auf jeden Fall die vier verborgenen Variablen `amount`, `business`, `item_name` und `submit` bzw. `image` gesetzt sein. Mit `action` geben Sie den Link zu PayPal an, für die Tests verwenden Sie am besten die Sandbox-URL, mit `value = "_xclick"` wird `cmd` auf `_cart`, also den Warenkorb gesetzt. `business` nimmt die Adresse Ihres PayPal-Kontos auf. Die Variable `item_name` enthält die von Ihrem Shopsystem übergebene Bestellnummer (`ORDER_UID`). Mit `no_note = 1` verzichten Sie darauf, dem Kunden ein Kommentarfeld anzubieten; eine 0 würde diese Möglichkeit geben. Soll die

⁴ <https://www.developer.paypal.com>

Währung Euro sein, so wird `currency_code` entsprechend gesetzt. Über `lc` definieren Sie den Ländercode.

Die Variable `amount` weist den Preis aus, was einfacher klingt, als es ist: `PRICE_TOTAL_TAX` entspricht dem Gesamtpreis inklusive Steuer, Porto und Versand, also etwa dem „Rechnungsendbetrag“.

`PRICE_GOODSTOTAL_NO_TAX` entspricht dem Gesamtnettowarenpreis ohne Porto und Versand.

Ist `amount = ###PRICE_TOTAL_TAX###` gesetzt, dürfen demnach `tax` und `shipping` nicht gesetzt sein, da in diesem Fall Steuer und Porto doppelt berechnet würden.

Mit `cancel_return` verweisen Sie auf eine Seite Ihres Webauftritts, auf die der Kunde zurückgeleitet wird, wenn der Zahlungsvorgang abgebrochen wurde. Dazu erstellen Sie eine Seite mit entsprechendem Text und weisen die – im Menü versteckte – Seite über die URL zu. Entsprechend gehen Sie für den Redirect zur Seite im Falle einer erfolgten Zahlung vor.

Für den Submit können Sie einen Button oder ein Bild nutzen. Setzen Sie das Gutscheinsystem (Abschnitt 5.1.8 ab Seite 103) ein, ist der Marker `PRICE_TOTAL_TAX` durch `PRICEVOUCHER_TOTAL_TAX` zu ersetzen.⁵

Bei Zahlung über PayPal ist darauf zu achten, dass alle Preise mit Dezimalpunkt angegeben werden, da PayPal das in Deutschland übliche Komma zur Trennung von Euro und Cent nicht akzeptiert. Möglicherweise ist bei Erscheinen des Buches bereits implementiert, dass der Punkt in ein Komma umgewandelt wird und damit die Preise in gewohnter Weise ausgewiesen werden.

Die entsprechenden Marker (`MESSAGE_PAYMENT`) für die Zahlungsmethode PayPal (in unserem Beispiel bei der Formularvariante 70) finden Sie im HTML-Template. Wenn in Ihrem Shop die Zahlung via PayPal eine andere Nummerierung aufweist, ändern Sie dies bitte entsprechend.

6.6.2 PayPal über Extension

Neben der Möglichkeit, die Abrechnung via PayPal über ein Formular zu implementieren, existiert eine eigene TYPO3-Extension namens „PayPal-Suite“. Deren Installation ist ab Seite 153 beschrieben. Bearbeiten Sie zunächst die HTML-Template-Dateien, wie ab Seite 149 erläutert.

Die Abwicklung über die Extension bietet die Option, dass die Zahlung erfolgt sein muss, *bevor* das System eine Bestellung/Order erstellt. Das verhindert „leere“ Aufträge und erleichtert die Buchhaltung, da eben nur zustande gekommenen Transaktionen eine Warenbestellung zugeordnet wird.

⁵ Weitere Informationen zur Variablenreferenz finden Sie unter https://www.paypal.com/IntegrationCenter/ic_std-variable-reference.html.

Dieses Procedere hat allerdings auch einen gravierenden Nachteil: Wenn irgendetwas nicht funktioniert und der Kunde die Zahlung nicht ausführen konnte, besitzen weder der Kunde noch der Shop-Betreiber irgendwelche Unterlagen zu dem Vorgang. Der Kunde nimmt möglicherweise an, dass er bezahlt hat, und wartet auf die Lieferung, die der Shop-Betreiber mangels Auftrag niemals auf den Weg bringt ... Hier muss man sich also Gedanken machen, an welcher Stelle im Bestellvorgang die Zahlung erfolgen soll.

Die PayPal-Extension kann übrigens den Warenkorb der über PayPal abgewickelten Produkte übermitteln, so dass Sie auch in Ihrem PayPal-Account sehen, welche Waren gekauft wurden.

Installation der PayPal-Suite

Prüfen Sie zunächst, dass die Extension `sr_static_info` *nicht* installiert ist.⁶

Anschließend sind folgende Extensions zu installieren:

- `lib 0.0.20+`
- `div 0.0.9+`
- `static_info_tables 2.0.5+`
- `paymentlib 0.2.1+`
- `paymentlib_paypal` (PayPal Payment Suite)
- `paysuite` (Payment Suite Extension)

Das Pluszeichen hinter der Versionsnummer bedeutet, dass eine Extension in dieser Version oder einer höheren installiert werden muss; die hier genannte Reihenfolge ist bei der Installation einzuhalten: Zunächst muss die `paymentlib_paypal`, *danach* die `paysuite` installiert werden. Legen Sie in Ihrem Shop einen SysOrdner an, in dem das System die Transaktionen abspeichern wird, und tragen Sie dessen Seiten-ID im Extension Manager zur **Payment code library Extension** ein.

```
PID [pid]
Page ID of a sysfolder where transaction records are saved
(Integer)
Default: 0
```

⁶ `sr_static_info` von Stanislas Rolland war als Erweiterung der `static_info_tables` konzipiert und wurde wieder in diese re-integriert. `sr_static_info` ist damit überflüssig geworden und verursacht zudem Fehler, wenn sie gemeinsam mit der anderen Extension verwendet wird. In den `static_info_tables` ist der vollständige Code ab Version 2.0.0 enthalten. Einige Extensions wie `tt_products` prüfen aber noch, ob sie installiert ist.

In `paymentlib_paypal` müssen Sie folgende Einstellungen vornehmen: **Your business e-mail** entspricht der E-Mail-Adresse Ihres PayPal-(Test-)Accounts.

Anschließend geben Sie die Test-E-Mail des Shopbetreibers an. Beachten Sie, dass diese Test-E-Mails nur funktionieren, wenn Sie gleichzeitig mit demselben Browser in der Sandbox mit Ihrem Test-Account eingeloggt sind. Sie können diese Test-E-Mails also nicht an Dritte weitergeben. Die Zugangsdaten (Passwort), die Sie beim Anlegen eingegeben haben, benötigen Sie für den Zugang zu dem PayPal-Sandbox-Konto des Shopbetreibers oder des Käufers.

Als PayPal-URL belassen Sie die Voreinstellung `https://www.Paypal.com/cgi-bin/webscr` – arbeiten Sie mit der Sandbox, setzen Sie `https://www.sandbox.Paypal.com/cgi-bin/webscr`, wie bereits im Beispiel über das Formular beschrieben ist.

Ein Beispiel-Setup hat dann etwa die folgende Form:

```
plugin.tt_products {
  paymentActivity = payment
  ...
  payment >
  payment {
    radio = 1
    TAXpercentage = 16
    ...
    90.title = Paypal - vor Abschluss der Bestellung
    90.image.file = EXT:paymentlib_paypal/res/logo_saferpay_color_low.png
    90.handleLib = paymentlib
    90.handleLib {
      extName = paymentlib_paypal
      extTitle = Paypal Gateway
      extInfo = PayPal enables any business or consumer with an email
        address to securely, conveniently, and cost-effectively send and
        receive payments online.
      extImage = typo3conf/ext/paymentlib_paypal/res/paypal_euro.gif
      gatewaymode = form
      paymentMethod = paypal_webpayment_euro
      Currency = EUR
      templateFile = EXT:tt_products/template/paymentlib.tpl
    }
  }
  ...
}
```

Weitere Zahlungsmethoden (`paymentMethod`) können Sie der Datei `paymentmethods.xml` der Extension `paymentlib_paypal` entnehmen.

Für ein individuelles Design kopieren Sie die Datei `tt_products/template/paymentlib.tpl` in ein Verzeichnis unterhalb `fileadmin` und än-

dern den Pfad zum Template im Setup entsprechend ab. Das Setup müssen Sie allerdings auch an die Namen Ihres Verzeichnisses und der PayPal-Template-Datei anpassen:

```
templateFile = fileadmin/templates/paymentlib.tpl
```

Die Umsetzung mit PayPal-Suite und Payment Library hat einige Vorteile gegenüber dem Zugang über das HTML-Formular:

- Bezahlung vor dem Abschluss der Bestellung
- einfache Konfiguration im Backend: Setup und Extension Manager
- genaues Auflisten der gekauften Produkte aus dem Warenkorb auch im PayPal-Konto möglich
- Mitprotokollieren über Tabellen der Payment Library
- Rückkehr in den Warenkorb bei Abbruch einer Bestellung und Fortsetzen von dort
- keine Kenntnisse über die PayPal-internen Formularfelder notwendig
- keine Konfiguration im HTML-Quelltext

Wenn Sie die Payment-Methode „finalize“ (vgl. Seite 157) verwenden, erreichen Sie dasselbe Ergebnis wie bei einer direkten Konfiguration im Formular.

Template-Anpassungen für den Gateway-Zugang

Die folgenden Template-Anpassungen sind ein allgemeines Beispiel, wie Sie auch andere Gateways entsprechend der Dokumentation in Ihr Payment-System integrieren.

Das HTML-Template wird im Beispielshop bekanntlich über die Constants des TypoScript-Templates `tt_products` eingebunden:

```
file.templateFile = fileadmin/templates/tt_products_css.html
```

Auch die Einbindung im Setup wäre möglich:⁷

```
templateFile = fileadmin/templates/tt_products_css.html
```

⁷ Beachten Sie in diesem Falle, dass das in den Constants notwendige, vorangestellte `file.` entfällt.

Damit das Bezahlskript oder andere Texte im Warenkorb, auf der Seite der Adresseingabe, der Zahlung, der Bestellbestätigung oder der E-Mail-Bestätigung darauf zugreifen können, muss das HTML-Template angepasst werden, und zwar im Subpart MESSAGE_PAYMENT, der wiederum weitere Subparts wie MESSAGE_PAYMENT_90 enthält. Für jede Zahlungsoption (analog zu den Versandoptionen unter MESSAGE_SHIPPING) kann es einen eigenen Eintrag geben. Am besten kennzeichnen Sie alle Subparts auch mit begin- und end-Kommentaren. Die Zuordnung erfolgt über die Zahl aus dem Setup, die Sie für eine Bezahlmethode festgelegt haben:

```
plugin.tt_products.payment {  
  
    #radio=0  
  
    TAXpercentage = 7  
  
    10.title = Rechnung  
    10.where.static_countries = cn_short_local = 'Deutschland'  
    ...  
    90.title = PayPal  
    ...  
}
```

Auf diese Weise können Sie zu jeder vom Kunden gewählten Zahlungsmethode einen passenden Text im Shop anzeigen lassen.

```
<!-- ###MESSAGE_PAYMENT###  
    This whole subpart, message_payment is substituted with the  
    subpart 'message_payment_[number-of-method-in-typoscript]' begin -->  
<!-- ###MESSAGE_PAYMENT_90### begin -->  
Zahlung via PayPal  
<!-- ###MESSAGE_PAYMENT_90### end -->  
...  
<!-- ###MESSAGE_PAYMENT### end -->
```

Diese Zeilen werden nur dann angezeigt, wenn die entsprechende Zahlungsmethode (hier 90) in der Select-Box ausgewählt wurde. Also fügen Sie beispielsweise folgende Zeilen am Ende des BASKET_PAYMENT_TEMPLATE ein.

```
<!-- ###BASKET_PAYMENT_TEMPLATE### start -->  
...  
<!-- ###MESSAGE_PAYMENT###  
    This whole subpart, message_payment is substituted with the  
    subpart 'message_payment_[number-of-method-in-typoscript]' begin -->  
  
    <!-- PayPal start -->
```

```

<!-- ###MESSAGE_PAYMENT_90### -->
    ###MESSAGE_PAYMENT_SCRIPT###
<!-- ###MESSAGE_PAYMENT_90### -->
<!-- PayPal stop -->
<!-- ###MESSAGE_PAYMENT### end -->
<!-- ###BASKET_PAYMENT_TEMPLATE### end -->

```

Der Marker MESSAGE_PAYMENT_SCRIPT wird durch die späteren Ausgaben des Bezahlskriptes ersetzt. Wo zuvor nur der Text „Zahlung via PayPal“ zu sehen war, erscheint nun der Button mit dem Link zu einem Gateway wie PayPal. Für jedes Gateway wird eine spezifische Template-Datei konfiguriert (sehen Sie dazu bitte auch den Abschnitt ab Seite 153). Beachten Sie, dass das Shop-Template immer ein Formular enthält, dass aber auch das Gateway-Skript ein eigenes Formular enthalten kann.

Wer die Ausgaben des Gateway-Skripts an anderen Stellen platzieren will, kann das nur durch weitere Änderungen an den beiden HTML-Templates erreichen: sowohl am HTML-Template des Shops als auch an dem der PayPal-Suite. Sie müssen darauf achten, dass keines der beiden benötigten Formulare in das andere verschachtelt wird. Das Formular des Shops hat den Shop als Ziel, das Formular des Gateways muss jedoch auf das Gateway führen.

Will man verhindern, dass der Bestellvorgang auch ohne erfolgte Zahlung abgeschlossen werden kann, muss der Button **Bestellung abschließen** versteckt werden, sobald der Kunde Zahlung über PayPal (oder ein anderes Gateway) ausgewählt hat. Andernfalls wäre der Abschluss einer Bestellung ohne vorherige Zahlung möglich.

Der Kaufvorgang ließe sich durch einen Eintrag in den Constants des TS-Templates des Shops auch so organisieren, dass der eigentliche Zahlvorgang erst nach Abschluss der Bestellung startet. Setzen Sie dort `paymentActivity = finalize`.

Dieser Eintrag bewirkt, dass die Ausgaben der Payment Library nicht auf der Payment-Seite (**Warenkorb: Kontrolle und Bezahlung** im Subpart BASKET_PAYMENT_TEMPLATE) angezeigt wird, sondern auf der Seite der Bestellbestätigung (**Warenkorb: Bestellung abschließen** im Subpart BASKET_ORDER_CONFIRMATION_TEMPLATE).

Das Vorgehen ist nun Folgendes: Für jede im Setup definierte Zahlungsmethode aktivieren Sie den Button **Bestellung abschließen**, nur nicht bei Zahlung über ein Gateway. Um das zu erreichen, müssen Sie das zu `tt_products` gehörende Template `paymentlib.tpl` bearbeiten (**ExtManager | Shop System | Edit files**). Achtung: Legen Sie zunächst eine Sicherung der Template-Dateien an, bevor Sie diese verändern!

Reduzieren Sie `paymentlib.tpl` der „PayPal-Suite“ zunächst auf:

```
<!-- ###PAYMENTLIB_FORM_TEMPLATE### start

This subpart is used to display the card number information dialog
before the transfer to the payment gateway
-->

<form class="paypal_form" action="###REDIRECT_URL###"
method="post" autocomplete="off" name="connectForm">
<input type="submit" name="pay" value="bezahlen mit PayPal!">
###HIDDEN_FIELDS###
</form>

<!-- ###PAYMENTLIB_FORM_TEMPLATE###          end -->
```

In diesem Fall können Sie, damit der Button rechts unten platziert bleibt, die Position für die Gateway-Bezahlung weiter nach oben schieben. Dafür muss aber das Ende des Formulars nach oben rücken, damit die Formulare nicht ineinander verschachtelt werden.

Ein Konstrukt wie

```
<form name="meinFormular" action="index.php?id=23">
<form name="meinFormular" action="paypal.com">
  <input type="button" name="gateway" value="zum Gateway"></td>
</form>
</form>
```

ist in HTML nicht erlaubt und führt dazu, dass mit einem Klick auf den Button **zum Gateway** nicht auf das Gateway gewechselt wird. Prüfen Sie also unbedingt die aus der Verschachtelung der beiden HTML-Templates resultierende Gesamt-HTML-Syntax mit einer Spezialsoftware wie einem HTML-Validator oder mit den entsprechenden Tools des Firefox-Browsers.

Alternativ können Sie aber auch mit JavaScript arbeiten und das Formular dynamisch so verändern, dass das Ziel der FORM nach Klick auf **zum Gateway** zum Paypal-Server führt. So haben Sie nur ein einziges Formular, und der Ziel-Server wird vor dem Absenden dynamisch über JavaScript gesetzt, abhängig davon, welcher Button geklickt wurde: Bei **zurück zum Warenkorb** bleibt alles unverändert, bei **Bezahlen über PayPal** wird die URL des PayPal-Servers als FORM-URL eingetragen.

Der Subpart MESSAGE_PAYMENT im HTML-Template `tt_products.html` hat dann als Ergebnis etwa folgende Form:

```
<!-- ###MESSAGE_PAYMENT### begin -->
<!-- ###MESSAGE_PAYMENT_70### -->
<p>Sie bezahlen über den Paypal Gateway nach Abschluss
  der Bestellung.</p>
<!-- ###MESSAGE_PAYMENT_70### -->
<!-- ###MESSAGE_PAYMENT_80### -->
```

```

<p>Sie bezahlen über den iPayment Gateway nach Abschluss
  der Bestellung.</p>
<!-- ###MESSAGE_PAYMENT_80### -->
<!-- ###MESSAGE_PAYMENT_90### -->
###MESSAGE_PAYMENT_SCRIPT###
<!-- ###MESSAGE_PAYMENT_90### -->
<!-- ###MESSAGE_PAYMENT_100### -->
###MESSAGE_PAYMENT_SCRIPT###
<!-- ###MESSAGE_PAYMENT_100### -->
<!-- ###MESSAGE_PAYMENT### end -->

<form method="post" action="###FORM_URL###" name="###FORM_NAME###"
  id="paymentForm">
<input type="submit" name="products_update" value="zurück zum
  Warenkorb">
<input type="hidden" name="products_cmd" value="cardno" />
<input type="submit" name="products_finalize" value="Bestellung
  abschließen!"
  onclick="document.forms[0].action='###FORM_URL_THANKS###';
  document.forms[0].target='###FORM_URL_TARGET###';">
</div>
</div>

```

Der untere Subpart MESSAGE_PAYMENT wurde also über das Formular des Shop-Templates (begrenzt durch die HTML-Tags `<form>...</form>`) verschoben. Der Marker MESSAGE_PAYMENT_SCRIPT kann nun selber wieder ein eigenes Formular (`<form>...</form>`) enthalten. Damit könnte eine Verschachtelung dieser beiden Formulare vermieden werden.

Achten Sie auch auf die Template-Subparts. Zwei Subparts dürfen sich nicht überschneiden.

```

<!-- ###SUBPART_A### begin -->

<!-- ###SUBPART_B### begin -->

<!-- ###SUBPART_A### end -->

<!-- ###SUBPART_B### end -->

```

Erlaubt sind nur Anordnungen wie

```

<!-- ###SUBPART_A### begin -->

<!-- ###SUBPART_A### end -->

<!-- ###SUBPART_B### begin -->

<!-- ###SUBPART_B### end -->

```

oder

```
<!-- ###SUBPART_A### begin -->

<!-- ###SUBPART_B### begin -->

<!-- ###SUBPART_B### end -->

<!-- ###SUBPART_A### end -->
```

Den Abschnitt `MESSAGE_PAYMENT_<zahl>` müssen Sie für jeden Setup-Eintrag mit den entsprechenden Nummern wiederholen und anpassen sowie im Setup das geänderte Payment-Template eintragen:

```
payment {
...
    90.templateFile = fileadmin/template/paymentlib.tpl
...
}
```

Erweitern Sie das bisherige Setup des Shops im Bereich Payment, indem Sie die obige Zeile in Ihrem Template-Setup einfügen.

Achtung: Wenn ein Kunde Gutscheinpunkte eingelöst hat und seine Bestellung über PayPal bezahlt, müssen Sie andere Preismarker verwenden, und zwar jene, die den Gesamtpreis ohne die eingelösten Gutscheinpunkte ausweisen (vgl. Seite 152). Allerdings sind dann die Produktlisten von PayPal nicht mehr korrekt, weil die Summe der Einzelpreise nicht mehr den Endpreis ergibt. Der Shop müsste hierfür erweitert werden.

Checkliste

- Alle Extensions installiert? (Nur bei der Extension-Variante relevant.)
- E-Mail-Adresse im Formular bzw. in der Extension korrekt? – Hier muss die PayPal-E-Mail-Adresse stehen.
- Richtige PHP-Version (5 und größer)? (Nur bei der Extension-Variante relevant.)
- Verfügt der Kunde über ein PayPal-Konto?
- Preis mit Dezimalpunkt?
- Gesamtpreis exkl. MwSt. größer als EUR 1,-? (ERROR: Setting details of transaction failed)

Neben Problemen bei der Implementierung können natürlich auch deutlich banalere Gründe einen Zahlungsabschluss verhindern: falsche Kreditkartendaten (vertippt, alte Karte etc.), die Karte verfügt nicht über den

notwendigen Kreditrahmen, der Kunde ist bei PayPal nicht registriert o.Ä. Tippfehler werden als ungültiger Versuch behandelt; scheitern mehrere Versuche, kann der Kunde (allerdings nur bei der PayPal-Suite-Variante von tt_products) nach dem Redirect zum Shop eine andere Zahlungsmethode auswählen, um die Bestellung doch noch abzuschließen.

6.6.3 ipayment über Formular

Als Alternative oder Ergänzung zur Zahlungsabwicklung über PayPal wollen wir nun ein Gateway beschreiben, das die Kreditkartenzahlung ohne vorherige Anmeldung des Kunden abwickelt; hier ist die Zahl der Anbieter hoch und deren technische Lösungen bzw. Vorgaben für den Shopbetreiber vielfältig. Zur Veranschaulichung des generellen Vorgehens und Beispiel für eine lauffähige Implementierung beschränken wir uns auf *ipayment* des Webhosters 1&1. Als Kunde (in diesem Falle also als Shopbetreiber) stehen Ihnen vier verschiedene sog. „Modi“ zur Verfügung, von denen wir im Folgenden den „Normalen Modus“ beschreiben.⁸

Wichtig: *ipayment* kennt weder Dezimalpunkt noch -komma – alle Beträge müssen in Cent angegeben werden! Da tt_products jedoch mit Kommazahlen arbeitet, muss zunächst über den Extension Manager die Datei tt_products/view/class.tx_ttproducts_basket_view.php bearbeitet und die Umrechnung in Cent angewiesen werden. Fügen Sie dazu die entsprechende Zeile aus dem Beispielcode ein. Wenn Sie mit dem Gutscheinsystem arbeiten, muss der Preismarker des Gutscheinsystems ebenfalls in Cent umgerechnet werden.⁹

```
// This is the total for everything
$this->basket->getCalculatedSums();
$markerArray['###PRICE_TOTAL_TAX###'] = $this->price->priceFormat($this->basket->calculatedArray['price']['total']);
// AHK: Umrechnung in Cent
$markerArray['###PRICE_TOTAL_CENT_TAX###'] =
    intval(100*$markerArray['###PRICE_TOTAL_TAX###']);
$markerArray['###PRICE_TOTAL_NO_TAX###'] = $this->price->priceFormat($this->basket->calculatedArray['priceNoTax']['total']);
$markerArray['###PRICE_TOTAL_ONLY_TAX###'] = $this->price->priceFormat($this->basket->calculatedArray['price']['total']-$this->basket->calculatedArray['priceNoTax']['total']);

$markerArray['###PRICE_VOUCHERTOTAL_TAX###'] = $this->price->priceFormat($this->basket->calculatedArray['price']['vouchertotal']);
```

⁸ Eine detaillierte Beschreibung finden Sie hier: http://www.1und1.info/downloads/ipayment-Technik_v1.0.pdf.

⁹ Falls Sie diese Dateien aufgrund der Rechtevergabe im TYPO3-Systems nicht editieren dürfen, müssen Sie die entsprechende Option über das Installtool aktivieren (siehe auch Anhang B).

```
// AHK: Umrechnung in Cent
$markerArray['###PRICE_VOCHERTOTAL_CENT_TAX###'] =
    intval(100*$markerArray['###PRICE_VOCHERTOTAL_TAX###']);
$markerArray['###PRICE_VOCHERTOTAL_NO_TAX###'] = $this->price-
->priceFormat($this->basket->calculatedArray['priceNoTax']['vouchertot
al']);

$this->paymentsshipping->getSubpartArray($subpartArray,
    $markerArray, $t['basketFramework']);
$this->basket->fe_users->getWrappedSubpartArray($subpartArray,
    $wrappedSubpartArray);
```

Im Abschnitt BASKET_ORDERCONFIRMATION_TEMPLATE des HTML-Templates muss der Marker PRICE_TOTAL_TAX bzw. PRICE_VOCHERTOTAL_TAX ebenfalls durch PRICE_TOTAL_CENT_TAX bzw. PRICE_VOCHERTOTAL_CENT_TAX ersetzt werden.

Im TS-Setup des Shop-Templates definieren Sie zunächst die mögliche Zahlungsmethode – im HTML-Template des Beispielshops wurde der Bereich des MESSAGE_PAYMENT bereits der Methode 80 zugewiesen. Falls Sie andere Zahlen im Setup verwenden, müssen Sie das HTML-Template an dieser Stelle ändern.

```
plugin.tt_products.payment {
    ...
    80.title = ipayment
}
```

Im HTML-Template finden Sie bereits das Formular für die Bezahlung über ipayment vor, im Subpart BASKET_ORDERCONFIRMATION_TEMPLATE.

```
<!-- ###MESSAGE_PAYMENT_80### -->

<p>Hier bezahlen Sie über das ipayment Gateway.</p>

<form target="ipayment" name="ipayment"
action="https://ipayment.de/merchant/99999/processor.php" method="post">

    <input type="hidden" name="trxuser_id" value="99999">
    <input type="hidden" name="trxpassword" value="0">
    <input type="hidden" name="trx_amount"
value="###PRICE_TOTAL_CENT_TAX###" />
    <input type="hidden" name="trx_currency" value="EUR">
    <input type="hidden" name="trx_typ" value="auth">
    <input type="hidden" name="trx_paymenttyp" value="cc">
    <input type="hidden" name="redirect_url"
value="http://mytypo3domain.de/thanks">

    <input type="submit" name="submit" id="button2"
value="Bezahlung mit Kreditkarte via iPayment">
```

```
alt="Bezahlen mit Kreditkarte via iPayment">

</form>

<!-- ###MESSAGE_PAYMENT_80### -->
```

Das Formular ist – genau wie bei der PayPal-Methode – wieder in einen MESSAGE_PAYMENT-Subpart eingebettet. Zu Testzwecken ist die ID des Testaccounts 99999 aktiviert. Durch die ID und das Passwort, die Sie als ipayment-Kunde erhalten, identifizieren Sie sich gegenüber dem Server. Wenn Sie in Ihrem Shop das Gutscheinsystem verwenden, muss an Stelle von PRICE_TOTAL_CENT_TAX der Marker PRICE_VOCHERTOTAL_CENT_TAX gesetzt sein. Über den Parameter currency weisen Sie die Währung zu; auth bestimmt, dass die Zahlung sofort erfolgt. Der Paymenttyp cc steht für die Zahlung mit Kreditkarte. Der Parameter für das Redirect führt den Benutzer zurück zum Shop auf eine „Dank-Seite“.

Mit diesem Formular können Sie Ihre Tests bereits durchführen – sofern Sie die URL für den Redirect ändern.

Weitere Angaben, wie z. B. die Adressdaten des Benutzers, können Sie in diesem Formular ebenfalls übergeben. Eine genaue Liste der Parameter finden Sie in dem oben genannten PDF des Anbieters. Im Beispiel haben wir uns gegen die Option entschieden, die Adressdaten des Kunden an das Formular zu übergeben. Der Kunde muss seine Anschrift beim Gateway selbst eintragen. Achten Sie beim Testen darauf, dass hier eine existierende Adresse eingegeben wird – das System prüft, ob diese Adresse existieren könnte.

Mit diesem Formular wechseln Sie auf die Seite des Payment-Anbieters. Dessen Bezahlformular ist nicht unbedingt ansprechend, und Sie können es in einem anderen Modus selbst anpassen. Wählen Sie ein Design, das dem Ihres Shops entspricht, erfolgt der Wechsel zum Gateway (also zur Seite des Zahlungsanbieters) für Ihren Shopkunden nahezu unmerklich; ob das allerdings wünschenswert ist, müssen Sie selbst entscheiden.

Wenn Sie sich den HTML-Quelltext des Shops ansehen, sehen Sie auch Formularaten und Kundennummer des Shopbetreibers – das ist vollkommen in Ordnung, denn es handelt sich nicht um (sicherheitsrelevante) Zugangsdaten.

Grundsätzlich sollte Ihre Webseite für die Zahlung mit ipayment immer HTTPS unterstützen, also per SSL gesichert sein. Allerdings funktioniert die Zahlung auch ohne diese Maßnahme.

Scheitert die Transaktion aufgrund falscher Kreditkartenangaben, hat der Kunde bei ipayment übrigens die Möglichkeit, andere Karten anzugeben. Abhängig von der Einstellung des „Sicherheitssystems“ ist die Anzahl der Versuche begrenzt – das ipayment-System sperrt dann die entsprechende IP. Diese Einstellungen werden im ipayment-Konfigurationsmenü vorgenommen.

Mit `tt_products` können Sie dann nach dem Redirect zum Shop eine andere Kreditkarte oder eine andere Bezahlmethode auswählen, um den Bestellvorgang fortzusetzen.

6.6.4 ipayment über Extension

Wie bei PayPal besteht auch bei der Zahlung über `ipayment` die Alternative, das Verfahren über eine entsprechende TYPO3-Extension zu implementieren.

Bearbeiten Sie zunächst die HTML-Template-Dateien, wie ab Seite 149 erläutert. Die Vorgangsweise ist hier analog zu der im Kapitel über PayPal beschriebenen.

Folgende Extensions müssen installiert werden:

- `lib` 0.0.20+
- `div` 0.0.9+
- `static_info_tables` 2.0.5+
- `paymentlib` 0.2.1+
- `paymentlib_ipayment` 0.1.1+ (Payment method „iPayment“)

Fertigen Sie eine Kopie von `paymentlib.tpl` im Verzeichnis `template` von `tt_products` an. Danach modifizieren Sie diese nach Ihren Wünschen.

```
<form action="###REDIRECT_URL###" method="post" autocomplete="off"
name="ipayment" >
<input type="submit" name="pay"
value="Verbinden zum Gateway ###PAYMENTLIB_TITLE###
Server">&nbsp;&nbsp;&nbsp;
###HIDDEN_FIELDS###
</form>

<table BORDER="0" CELLPADDING="0" CELLSPACING="5">
<TR>
<TD><FONT FACE="Verdana, Arial, Helvetica"
SIZE=1>###PAYMENTLIB_INFO###</FONT></TD>
</TR>
</table>

<!-- ###PAYMENTLIB_FORM_TEMPLATE###          end -->
```

Im Setup des Shop-Templates müssen Sie (im Beispiel unter der Nummer 100) für `ipayment` den Marker `MESSAGE_PAYMENT_SCRIPT` einfügen.

```

<!-- ###MESSAGE_PAYMENT### end -->
...
<!-- ###MESSAGE_PAYMENT_100### -->
###MESSAGE_PAYMENT_SCRIPT###
<!-- ###MESSAGE_PAYMENT_100### -->
<!-- ###MESSAGE_PAYMENT### end -->

```

Für die `ipayment`-Extension ist es erforderlich, dass Sie die Kreditkarte bereits im Setup eintragen.

```

100.title = IPayment American Express - vor Bestellung
100.handleLib = paymentlib
100.handleLib {
  extName = paymentlib_ipayment
  extTitle = iPayment
  templateFile = fileadmin/osp/templates/ipayment_paymentlib.tpl
  gatewaymode = form
  paymentMethod = paymentlib_ipayment_cc_americanexpress
  Currency = EUR
}

```

Sie müssen für jede Kreditkartengesellschaft einen eigenen Eintrag im Setup machen. Der Typ der Kreditkarte wird ins Setup unter `paymentMethod` eingetragen.

In der Datei `paymentmethods.xml` im Verzeichnis der Extension `paymentlib_ipayment` sind die Bezahlmethoden aufgeführt.

```

paymentlib_ipayment_cc_visa
paymentlib_ipayment_cc_mastercard
paymentlib_ipayment_cc_americanexpress
paymentlib_ipayment_elv

```

Nach erfolgreicher Bezahlung wird die Bestellung im Shop durchgeführt. Wenn Sie aber auf **Zurück** klicken, gelangen Sie wieder zurück in den Warenkorb des Shops.

7

Kapitel

Calendar Base

Eine sinnvolle Ergänzung zum Shop ist der von Mario Matzulla entwickelte *Calendar Base*. Einsatzszenario für dessen Anbindung an `tt_products` sind beispielsweise saisonabhängige Angebote. Zwar bietet auch der Start-/Stopp-Mechanismus von `tt_products` ähnliche Funktionalität, umfangreicher und flexibler sind die Konfigurationsmöglichkeiten allerdings mit diesem separaten Kalender-Tool.

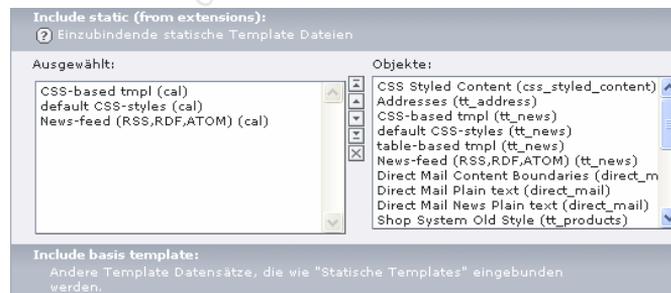
Um Saisonartikel also zum Beispiel nicht nur im Shop darzustellen, sondern dem Kunden auch in einer Kalenderansicht zu präsentieren, werden `tt_products` und *Calendar Base* miteinander verknüpft und die entsprechenden Artikel mit Beginn der Saison als „Ereignisse“ hinterlegt. Per Klick auf den Eintrag im Kalender erfolgt dann die Weiterleitung zur Einzelansicht samt Bestellmöglichkeit. Das setzt natürlich voraus, dass die Warenkorbfunktionalitäten auch in der Einzelansicht im Subpart `PID_ITEM_DISPLAY` definiert sind.

7.1 Installation

Installieren Sie zunächst die Extension Calendar Base (cal) über den Extension Manager; bitte beachten Sie, dass sich die im Beispielshop eingesezte Version 0.16.0 noch im Beta-Stadium befindet. Installieren Sie außerdem die Kalender-Services `cal service for typoscript table configs` (`cal_ts_service`), die für die Einbindung von Tabellen – in diesem Falle von `tt_products` - via TypoScript notwendig sind (Abbildung 7.1).

Erstellen Sie in Ihrem Template-Ordner ein TS-Template namens `+cal`. Weisen Sie die statischen Templates zu und speichern Sie den Datensatz ab. Dieses Template binden Sie in gewohnter Weise in Ihr Haupttemplate ein.

Abbildung 7.1:
Statisches Template



Im nächsten Schritt erstellen Sie die Kalenderseiten; in diesem Beispiel ist **Kalenderübersicht** die Hauptseite des Kalenders.

Weiterhin legen Sie auf dieser Ebene einen SysOrdner für die Kalendereinträge („Ereignisse“) an. Dieser muss – wie sollte es anders sein – den Kalenderseiten später als Ausgangspunkt zugewiesen werden.

7.1.1 Jahresansicht

Auf der Startseite des Kalenders fügen Sie nun das Plugin **TYPO3 Kalender** ein und definieren die Anzeige über die Flexforms. In der Registerkarte **Allgemeine Einstellungen** (Abbildung 7.2) wählen Sie die möglichen Ansichten aus der Objektliste aus, in unserem Fall Jahres-, Monats-, Wochen-, Tages-, Suche- und Ereignisansicht. Die Ereignisansicht ist nichts anderes als eine Einzel- bzw. Detailansicht eines Termins.

Tragen Sie den Namen des Kalenders (hier **Kalender**) ins Feld **Kalender** ein und wählen Sie als Wochenstarttag den Montag aus. **Ausgangspunkt** ist der eben erstellte Kalender-SysOrdner sowie der (oder die) Produkte-SysOrdner. Alle SysOrdner, aus denen Produkte im Kalender angezeigt werden sollen, müssen zugewiesen werden.

Erweiterung:
Wählen Sie das Plugin aus, das Sie einfügen wollen. (Verfügbarkeit hängt vom Template ab).

TYPO3 Kalender

Plugin Optionen:
FlexForm Konfiguration von Plugins.

Allgemeine Einstellungen | Kategorie | Jahres-Ansicht | Monats-Ansicht

Wochen-Ansicht | Tages-Ansicht | Listen-Ansicht | Ereignis-Ansicht

Ics-Ansicht | Andere Ansichten | Typoscript

Erlaubte Ansichten

Ausgewählt:

- Jahres-Ansicht
- Monats-Ansicht
- Wochen-Ansicht
- Tages-Ansicht
- Ereignis-Ansicht

Objekte:

- Monats-Ansicht
- Jahres-Ansicht
- Wochen-Ansicht
- Tages-Ansicht
- Ereignis-Ansicht
- Listen-Ansicht

Kalendername: Kalender

Abonnieren erlauben:

FE-Benutzern abonnieren erlauben:

Wochenstarttag: Montag

Ausgangspunkt:
Seite(n) einfügen

Produkte
Kalendereinträge

Produkte [236]

Seite

Abbildung 7.2:
Flexforms
Registerkarte
Allgemeine
Einstellungen

Auf der Registerkarte **Jahresansicht** weisen Sie die Seite für die Jahresansicht zu, die hier der aktuellen Seite entspricht. Sofern diese Übereinstimmung gegeben ist, muss die Seite nicht unbedingt auf der Registerkarte eingetragen werden – schaden tut es aber auch nicht. Entsprechend gehen Sie auf den Registerkarten **Tages-**, **Wochen-**, und **Monatsansicht** vor, die ebenfalls auf der Kalenderhauptseite angezeigt werden sollen.

Für die Registerkarte **Tagesansicht** (Abbildung 7.3) vergeben Sie außerdem eine Start- und eine Endzeit für den Kalendertag sowie eine Einheit für die Unterteilung der Stunden (hier 15 Minuten).

Erweiterung:
Wählen Sie das Plugin aus, das Sie einfügen wollen. (Verfügbarkeit hängt vom Template ab).

TYPO3 Kalender

Plugin Optionen:
FlexForm Konfiguration von Plugins.

Allgemeine Einstellungen | Kategorie | Jahres-Ansicht | Monats-Ansicht

Wochen-Ansicht | Tages-Ansicht | Listen-Ansicht | Ereignis-Ansicht

Ics-Ansicht | Andere Ansichten | Typoscript

Tages-Ansicht Seite: Kalenderübersicht

Tages Startzeit (hhmm): 07:00

Tages Endzeit (hhmm): 23:00

Unterteilt die Stunde alle x Minuten: 15

Ausgangspunkt:
Seite(n) einfügen

Produkte
Kalendereinträge

Produkte [236]

Seite

Abbildung 7.3:
Flexforms
Tagesansicht

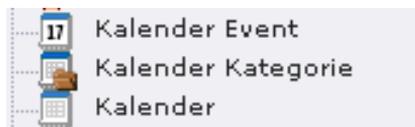
Erstellen Sie nun eine weitere Seite auf der Ebene der Kalenderübersicht und nennen Sie diese **Aktuelle Angebote**. Fügen Sie das Kalender-Plugin ein und weisen über die **Allgemeinen Einstellungen** als erlaubte Ansichten **Listen-** und **Ereignis-Ansicht** zu.

Für die Registerkarte der Listenansicht (Abbildung 7.4) können Sie Start- und die Endzeit zuweisen und festlegen, wie viele Ereignisse in der Liste maximal angezeigt werden dürfen. In unserem Beispiel sind jedoch Start- und Endzeit über das TS-Setup definiert. Wählen Sie dazu in den Flexforms **Use Typoscript Default**.¹ Als Endzeit soll **1 year from now** gelten, und es erscheinen höchstens 100 Ereignisse. Diese Definition findet sich ebenfalls im TS-Setup des Kalenders. Weisen Sie der Seite jetzt als Ausgangspunkt den SysOrdner mit den Produkten zu.

Abbildung 7.4:
Flexforms Listen-
und Ereignisansicht

Im SysOrdner der Kalendereinträge erstellen Sie nun zunächst einmal einen ersten Kalender über **Neuen Datensatz anlegen | Kalender** (Abbildung 7.5).

Abbildung 7.5:
Kalender anlegen



¹ In älteren Versionen des Kalenders gab es an dieser Stelle bisweilen Probleme: Ereignisse mit dem aktuellen Datum als Startdatum wurden nicht angezeigt; dort musste das Startdatum auf `yesterday` bzw. `-1` gesetzt werden.

Vergeben Sie den Namen Kalender im Backendformular und speichern Sie den Datensatz ab. Der zweite Eintrag, im selben SysOrdner anzulegen, ist ein erstes Ereignis (**Neuen Datensatz anlegen | Calendar Event**). In der Registerkarte **Allgemein** des zugehörigen Backendformulars (Abbildung 7.6) wählen Sie **Ereignis mit Beschreibung**, was der Voreinstellung entspricht, vergeben einen Titel und weisen Start- und Enddatum zu.

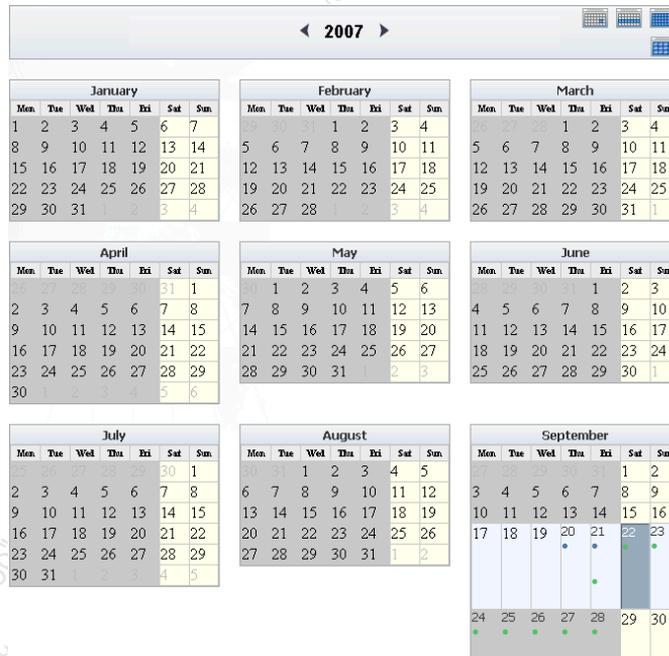
The screenshot shows the 'Allgemein' tab of a calendar event creation form. The form is divided into several sections:

- Ereignis Typ:** A dropdown menu is set to 'Ereignis mit Beschreibung'. Below it is a help icon and the text 'Hier können Sie den TYP des Ereignis einstellen'.
- Titel:** A text input field contains 'Ein Termin'. Below it are 'Verstecken:' (checkbox) and 'Sprache:' (dropdown set to 'Standard').
- Start:** 'Event Start Tag' is '18-09-2007'. 'ganztägiges Ereignis' is checked.
- Ende:** 'Event Ende Tag' is '18-09-2007'.
- Kalender:** A dropdown menu is set to 'Kalender'.
- Kategorie:** A section with the text 'Hier können Sie eine Kategorie für Ihr Ereignis auswählen.' It contains two empty panes: 'Ausgewählt:' and 'Objekte:'.
- Beschreibung:** A section with the text 'Geben Sie hier eine möglichst aussagekräftige Beschreibung des Ereignisses ein.' It includes a rich text editor with a toolbar (font, bold, italic, underline, link, etc.) and a text area containing 'Dies ist ein erster Test für einen Kalendereintrag.'

Abbildung 7.6:
Kalender-Ereignis
anlegen

Setzen Sie den Haken bei **ganztägiges Ereignis**, damit die Produktanzeige von einer Uhrzeit unabhängig ist. Im Feld **Kalender** erscheint daraufhin der soeben erstellte Kalender; sollten Sie mehrere anlegen, erscheint eine Auswahl. Zusammen mit einem kurzen Beschreibungstext für das Ereignis speichern Sie den Datensatz ab. Die Kategorien bleiben unberücksichtigt.

Abbildung 7.7:
Jahresansicht



7.1.2 TS-Template

Wechseln Sie nun in den Template-Ordner und wählen Sie das Template `+cal` zur Bearbeitung aus. Kopieren Sie den folgenden Code zur Konfiguration des Kalenders in das TS-Setup:

```
plugin.tx_cal_controller {
    view {
        list {
            listTemplate = fileadmin/templates/cal_list.tmpl
            event < plugin.tx_cal_controller.view.list.event

            event {
                # @description The default image path is related to the
                # tx_cal_{object type}. Therefore the path has to be
                # overwritten
            }
        }
    }
}
```

```

        image.overridePath = uploads/pics/
        image.maxW = 100
        price = TEXT
        price {
            dataWrap =
        {LL:EXT:tt_products/locallang_db.xml:tt_products.price} |
        }
        text1 = TEXT
        text1 {
            wrap = Angebot:
        }
    }
} # Ende List

} # Ende View

display {
    tt_products {
        # @description This is the heart of the connection SQL.
        # It will be used in conjunction with the view-where
        # (findAllWithinWhere, findAll and findWhere) to retrieve
        # the according records.
        event_select {
            selectFields = tt_products.*
        }

        # @description This is the heart of the connection SQL.
        # It will be used in conjunction with the view-where
        # (findAllWithinWhere, findAll and findWhere) to retrieve
        # the according records plus their categories.
        event_select_with_cat {
        }

        # @description Configure the table the records come from.
        pidTable = tt_products

        # @description Define the where clause for the views:
        # day, week, month, year, list, rss
        # Use ###START### and ###END### as marker to be replaced
        # during runtime with the timeframe
        findAllWithinWhere =
        ((tt_products.sellstarttime>###START###
        AND tt_products.sellstarttime<###END###)
        OR (tt_products.sellendtime<###END###
        AND tt_products.sellendtime>###START###)
        OR (tt_products.sellendtime>###END###
        AND tt_products.sellstarttime<###START###))

        # @description Define the where clause for the single
        # event views: event, ics
        # Use ###START### and ###END### as marker to be replaced
        # during runtime with the timeframe

```

```

findWhere = tt_products.uid = ###UID###

# @description Define the field containing the timestamp
# for the start time
startTimeField = sellstarttime

# @description Define the field containing the timestamp
# for the end time
endTimeField = sellendtime

# @description Enable this to create a link to another
# page containing the single view of the external plugin.
# Use ###DB_FIELD### to retrieve record related
# informations from the database (DB_FIELD is a
# placeholder)
externalPlugin = 1

externalPlugin {

# @description Define the pid for the external plugin
# single view
singleViewPid = 237

# @description Array containing the url paramter to be
# added to the url
additionalParams = tx_ttproducts_pil[product]=###UID###
}

# @description If there is no end time, you can define
# a default length (in minutes)
defaultLength =

# @description For a quick and easy integration, map
# your record fields to the standard event fields
# You can always use ###DB_FIELD### (where DB_FIELD is
# a placeholder for a database row entry) to insert
# database values
fieldMapping {
    title = title
    description = note
    location =
    organizer =
    image = image
}

# @description The template to be used with these records
template = EXT:cal_ts_service/template/ts.tmpl

# @description The header style class to be used
headerStyle = green_catheader

# @description The body style class to be used

```

```

        bodyStyle = green_catbody

        # @description The title in the legend description for
        # these records
        legendDescription = Angebote

        search {
            # @description The fields, which are allowed to be
            # searched through
            searchEventFieldList = title,description
        }

        # @description Define the SQL to retrieve the
        # categories
        cat_select {
        }

        # @description The table containing the category
        # records
        catTable =

    }
}

```

`view` definiert die Ansichten, und zwar zunächst die Listenansicht, der ein HTML-Template zugewiesen wird. Innerhalb dieser Listenansicht erscheint eine Einzelansicht des Ereignisses (`event`) für die Produkte. Anschließend wird der Default-Pfad zur zugehörigen Abbildung überschrieben, damit das Produktbild angezeigt werden kann. Die erlaubte Bildgröße ist ebenfalls zuzuweisen. Damit der Preis des Produkts in der Kalenderansicht sichtbar ist, muss das entsprechende Textfeld zugeordnet werden. Den Preis holt der Kalender über den zugewiesenen `Wrap` aus der XML-Sprachdatei `tt_products/locallang_db.xml` von `tt_products`. Ein zusätzlicher Text – in diesem Beispiel „Angebot:“ – wird über den Marker `text1` zugewiesen.

`display` weist dem Kalender die Shop-Extension zu und erlaubt ein `Select` für die Tabelle `tt_products` (inklusive der Kategorien). Zudem werden alle Datensätze mit gesetztem Verkaufsstart und -ende für die Anzeige ausgewählt. Diese neuen Felder werden später im Produktdatensatz mit einem Datum versehen.

Im Bereich `externalPlugin` wird die PID der Einzelansichtsseite des Shops angegeben. `fieldMapping` enthält die Felder, die in der Kalender-Einzelansicht des Produkts erscheinen. Es folgen Angaben zum Service-Template und den Stilen des Kalenders sowie zu den Feldern, in denen der Kalender suchen darf. Im Anschluss werden Anzeige und Auswahl der Kategorien definiert.

Im Frontend erscheint der Kalendereintrag gemäß diesen Vorgaben. Abbildung 7.7 auf Seite 172 zeigt die Jahresansicht des Kalenders und im September bereits Einträge für das Produkt, das Sie allerdings noch zuweisen müssen. Die grünen Punkte (21.-28.9.) bezeichnen das Produkt, die blauen (20. und 21.9.) die normalen Kalendereinträge, unabhängig vom Shop. Die Punkte-Icons können Sie natürlich anpassen.

Die Konfiguration der Listenansicht ergibt sich aus dem Code des HTML-Listentemplates des Kalenders `list.tpl`, das Sie ebenfalls anpassen können. Dieses HTML-Template ist Teil der Kalender-Extension.

Diese Kalender-Konfiguration soll genügen. Für Anpassungen im Detail sei *Firebug* empfohlen, eine Extension für den Webbrowser *Firefox* von Joe Hewitt.² Dieses Tool wird Ihnen bei der Anpassung von HTML-Templates und CSS-Dateien sehr viel Arbeit ersparen. Sie erhalten die Formatierung der Seite in HTML und CSS angezeigt und können in beiden Dateien zu Testzwecken Änderungen vornehmen, die ebenfalls im Browser sichtbar werden. Der Internet Explorer bietet das Developer Tool, das Ihnen ebenfalls Unterstützung bei der Formatierung bietet. An dieser Stelle sei zudem erwähnt, dass der Kalender einige Anpassungen der `div`-Tags nur über den TYPO3-Objectbrowser zulässt. Hilfe finden Sie in der Dokumentation³ und auf einschlägigen Webseiten.

7.2 Saisonartikel

Wenn Sie Ihre Saisonartikel, Vorankündigungen oder Neuheiten über den Kalender mitteilen möchten, müssen Sie das Produkt einem Kalendereintrag zuweisen.

Seit Version 2.6.0 verfügt `tt_products` bei der Produkteingabe über zusätzliche Datumsfelder: **Verkaufsstart** und **Verkaufsende** (Abbildung 7.8). Damit Artikel im Kalender erscheinen, müssen Sie dem Produkt diese Datumsangaben zuweisen.

Abbildung 7.8:
Neue Datumsfelder
im Produkt-
Datensatz

The screenshot shows a form titled 'Verstecken:' with a checkbox. Below it are five fields: 'Verkaufsstart:' with a checked checkbox and a date input containing '18-9-2007'; 'Verkaufsende:' with a checked checkbox and a date input containing '19-9-2007'; 'Start:' with an unchecked checkbox and an empty date input; 'Stop:' with an unchecked checkbox and an empty date input; and 'Zugriff:' with a dropdown menu.

² <http://www.getfirebug.com/>

³ <http://typo3.org/documentation/document-library/extension-manuals/cal/current/>

Produktanzeige in den Kalenderansichten

In diesem Beispiel wurde bereits die Seite **Aktuelle Angebote** für die Produkt-Listenansicht angelegt, das Kalender-Plugin **TYPO3 Kalender** zugewiesen und als erlaubte Ansichten die **spezielle Listenansicht** und die **Einzelansicht** definiert. Als Ausgangspunkt wurde der Produkte-SysOrdner zugewiesen. Alle Produkte, denen das Verkaufsstart- und Verkaufsendedatum zugewiesen wurde, erscheinen auf dieser neuen Seite **Aktuelle Angebote** (Abbildung 7.9). Der Link auf dem Produkttitel führt direkt zur Einzelansicht des Produkts im Shop, wo es der Kunde auch bestellen kann, sofern Sie die Bestellung in der Einzelansicht konfiguriert haben.



Abbildung 7.9:
Produktlistenansicht
im Kalender

Klicken Sie nun in der Einzelansicht des Produkts den Link **zurück zur Liste**, gelangen Sie zur Listenansicht Ihres Shops, *nicht* zurück in die Kalenderansicht. Wollen Sie dieses Verhalten ändern und wieder zum Kalender wechseln lassen, empfiehlt sich der JavaScript-Befehl `history.back()`, der immer zurück zu der Seite führt, von der aus Sie die aktuelle Seite aufgerufen haben.

Ersetzen Sie dazu den Bereich

```
<!--###LINK_ITEM###-->
  zurück zur Liste
  <!--###LINK_ITEM###-->
```

im Subpart `ITEM_SINGLE_DISPLAY` durch den folgenden Code:

```
<a href="javascript:history.back()">zurück zur Liste</a>
```

Suche

Über die Suchfunktion können Sie innerhalb des Kalenders z. B. nach den dargestellten Produkten suchen, denn über das TS-Setup des Kalenders würden die Suchfelder des Produkts definiert. Im TS-Setup ist die Suche erlaubt, in den Flexforms muss sie ebenfalls als erlaubte Ansicht gesetzt sein, damit die Ausgabe auf der gewünschten Seite erfolgen kann, in diesem Fall auf der Seite 314, auf der sich – bis auf die Listenansicht – alle erlaubten

Ansichten befinden. Der eingegebene Suchbegriff listet alle Einträge, die den Suchbegriff enthalten, unabhängig vom Datum. Auf der Ergebnisseite können Sie die Suche weiter einschränken, z. B. durch das Datum.

Abbildung 7.10:
Kalender-Suche

Suche nach Eintrag

Suchen:

Kategorie:

Ort:

Organisator:

Startdatum:

Enddatum:

Suchtext:

Seite 1 von 1 1

Termin im November

Eine Beschreibung des Termins.

Seite 1 von 1 1

Zurück

8

Kapitel

Mehrere Shops in einer TYPO3-Instanz

tt_products ist ein Shopsystem und damit für die Präsentation von Waren und die Abwicklung von Kaufvorgängen konzipiert. Die dazu notwendigen Funktionalitäten (Listenansicht, Einzelansicht, Suche etc.) lassen sich jedoch auch für andere Anforderungen nutzen, von denen wir hier exemplarisch zwei vorstellen möchten.

Es geht also nicht wie bislang primär darum, notwendige Features im Beispielshop zu implementieren, sondern um einen etwas flexibleren Umgang mit der Extension über die eigentliche Shop-Anwendung hinaus. Im Vordergrund stehen dabei Darstellungs- und Listenfunktionen, nicht Warenkorb und Zahlungsabwicklung.

Anders ausgedrückt: Es geht um das Verfahren, mehrere Shops in einer TYPO3-Rootseite zu nutzen, wobei eben nicht jeder dieser „Shops“ dem Online-Verkauf dient. Als Beispiele dienen hier ein Katalog und ein Autorenverzeichnis.

8.1 Katalog

Ziel ist ein Katalog, der das gesamte Verlagsangebot zeigt, jedoch keine Bestellmöglichkeit bietet.

Legen Sie eine Seite namens **Aktueller Katalog** an. Erstellen Sie direkt darin ein Extension-Template und geben Sie diesem den Titel **+Leer Katalog**. Dieses Template bleibt leer! Speichern Sie es ab und binden Sie es über **Include basis template** in das Haupttemplate Ihres Internetauftritts ein.

Erstellen Sie ein weiteres Extension-Template für diesen Katalog im Template-SysOrdner; als Titel vergeben Sie **+Katalog**. Vergessen Sie nicht, diesem das statische Template **Shop System Old Style(tt_products)** zuzuordnen, wenn Sie, wie im Beispielsshop, den Katalog im Seitenbaum vom Shop komplett trennen. Wenn Sie das Shop-Template bereits global eingefügt haben, müssen Sie das statische Shop-Template allerdings kein zweites Mal einbinden – der Template Object Browsers hilft Ihnen dabei, dies zu prüfen. Da hier jedoch die Shops unabhängig voneinander betrieben werden, sollte auch das TypoScript-Setup getrennt sein.

Danach weisen Sie dieses Katalog-Template über **Include basis template** dem leeren Template **+Leer Katalog** zu.

Tragen Sie nun folgenden Code in die TS-Constants ein:

```
# Shop HTML Template
plugin.tt_products.file.templateFile =
    fileadmin/osp/templates/tt_products_katalog.html

# PIDs zu den Shopseiten
plugin.tt_products.PIDlistDisplay = 230
plugin.tt_products.PIDitemDisplay = 237

# Bildgrößen
plugin.tt_products.maxH_basket = 100
plugin.tt_products.maxW_popup = 400
plugin.tt_products.maxH_list = 100
plugin.tt_products.maxW_basket = 75
plugin.tt_products.maxW_single = 100

# Artikel
plugin.tt_products.useArticles = 1

# Größen
plugin.tt_products.selectSize = 1

plugin.tt_products.NoSingleViewOnList = 1
```

Und folgenden Code ins TS-Setup:

```
# Default PAGE object:
page = PAGE
page.typeNum = 0
page.stylesheet = fileadmin/osp/styles/standard.css

# Sortierung nach Titel
plugin.tt_products.conf.tt_products.ALL.orderBy = title

# Default CSS entfernen
plugin.tt_products._CSS_DEFAULT_STYLE =
```

Kopieren Sie nun das HTML-Template des „richtigen“ Shops, speichern Sie es ab und weisen Sie es über die TS-Constants zu. Sie benötigen allerdings nur die Listen- und die Einzelansicht sowie die Suche. Alle anderen Bereiche des HTML-Template werden nicht angesprochen und können gelöscht werden. Entfernen Sie die nicht benötigten Elemente für den Warenkorbbereich in der Listen- und Einzelansicht und speichern Sie das Template ab. Wenn Sie den Produktpreis ausweisen möchten, sollten Sie die Preismarker natürlich nicht entfernen.

Wenn Sie nun auf der Seite **Katalog** das Shop-Plugin einfügen, können Sie als Ausgangspunkt den SysOrdner mit Ihren Produkten angeben. Die Inhalte des Shops und des Katalogs sind in diesem Beispiel identisch. Wenn Sie in Ihrem Katalog zusätzliche Produkte zeigen möchten (beispielsweise Vorkündigungen, die im Shop noch nicht angezeigt werden sollen), erstellen Sie einen weiteren SysOrdner, legen diese Produkte dort an und binden den SysOrdner ebenfalls als Ausgangspunkt der Katalogseite ein. Erstellen Sie eine weitere Shopseite für die Einzelansicht.

Im Frontend erhalten Sie nun Listen- und Einzelansicht Ihrer Produkte ohne die Warenkorbfunktionalitäten. Durch die wegfallenden Felder wird sich die Darstellung etwas verschieben. Sie können die Klassen in diesem Template anders benennen und via CSS neue Werte zuweisen oder die CSS-Datei kopieren, umbenennen und über die Constants des Extension-Template zuweisen.

Noch komfortabler wird der Katalog, wenn Sie Kategorien verwenden. Die Kategorien entsprechen dann quasi einem Inhaltsverzeichnis. Dieses weist wiederum auf die Listenansicht, von der aus Sie die jeweilige Einzelansicht aufrufen.

Das oben erwähnte Template weisen Sie über die Constants des Extension-Template +Katalog zu.

Damit sowohl der Katalog als auch der Shop reibungslos funktionieren, müssen Sie auch für Ihren Shop ein leeres TS-Template auf der Startseite

erstellen, welches auf das Shop-Template im Template-SysOrdner verweist. Dieses leere TS-Template binden Sie in Ihr Haupttemplate ein.

Das leere Template dient dazu, der Seite, auf der es eingefügt wird, und auf den darunterliegenden andere Constants- und Setup-Einträge für die Extension `tt_products` zuzuweisen. Diese Constants und Setup hätten natürlich auch in dem Template direkt eingetragen werden können; um aber die Übersicht zu behalten, ist es sinnvoll, alle relevanten Templates in einem gemeinsamen Template-SysOrdner abzuspeichern. Auf den Seiten selbst verbleiben dann leere Templates, die ein oder mehrere andere aus dem SysOrdner einbinden.

8.2 Autorenverzeichnis

Eine weitere Einsatzmöglichkeit für `tt_products` soll ein Autorenverzeichnis in unserem Beispielshop sein. Hier werden zunächst alle Autoren des Verlags gelistet und in der Einzelansicht vorgestellt. Auch für dieses Beispiel werden wieder nur Listen- und Einzelansicht sowie die Shop-Suche verwendet. Prinzipiell gehen Sie genauso vor, wie beim Katalog-Beispiel.

Erstellen Sie eine Seite für Ihr Autorenverzeichnis, als Seitentitel vergeben Sie **Unsere Autoren**. Erstellen Sie auf dieser Seite ein leeres Extension-Template mit dem Titel `+Leer Autoren` und speichern Sie es ab. Binden Sie dieses Template über **Include basis template** in das Haupttemplate Ihres Internetauftritts ein. Erstellen Sie ein weiteres Template in Ihrem Template-SysOrdner, vergeben Sie den Titel `+Autoren` und speichern Sie es zunächst ab. Dieses Template weisen Sie dem leeren Autorentemplate über **Include basis template** zu.

Kopieren Sie nun wieder das HTML-Template des Shops und entfernen Sie die nicht benötigten Bereiche wie im vorigen Abschnitt beschrieben. Danach speichern Sie das Template ab und weisen es über die TS-Constants (`template.file`) Ihres Extension-Template zu. Binden Sie die zugehörige CSS-Datei ein.

Hier die TS-Constants des Autorenverzeichnisses:

```
# Shop HTML Template
plugin.tt_products.file.templateFile =
    fileadmin/osp/templates/tt_products_autoren.html

# E-Mail und Subject / Domain
plugin.tt_products.orderEmail_fromName =
plugin.tt_products.orderEmail_to =
plugin.tt_products.orderEmail_subject =
plugin.tt_products.orderEmail_from =
```

```

plugin.tt_products.domain = www.mytypo3domain.de

# PIDs zu den Shopseiten
#plugin.tt_products.PIDlistDisplay = 315
#plugin.tt_products.PIDitemDisplay = 316
#plugin.tt_products.PIDsearch = 215

# Bildgrößen
plugin.tt_products.maxW_popup = 400
plugin.tt_products.maxH_list = 100
plugin.tt_products.maxH_single = 100

# Farben
plugin.tt_products.color3 =
plugin.tt_products.color2 =
plugin.tt_products.color1 =

plugin.tt_products.NoSingleViewOnList = 1

```

Der Folgende Code definiert das TS-Setup des Autorentemplate:

```

# Default PAGE object:
page = PAGE
page.typeNum = 0
page.stylesheet = fileadmin/osp/styles/standard.css

# Sortierung nach Titel
plugin.tt_products.conf.tt_products.ALL.orderBy = title

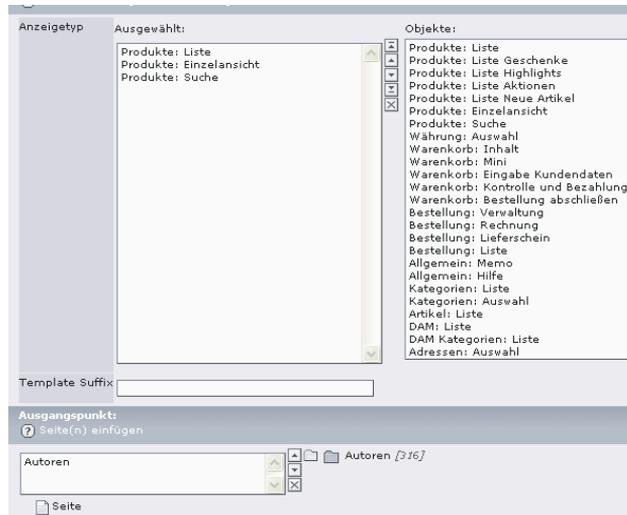
# Default CSS entfernen
plugin.tt_products._CSS_DEFAULT_STYLE =

```

Erstellen Sie jetzt einen SysOrdner und legen Sie darin Ihre Autoren als – im tt_products-Jargon – „Produkte“ an. Verwenden Sie als Titel den Namen des Autors, fügen Sie ein Bild des Autors ein und legen Sie beispielsweise seine Kurz-Vita in dem Textfeld ab.

Fügen Sie jetzt das Shop-Plugin in die Seite **Unsere Autoren** ein. Im Unterschied zum Shop und zum Katalog wählen Sie jetzt als Anzeigentyp unter Objekte **Produkte: Liste**, **Produkte: Einzelansicht** und evtl. **Produkte: Suche**. Binden Sie den Autoren-SysOrdner als Ausgangspunkt dieser Seite ein und speichern Sie den Datensatz ab (Abbildung 8.1).

Abbildung 8.1:
Plugins einfügen



Im Autorenverzeichnis soll die Listenansicht auf derselben Seite angezeigt werden wie die Einzelansicht. Oben die Liste, darunter die Einzelansicht und darunter die Suche.

Damit das auch gefällig aussieht, ist in der CSS-Datei für die Listenansicht eine Scrollbox mit Größenangabe gesetzt, in der die Autorennamen erscheinen.

Klicken Sie jetzt im Frontend einen der Autorennamen an, erhalten Sie detaillierte Informationen aus der Einzelansicht genau darunter, wie Abbildung 8.2 zeigt.

Abbildung 8.2:
Autorenverzeichnis



Möchte Sie weitere Details aus der Einzelansicht nutzen, z. B. das Feld `www` für die Homepage des Autors oder `related products`, um Koautoren anzuzeigen, fügen Sie diese Angaben hinzu. Sie können auch ein PDF einbinden, das zusätzliche Informationen enthält. Dieses Prinzip funktioniert allerdings nur solange, wie jeder Autor nur an einem Buch beteiligt ist. Bei mehreren Autoren können nur alle gemeinsam angezeigt werden.

Es ist also möglich, mehrere, voneinander unabhängige Shops innerhalb einer Rootseite zu betreiben, wobei Sie dasselbe oder auch unterschiedliche HTML-Templates einsetzen. Auf der Startseite des jeweiligen Shops muss ein leeres Extension-Template angelegt und über `Include basis template` in das Template der Rootseite eingebunden sein. Ebenso muss es ein weiteres Extension-Template im Template-SysOrdner geben, das die Shop-Konfiguration enthält. Dieses Template muss zudem das statische Template enthalten und wiederum über `Include basis template` in das leere Template eingebunden sein. Oder Sie arbeiten so, dass das statische Shop-Template ein einziges Mal auf der obersten Seite eingebunden wird, nicht mehrfach auf einzelnen Shopseiten. Das hat allerdings den gravierenden Nachteil, dass auch auf den Seiten, die keine Shop-Elemente enthalten, dieses umfassende Setup gilt, was für Seiten ohne Shop-Funktionalität nur unnötiger Ballast wäre. Versuchen Sie darum bei größeren Seiten, nur bei jenen Seiten ein Extension-Template einzubinden, wo dies auch wirklich notwendig ist.

Im Beispielshop wird mit verteilten CSS-Dateien gearbeitet, die über `Include url` integriert sind. Wenn Sie unterschiedliche CSS-Dateien für die verschiedenen Shops einrichten möchten, empfiehlt sich dieses Vorgehen unter Umständen nicht. Binden Sie in diesem Falle die CSS-Dateien im TS-Setup oder den Constants des jeweiligen Shop-Template direkt ein.

9

Kapitel

Alternative TYPO3-Shopsysteme

In diesem Kapitel wollen wir einige weitere Shop-Extensions für TYPO3 vorstellen. Diese haben wir installiert und Bestellmöglichkeiten implementiert, um deren Grundfunktionalitäten zu testen. Sie werden allerdings nicht so ausführlich beschrieben wie der Beispielshop auf Grundlage von `tt_products`. So haben wir uns mit den für die Lauffähigkeit notwendigen Konfigurationsschritten begnügt und keine Anpassungen an CSS- und HTML-Dateien vorgenommen. Selbstverständlich lassen sich auch diese Shops umfassend konfigurieren und anpassen.

Eine Besonderheit stellt das Shopsystem *GSA Shop* dar, dessen Installation hier simuliert wird, denn es ist lediglich ein Ausblick auf eine geplante TYPO3-stand-alone-Version.

9.1 Commerce

Am Beispiel eines Bekleidungsshops wird eine einfache Installation der von den Firmen Marketing Factory Consulting GmbH, e-netconsulting KG und n@work Internet Informationssysteme GmbH entwickelten Extension *Commerce*¹ beschrieben. Da sich die Artikelverwaltung grundlegend von *tt_products* unterscheidet und hier etwas ausführlicher beschrieben wird, eignet sich der Buchshop nicht so gut. Am Beispiel von Waren, wie z. B. einem T-Shirt, das es in unterschiedlichen Farben und Größen – also zahlreichen Varianten – geben kann, lässt sich die Artikelverwaltung wesentlich besser demonstrieren.

Commerce wird für große Shops mit einem umfangreichen Warenbestand entwickelt und bietet einige Module, die *tt_products* nicht enthält. Auf diese Module soll im weiteren Verlauf dieses Kapitels eingegangen werden.

9.1.1 Installation

Installieren Sie über den Extension Manager die Extension mit dem Key `commerce` und alle darüber angeforderten Extensions. Bevor Sie die dann notwendigen Einstellungen an Commerce vornehmen können, müssen zunächst einige Shopseiten eingerichtet und weitere Vorbereitungen getroffen werden.

Beachten Sie, dass die Graytree Library mindestens in der Version 0.1.1 installiert werden muss; zudem wird hier PHP in Version 5.1.2 und TYPO3 in Version 4.2-dev eingesetzt. Folgende Extensions müssen demnach in das System integriert werden:

- Commerce 0.9.3 (`commerce`)
- Graytree Library 0.1.1 (`graytree`)
- Address list 2.1.0 (`tt_address`)
- Dynamic Flexforms 1.11.0 (`dynaformflex`)
- Money code Library 1.2.1 (`moneylib`)
- Static Info Tables 2.0.5 (`static_info_tables`)
- Frontend User Registration 2.5.7 (`sr_feuser_register`)
- New Login Box 2.2.9 (`newloginbox`)

¹ http://www.marketing-factory.de/TYPO3_Online_Shop_Extension_tx_commerce_TYPO3_COMMERCE.404.0.html

9.1.2 TS-Template

In Ihrem Template-SysFolder erstellen Sie ein neues TS-Template für Commerce; das allgemeine Vorgehen am Beispiel `tt_products` beschreibt Abschnitt 1.3 ab Seite 23. Vergeben Sie den Namen `+commerce`, weisen Sie über `Include static (from extension)` das statische Template `COMMERCE` zu und speichern Sie den Datensatz ab. Binden Sie dieses Template wie gewohnt in das TS-Template Ihrer Rootseite ein.

9.1.3 Seiten und Plugins

Erstellen Sie jetzt die benötigten Seiten für Ihren Shop.

Listenansicht

Shop sei die Hauptseite Ihres Shops, auf der Ihre Produkte und Artikel gelistet werden; sie soll – im Gegensatz zu anderen Seiten, die später hinzukommen – im Menü sichtbar sein. Weisen Sie dieser Seite das Plugin zu: Wählen Sie dazu **Allgemeines Plugin** aus der Liste und im Formular **Commerce: Produkte Liste** (Abbildung 9.1). Die Plugin-Optionen der Flexforms lassen Sie zunächst unberücksichtigt, und auch den Ausgangspunkt können Sie ignorieren, denn dieser muss im Commerce-Shop nicht gesetzt werden. Speichern Sie anschließend den Datensatz ab.

The screenshot shows a configuration window for a plugin. At the top, it says 'Erweiterung:' followed by a question mark icon and the text 'Wählen Sie das Plugin aus, das Sie einfügen'. Below this is a dropdown menu with 'Commerce: Produkt Liste' selected. Underneath is a section titled 'Plugin Optionen:' with a question mark icon and the text 'FlexForm Konfiguration von Plugins.'. At the bottom, there are two tabs: 'Produktdaten' (which is highlighted with a yellow border) and 'Templatedaten'.

Abbildung 9.1:
Plugin Commerce:
Produkte Liste

Falls Sie eine Listenansicht wünschen, auf der nur ein bestimmtes Produkt mit den zugehörigen Artikeln (also zum Beispiel ein Hosen-Modell in unterschiedlichen Farben) angezeigt wird, wählen Sie später – denn Produkte und Artikel sind zum jetzigen Zeitpunkt noch nicht angelegt – im Feld **Detailprodukt (für Singleansicht)** das gewünschte Produkt und speichern Sie den Datensatz ab. Auf dieser Seite wird dann nur das ausgewählte Produkt mit allen zugehörigen Artikeln angezeigt.

Einzelansicht

Die Einzelansicht des Produkts erreichen Sie im Frontend per Klick auf den Produkttitel; es ist nicht notwendig, eine separate Seite für die Einzelansicht anzulegen, denn diese wird immer auf derselben Seite wie die Listenansicht angezeigt. Ein Text, den Sie dem Elternprodukt – nicht den einzelnen Artikeln – zuweisen, erscheint bei Klick auf den Titel des Elternelements (also des Produkts). Dies entspricht der Einzelansicht des Produkts.

Warenkorb

Erstellen Sie eine Seite für den Warenkorb und weisen Sie das Plugin (**Allgemeines Plugin**) zu. Im Formular wählen Sie aus der Liste der Erweiterungen **Commerce: Warenkorb** und speichern den Datensatz ab.

Abbildung 9.2:
Plugin Commerce:
Warenkorb

The screenshot shows a configuration form with two sections. The first section, titled 'Erweiterung:', contains a help icon and the text 'Wählen Sie das Plugin aus, das Sie einfügen wollen. (Verfüg'. Below this is a dropdown menu with 'Commerce: Warenkorb' selected. The second section, titled 'Layout:', contains a help icon and the text 'Wählen Sie ein Layout aus.'. Below this is a dropdown menu with 'Normal' selected.

Bestellungen

Für die eigentliche Bestellung wird eine weitere Warenkorbseite benötigt. Dazu wählen Sie **Allgemeines Plugin** und im Formular aus den Erweiterungen **Commerce: Checkout** aus. Speichern Sie anschließend den Datensatz ab.

Abbildung 9.3:
Plugin Commerce:
Checkout

The screenshot shows a configuration form with one section titled 'Erweiterung:'. It contains a help icon and the text 'Wählen Sie das Plugin aus, das Sie einfügen wolle'. Below this is a dropdown menu with 'Commerce: Checkout' selected.

Adressverwaltung

Erstellen Sie eine Seite, über die die Kundendaten verwaltet werden (**Allgemeines Plugin**, im Formular **Commerce: Adressverwaltung**).



Abbildung 9.4:
Plugin Commerce:
Adressverwaltung

Rechnung

Erstellen Sie eine Seite für die Rechnungen nach dem nun schon bekannten Muster: **Allgemeines Plugin | Commerce: Rechnung**. Auf dieser Seite kann der Kunde später seine Rechnungen einsehen.



Abbildung 9.5:
Plugin Commerce:
Rechnung

Seite und Content leerer Warenkorb leere Bestellung

Erstellen Sie je eine Textseite, die im Frontend erscheint, wenn der Warenkorb leer ist bzw. noch keine Bestellung vorliegt. Wählen Sie für diese Seiten entsprechenden Text (also beispielsweise „Der Warenkorb ist leer“ oder „Bitte wählen Sie einen Artikel“) und speichern Sie die Datensätze ab. Diese Seiten sollten im Menü versteckt sein.

SysOrdner

Erstellen Sie je einen SysOrdner für Benutzer und Kundenadressen. Die Adressen werden von der Extension `tt_address` im Ordner **Adressen** verwaltet. Dass verschiedene Ordner notwendig sind resultiert daraus, dass die von Commerce verwendete Frontendbenutzerregistrierung nur eine Adresse je Benutzer kennt, Commerce aber Rechnungs- und Lieferadressen unterscheidet. Pro Benutzer können somit also verschiedene Adressen verwaltet werden.

Die Benutzer werden entsprechend im SysOrdner **Benutzer** verwaltet; die Adressdaten müssen spätestens beim Auschecken des Warenkorbs angegeben werden. Sofern der Kunde die Bestellung danach abbricht, werden seine Daten nicht im System gespeichert. Kommt es zum Abschluss der Be-

stellung, werden in diesem Ordner Benutzerdaten samt Bestellungen verwaltet.

Mit der Bestellbestätigung werden dem Kunden ein Benutzername und ein Passwort zugesendet, die er für spätere Einkäufe im Shop nutzen kann.

9.1.4 PIDs zuweisen

Sind alle notwendigen Seiten für den Shop erstellt, wechseln Sie in den Extension Manager und weisen dort zumindest die PID der Seite **Rechnung (Page-ID for Invoice)** und der Listenansicht (**Preview Page Type**) zu. Klicken Sie anschließend auf **Update** und öffnen Sie dann das TS-Template (`+commerce`).

Über den Constant-Editor weisen Sie jetzt die weiteren IDs zu und nehmen ggf. weitere Einstellungen vor. Wählen Sie dazu die Kategorie **PLUGIN.COMMERCE: GENERAL** aus.

Setzen Sie je einen Haken bei allen Einträgen im Bereich **Links and Targets** und klicken Sie den **Update**-Button. Tragen Sie nun die PIDs der Seiten ein:

Address PID

SysOrdner der Kundenadressen

USER PID

SysOrdner der Frontendbenutzer

Edit Address PID

Seite Adressverwaltung

Basket PID

Warenkorbseite

Override PID

Listenansicht

Checkout PID

Seite Bestellungen

Empty basket PID

Textseite bei leerem Warenkorb

Empty Checkout PID

Textseite bei leerer Bestellung

Diese Einstellungen sollen zunächst genügen; für alle weiteren Konstanten werden die Voreinstellungen genutzt.

9.1.5 Webmodul Commerce

Jetzt – oder nachdem Sie den Cache gelöscht haben – erscheint im Webmodul die Rubrik **Commerce** mit den Bereichen **Kategorie**, **Bestellung**, **Stammdaten** und **Statistics**.



Abbildung 9.6:
Commerce im
Webmodul

Kategorien

Commerce verwaltet Produkte und die zugehörigen Artikel in sog. *Kategorien*. Die Darstellung des Kategoriebaums im Backend steuert die Extension Graytree Library. Es gibt Haupt- und Unterkategorien, innerhalb derer Produkte und Artikel verwaltet werden.

Klicken Sie im Webmodul **Commerce** auf **Kategorie**, um eine erste Kategorie anzulegen. In der Liste gibt es bereits einen Eintrag **System**; wählen Sie nun **Neuen Datensatz anlegen | Kategorie** und vergeben Sie im Backendformular einen Namen, im Beispiel **Kleidung**. Diese besitzt in der Beispielinstallation die UID 2, die bereits in den Constants voreingestellt ist. Soll diese Kategorie nicht Ihre Hauptkategorie sein, weisen Sie in den Constants die UID der gewünschten Hauptkategorie zu.

Erstellen Sie nun auf der darunter liegenden Ebene, also innerhalb von **Kleidung** eine weitere Kategorie **T-Shirts** und bei Bedarf auch darin eine Unterkategorie, z. B. **Kinder-T-Shirts**. Sie können jede Kategorie auch mit der linken Maustaste anklicken und über **neue Unterkategorie** weitere Unterpunkte einfügen.

Ein Verschieben von Produkten über Ausschneiden/Einfügen in andere Kategorien ist nicht möglich. Möchten Sie ein Produkt in eine andere Kategorie verschieben, rufen Sie das Produkt auf und weisen über das Formular eine andere Kategorie zu. Das entsprechende Formular und eine Ansicht des Kategoriebaums zeigt Abbildung 9.7.

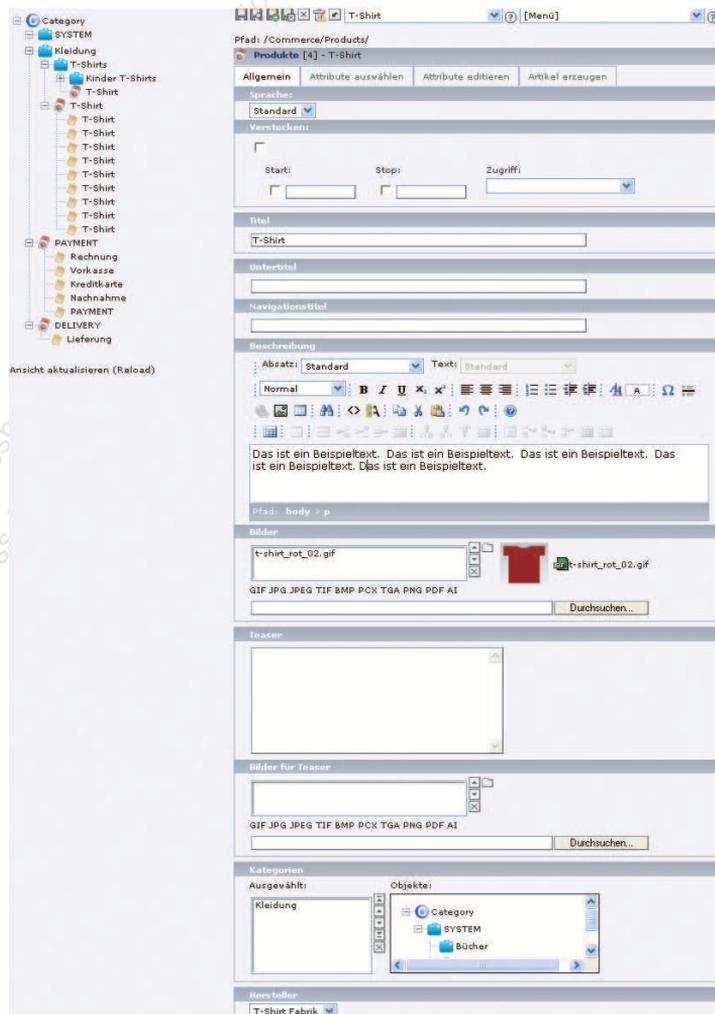
Produkte

Über das **Kategorie**-Modul legen Sie anschließend auch die einzelnen Produkte an, also beispielsweise in der Kategorie **T-Shirts** über **neuen Datensatz anlegen | Produkte**. Vergeben Sie einen Titel und einen kurzen Infotext,

fügen Sie ein Bild ein und speichern Sie den Datensatz ab. Alle weiteren Informationen wie Teasertext, Teaserbild (für die Vorschau) etc. können Sie über die Registerkarte **Allgemein** hinterlegen. Die übrigen Registerkarten bleiben zunächst unberücksichtigt. Leider fehlt auch bei Commerce die Möglichkeit, einen alternativen Text zum Bild zu speichern, was für einen barrierearmen Shop aber dringend notwendig wäre.

Im Seitenbaum der Kategorien finden Sie nun Ihr Produkt in der Kategorienhierarchie eingeordnet, hier also in der Kategorie Kleidung die Unterkategorie T-Shirts mit dem Produkt T-Shirt. Innerhalb der Unterkategorie liegt eine weitere Unterkategorie namens Kinder-T-Shirts.

Abbildung 9.7:
Produkt T-Shirt
(Commerce)



Produkte sind demnach Elternelemente aller spezifischen Varianten, also der Artikel. Ein Artikel definiert sich also beispielsweise innerhalb der Kategorie T-Shirts durch Stoffgewicht, Größe, Farbe etc. Doch bevor Sie diese Artikel anlegen, wenden Sie sich kurz den Themen Bezahlung und Lieferung zu, deren Methoden ebenfalls in Kategorien definiert werden.

Bezahlung und Lieferung

Im Kategoriebaum sind im Bereich **Payment** die möglichen Bezahlmethoden definiert. Bei Commerce ist das Backendformular für eine Bezahlmethode ähnlich dem für einen Artikel aufgebaut. Einige Methoden sind bereits voreingestellt, können aber verändert und erweitert werden. Wählen Sie z. B. die Bezahlmethode **cash on delivery** und ändern Sie den Titel in der Registerkarte **Allgemein** zu **Nachnahme**. Wechseln Sie in die Registerkarte **Preise** und tragen Sie dort den Preis für die Nachnahme ein.

Abbildung 9.8:
Zahlungsmethode
(Commerce)

Um eine neue Bezahlmethode hinzuzufügen, erstellen Sie einen weiteren „Artikel“: Dazu wählen Sie das „Produkt“ **PAYMENT** aus der Liste in Abbildung 9.9 aus und im daraufhin erscheinenden Formular die Registerkarte **Artikel** erzeugen. Dort wählen Sie wiederum die Registerkarte **erzeugbare Artikel** (Abbildung 9.10), setzen den Haken bei **Einen leeren Datensatz erzeugen** und speichern diesen ab. In Ihrem Kategoriebaum erscheint nun eine weitere Bezahlmethode; den entsprechenden Datensatz können Sie editieren und so die neue Bezahlmethode einrichten. Das gleiche Vorgehen gilt für die Liefermethoden.



Abbildung 9.9:
Payment und
Delivery (Commerce)

Abbildung 9.10:
Neue
Zahlungsmethode
hinzufügen
(Commerce)



Stammdaten

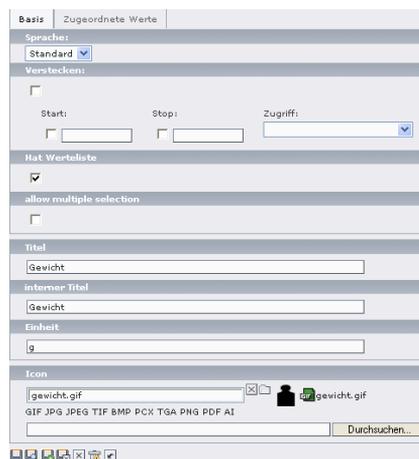
Verlassen Sie für einige weitere Vorarbeiten zunächst die Kategorien und wenden Sie sich den Stammdaten zu. Dort sollen die Produktinformationen vervollständigt werden. Wählen dazu im Webmodul Commerce den Punkt Stammdaten aus, der Zugriff auf weitere Unterpunkte erlaubt.

Ein wichtiges Merkmal von Commerce sind die Attribute.

Ein *Attribut* ist eine Produkteigenschaft („Größe“, „Farbe“, „Material“, ...), die mit einem passenden *Wert* („XL“, „rot“, „Baumwolle“, ...) verbunden wird. Aus den verschiedenen Attributen setzen sich dann die vom Kunden später bestellbaren Artikel zusammen, also z. B. ein rotes Baumwoll-T-Shirt in Größe XL. Um dies zu testen, müssen also zunächst einige Attribute definiert und mit zugehörigen Werten belegt werden.

Über **Neues Attribut** erstellen Sie naheliegenderweise ein Attribut (z. B. Gewicht) und vergeben einen Titel und eine Einheit (z. B. Gramm). Ein hier ergänztes Bild dient in der Übersicht dann als Symbol für das Gewicht.

Abbildung 9.11:
Attribut anlegen
(Commerce)



Setzen Sie den Haken bei Hat Werteliste, denn sobald Sie das Attribut speichern, können Sie diesem Werte zuweisen. Klicken Sie auf die Registerkarte Zugeordnete Werte und erstellen Sie entsprechenden Werte.

Abbildung 9.12:
Attribut-Werte
hinzufügen
(Commerce)

Im Beispiel wurden die Werte 100, 200 und 300 als Stoffgewichte/-qualitäten der Shirts zugewiesen. Im Backendformular legen Sie weiter fest, dass die Einheit Gramm in Ganzzahlen erscheinen soll. Anschließend legen Sie ein zweites Attribut Farben an und weisen mögliche Werte zu. Weitere Informationen zu den Attributen finden Sie ab Seite 198.

Abbildung 9.13:
Hersteller anlegen
(Commerce)

Bei den Produkten lassen sich darüber hinaus Angaben zum **Hersteller** hinterlegen, hier also beispielsweise T-Shirt Hersteller (Abbildung 9.13).

Analog ist das Vorgehen bei den **Lieferanten**; im Beispielshop wurde ein T-Shirt Lieferant angelegt.

9.1.6 Produkte und Artikel

Damit sind alle Vorbereitungen für die Erzeugung der einzelnen Artikel abgeschlossen, denn diese werden über die soeben erstellten Attribute und Adressen spezifiziert.

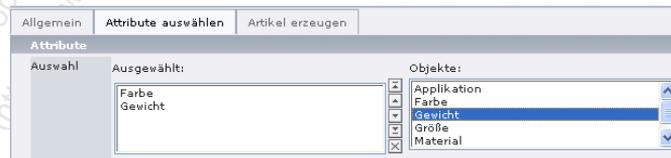
Wechseln Sie wieder zu den Kategorien. Der Kategoriebaum des Shops sieht wie folgt aus:

Abbildung 9.14:
Kategoriebaum
(Commerce)



Wählen Sie dort das Elternprodukt T-Shirt und darin die Registerkarte **Attribut auswählen**; im Feld **Auswahl** weisen Sie die Attributobjekte **Gewicht** und **Farbe** zu und speichern den Datensatz ab.

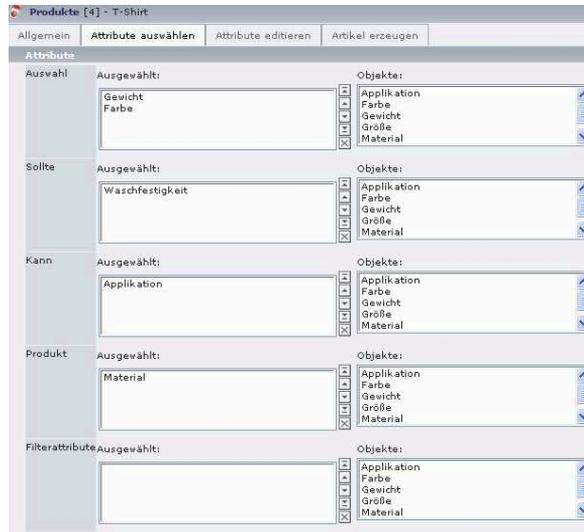
Abbildung 9.15:
Attribute-Auswahl
(Commerce)



9.1.7 Artikel erzeugen

Wechseln Sie anschließend auf die Registerkarte **Artikel erzeugen** und dort in die Sektion **erzeugbare Artikel**, die eine Liste mit Checkboxes öffnet. Soll keines der Attribute ausgewählt werden, können Sie **Einen leeren Datensatz** erzeugen, und zwar so viele, wie Sie Artikel zu diesem Produkt anbieten möchten. Diese Artikel füllen Sie anschließend mit Leben, indem Sie die Attribute in der Registerkarte **existierende Artikel** über eine Combobox zuweisen. Wählen Sie **Alle Artikel auswählen** (Abbildung 9.16), werden alle Artikel erstellt, die sich durch die Kombination der Attributwerte ergeben. Sind also T-Shirts in drei Farben und jeweils drei Qualitäten lieferbar, ergeben sich daraus automatisiert 9 verschiedene Artikel. Werden nicht alle Kombinationen ausgewählt, weil das T-Shirt z. B. in der Farbe Weiß nur mit

Abbildung 9.18:
Auswahl der
Attribute
(Commerce)



Wenn Sie es sich jedoch anders überlegen, können Sie auch zu einem späteren Zeitpunkt eine Werteliste erstellen. In diesem Fall werden die bis dahin manuell gepflegten Werte automatisch übernommen, sobald Sie den Haken **Werteliste** setzen. Natürlich können Sie auch weitere Werte ergänzen.

Abbildung 9.19:
Attribut
Waschfestigkeit
zuweisen
(Commerce)



In das Feld **Kann** setzen Sie alle die Attribute, die zutreffen *können*; beispielsweise könnte das T-Shirt eine Applikation auf der Vorder- oder Rückseite haben. Dieses Attribut hat eine Werteliste mit den Werten *vorne* und *hinten*, und wenn Sie im Artikel die Attribute editieren, lässt sich einer der beiden Werte zuweisen.

Die Unterscheidung in **Kann** und **Sollte** ist lediglich eine für den Redakteur: Das **Kann**-Feld kann befüllt werden, das **Sollte**-Feld muss befüllt werden. Allerdings handelt es sich dabei nicht um eine technische Unterscheidung. Sofern ein **Sollte**-Feld nicht gefüllt wurde, wird keinerlei Warnung o. Ä. ausgegeben.

The screenshot shows the 'Attribute editieren' form for a T-shirt product. The form has tabs for 'Allgemein', 'Attribute editieren', and 'Preise'. Under the 'Attribute editieren' tab, the following attributes are visible:

| | |
|-----------------|-----------------------------------|
| Gewicht | 100 |
| Farbe | weiss |
| Waschfestigkeit | waschbar 60 Grad, Waschfestigkeit |
| Applikation | vorne, Applikation |
| Material | Baumwolle |

Abbildung 9.20:
Attribut Applikation
zuweisen
(Commerce)

In das Feld **Produkt** setzen Sie alle Attribute mit globalen Werten, also solchen, die für alle Artikel des Produkts gelten. Ein Beispiel wäre hier **Material** für den Fall, dass alle Artikel aus dem gleichen Material hergestellt sind. Erstellen Sie ein Attribut, weisen Sie es über **Produkt** im Produktdatensatz zu und speichern Sie den Datensatz. Es erscheint eine weitere Registerkarte **Attribute editieren**, in der Sie das Attribut zuweisen. Es kann pro Produkt inklusive allen zugehörigen Artikeln nur ein globaler Wert ausgewählt werden. Im Frontend erscheint die Anzeige in der Einzelansicht des Produkts und wird nicht bei jedem Artikel angezeigt; in diesem Beispielshop sind also alle T-Shirts aus „100% Baumwolle“ gefertigt.

The screenshot shows the 'Attribute editieren' form for a T-shirt product. The form has tabs for 'Allgemein', 'Attribute auswählen', 'Attribute editieren', and 'Artikel erzeugen'. Under the 'Attribute editieren' tab, the following attribute is visible:

| | |
|----------|---------------------|
| Material | Baumwolle, Material |
|----------|---------------------|

Abbildung 9.21:
Attribut Material im
Produkt zuweisen
(Commerce)

Im Feld **Filterattribute** sollen jene Attribute, die später für die Suche oder sonstige Filter genutzt werden, eingetragen werden. Da diese Filterattribute allerdings sehr komplex sind und für jeden Shop individuell zugewiesen werden müssen, muss jeder Shopbetreiber diese Filterattribute selbst implementieren. Komplexe UND- bzw. ODER-Verknüpfungen müssen dem Artikelstamm zugewiesen werden und sind für jeden Shop andere. In diesem Beispielshop wurden keine Filterattribute verwendet.

Im Artikel tragen Sie zudem den jeweiligen Steuersatz ein. Eine globale Zuweisung des Steuersatzes sieht Commerce nicht vor.² Ist der Hersteller für alle Artikel des Produkts derselbe, wird er über das Produkt (nicht den Artikel) zugewiesen. Der Lieferant kann abhängig vom Artikel oder von der Liefermethode sein und wird entsprechend über eine der beiden Kategorien definiert.

In der Registerkarte **Preise** eines Artikels finden Sie umfangreiche Eingabemöglichkeiten. Vergeben Sie zunächst einen Preis von EUR 10,- und setzen Sie bei **Staffelpreismenge von** und **Staffelpreismenge bis** jeweils eine 1 (Abbildung 9.22).

² Die Zusatz-Extension `commerce_germantax` ermöglicht diese globale Zuweisung.

Abbildung 9.22:
erster Artikelpreis
(Commerce)

Tragen Sie hingegen bei **Staffelpreismenge bis** eine 10 ein, gilt der gesetzte Preis bis zur Bestellung von 10 Stück des jeweiligen Artikels.

Möchten Sie ergänzend für die Staffelmenge 11–20 Stück einen anderen Preis definieren, setzen Sie zunächst den Haken bei **Neuen Preis hinzufügen** (Abbildung 9.23) und speichern den Datensatz ab.

Abbildung 9.23:
Neuen Preis
hinzufügen
(Commerce)



Erst dann lässt sich ein weiterer Staffelpreis von z. B. EUR 8,- mit den Werten 11 bzw. 20 definieren (Abbildung 9.24).

Staffelpreise lassen sich über Start- und Stoppdatum auch für einen bestimmten Zeitraum definieren; auch gruppenspezifische Preise sind über das Feld **Zugriff** möglich. Sofern Sie eine Staffelung je nach Benutzergruppe anbieten möchten, erstellen Sie für jede dieser Gruppen einen neuen Preis bzw. eine neue Preisstaffel. Beachten Sie aber, dass dies auch schnell unübersichtlich und damit fehleranfällig wird. Ist ein Benutzer mehreren Benutzergruppen zugeordnet – was eigentlich zu vermeiden ist –, erhält er den jeweils günstigsten Preis.

| | |
|-----------------------|---|
| Verstecken: | <input type="checkbox"/> |
| Start: | <input type="text"/> |
| Stop: | <input type="text"/> |
| Zugriff: | <input type="text" value="v"/> |
| Einkaufspreis | <input type="text"/> |
| Staffelpreismenge von | <input type="text" value="1"/> |
| Staffelpreismenge bis | <input type="text" value="10"/> |
| | <input type="button" value="Preis löschen"/> |
| Bruttopreis | <input type="text" value="8.00"/> |
| Nettopreis | <input type="text" value="8.00"/> |
| Verstecken: | <input type="checkbox"/> |
| Start: | <input type="text"/> |
| Stop: | <input type="text"/> |
| Zugriff: | <input type="text" value="v"/> |
| Einkaufspreis | <input type="text"/> |
| Staffelpreismenge von | <input type="text" value="11"/> |
| Staffelpreismenge bis | <input type="text" value="20"/> |
| | <input checked="" type="checkbox"/> Neuen Preis hinzufügen |

Abbildung 9.24:
Preisstaffel von
11-20 Artikeln
(Commerce)

9.1.8 Warenkorb

Die ersten Testbestellungen führen Sie aus der Listenansicht durch, indem Sie – wie sollte es anders sein – Artikel in den Warenkorb legen. Die daraufhin erscheinende Warenkorbansicht gibt einen Überblick mit Preisangaben unten rechts, und zwar per Default in folgender Reihenfolge: Nettogesamtwarenpreis, Bruttogesamtwarenpreis, Nettoversandkosten, Bruttoversandkosten, Nettoabwicklungsgebühren und Bruttoabwicklungsgebühren (sofern Sie diese zusätzlichen Kosten in der jeweiligen Bezahlmethode definiert haben).

9.1.9 Bestellungen

Wechseln Sie nun in der Rubrik **Commerce** auf **Bestellungen**, wo unter **Incoming** die Testbestellung, d. h. alle vom Shop-Administrator noch nicht bearbeiteten Bestellungen gespeichert sind. Per Klick auf das Icon vor dem Datensatz erhalten Sie eine Ansicht der zugehörigen Kundendaten.

Abbildung 9.25:
Bestellungen im
Ordner Incoming
(Commerce)

| Bestell-ID | Bestelldatum | Organisation | Nachname | Name | Adresse | PLZ | Stadt | E-Mail | Telefon | Summe Bruttopreis | [Erstanz] | [Payments] | [Lieferer] | [Sonstisches] | [Artikel] |
|-------------------------|--------------|--------------|----------|--------|----------------|----------|----------|-------------------------|---------|-------------------|-----------|------------|------------|---------------|-----------|
| 470e481e62c3b0.46772475 | 11-10-07 | testco | Teststr. | 11111 | Teststr. 11111 | Teststr. | Teststr. | testco@mytypo3domain.de | € 2,00 | 13-10-07 | 1 | | | | |
| 46f9ba1647c22.95242627 | 27-09-07 | gfgfgf | gfgfgf | gfgfgf | gfgfgf | gfgfgf | gfgfgf | testco@mytypo3domain.de | € 2,00 | 13-10-07 | 1 | | | | |

Klicken Sie die E-Mail-Adresse des Kunden an, wird die Ansicht gesplittet in Kunden- und Bestelldaten. Über die obere rechte Combobox wechseln Sie zwischen beiden Angaben.

Um nun diese Bestellung in einen anderen Ordner zu verschieben, beispielsweise aus **Incoming** in **Working**, müssen Sie entweder am Ende der Bestellung einen Haken setzen und in der Combobox den entsprechenden Ordner auswählen oder die Bestellung durch Klick (rechte Maustaste, **Bearbeiten**) auf das Icon vor der Bestellung auswählen. Damit können Sie den jeweiligen Zustand der Order verfolgen (Registerkarten **Kundendaten**, **Lieferadresse** und **Artikel**), bis die Ware ausgeliefert und der Vorgang abgeschlossen ist.

Abbildung 9.26:
Bestellungen in
anderen Ordner
verschieben
(Commerce)

Gelöschte und stornierte Aufträge sollten sofort in einen Storno-Ordner wandern, den Sie von der Aggregation ausnehmen (vgl. Seite 206). Andernfalls geraten die Beträge in Ihre „echten“ Berechnungen und verfälschen die Statistik. Diesen Ordner erstellen Sie als SysOrdner im Bereich der „Orders“; im Feld **Enthält Erweiterung** weisen Sie **Commerce Systemordner** zu (Abbildung 9.27). Im Webmodul **Commerce | Bestellungen** können Sie nun Ihre stornierten Bestellungen über die Combobox in diesen Ordner verschieben.

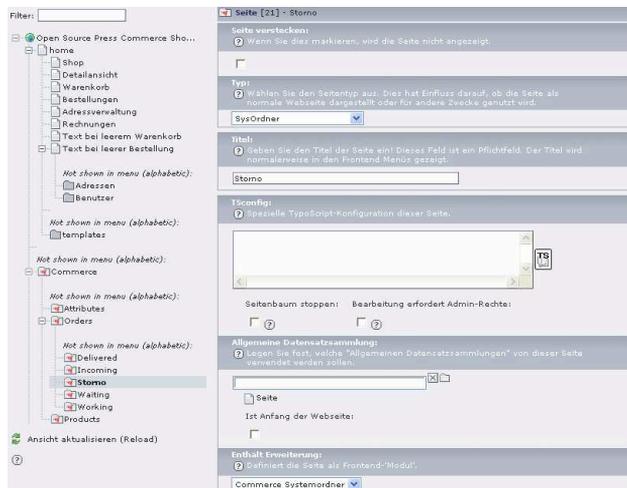


Abbildung 9.27:
Storno-Ordner
anlegen (Commerce)

Möchten Sie alle Aufträge eines Ordners exportieren, und zwar mit allen zur Verfügung stehenden Feldern, exportieren Sie die Datei über das CSV-Symbol und fügen es ins gewünschte Programm wieder ein. Sie können für diese Tabelle auch ein kleines PHP-Skript schreiben und die Daten direkt in Ihre Faktura übernehmen.

Move Order Mail

Ein weiteres interessantes Feature von Commerce sind die so genannten *Move Order Mails*, automatisierte E-Mails, die Sie sowohl intern, als auch an Ihre Kunden versenden können, sobald sich der Status einer Bestellung geändert hat und die Bestellung in einen anderen Ordner gewandert ist. So können Sie den Kunden informieren, sobald z. B. die Zahlung eingegangen oder die Ware versendet ist. Aber auch die Mitarbeiter z. B. in der Versandabteilung erhalten einen Hinweis, dass die Ware für den Versand vorbereitet werden soll.

Wechseln Sie im Webmodul in die Listenansicht der Bestellungen, klicken Sie den Ordner Delivery an, für den Sie diese Änderungsnachricht anlegen möchten, und legen einen neuen Datensatz über Move Order Mail an.



Abbildung 9.28:
Move Order Mail
(Commerce)

Vergeben Sie einen Titel für diese Mail und definieren Sie zunächst, wann sie versendet werden soll (sobald sie in diesen Ordner verschoben wird oder sobald sie ihn verlässt). Wählen Sie also beispielsweise **Mail when move in** und weisen Sie ein HTML-Template für diese Nachricht über **Reference to Mailtemplate** zu. Für den ersten Test wurde das Template `template_usermail.html` verwendet (Abbildung 9.29). Da jedoch die Marker über den Inhalt der Bestellung nicht ersetzt werden, sollten Sie dieses Template unter anderem Namen abspeichern und entsprechend anpassen, also nicht verwendete Marker entfernen.

Sofern Sie keine weiteren Einstellungen vornehmen, geht diese Nachricht an den Kunden, sobald dessen Bestellung den Status **Delivered** annimmt.

Über das **BCC**-Feld lassen Sie die Nachricht weiteren Empfängern zukommen, also z. B. eine an den Kunden, eine zweite an den Shop-Administrator und eine dritte an den jeweiligen Disponenten. Für jeden Empfänger bzw. für jede Move Order Mail können Sie ein eigenes Template mit individuellem Text versenden.

Abbildung 9.29:
Formular für Move
Order Mail
(Commerce)

The screenshot shows a configuration form for a Move Order Mail. The form is organized into several sections:

- Sprache:** A dropdown menu set to "Standard".
- Verstecken:** A checkbox that is currently unchecked.
- Start:** A date input field.
- Stop:** A date input field.
- Zugriff:** A dropdown menu.
- Name of Template:** A text input field containing "Delivery Mail".
- When should mail be send:** A dropdown menu set to "Mail when move in".
- Reference to Mailtemplate:** A text input field containing "template_usermail.html" with a file selection icon.
- * -PHP -PHP3:** A text input field with a "Durchsuchen..." button.
- E-Mail Sendername (if not set: Mailaddress):** An empty text input field.
- E-Mailaddress of Sender (if not set: Moduledefault):** An empty text input field.
- Anderer Mailempfänger als Kunde:** An empty text input field.
- BCC Mail an:** An empty text input field.

9.1.10 Statistiken

Wechseln Sie nun im Webmodul **Commerce** in den Bereich **Statistics**, um Statistiken zu Ihren Bestellungen einzusehen. Wichtig ist in diesem Zusammenhang der Begriff *Aggregation*, also die Summe von Werten in einer

Tabelle, in diesem Fall also die Summe der Bestellungen. Die Statistiken werden in so genannten Aggregationstabellen gespeichert; Voraussetzung dafür ist, dass Sie zunächst die **Komplette Aggregation** einmal auslösen.



Abbildung 9.30:
Statistics: Komplette
Aggregation
(Commerce)

Daraufhin werden alle Aufträge bzw. Bestellungen erfasst und ausgewertet. Im Folgenden ist dann nur noch die **Zuwachsaggregation** zu aktivieren, um Veränderungen am Datenbestand zu erfassen, was deutlich weniger Zeit in Anspruch nimmt als eine vollständige Auswertung.



Abbildung 9.31:
Statistics:
Zuwachsaggregation
(Commerce)

Mit der Auswahl **Statistiken** in der Combobox erhalten Sie eine Übersicht, in der die Aufträge des aktuellen Tages allerdings nicht berücksichtigt sind. Auch monats- bis stundengenaue Ansichten sind möglich.

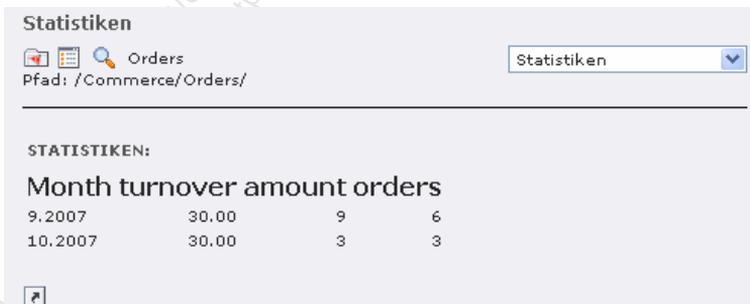
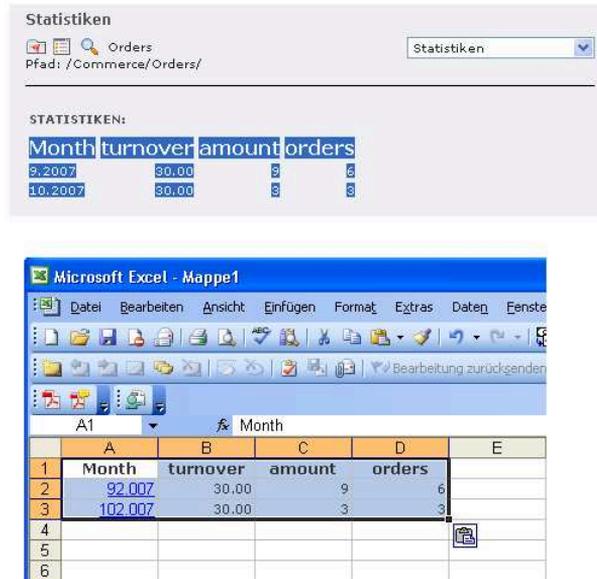


Abbildung 9.32:
Monatliche Statistik
(Commerce)

Wenn Sie eine Statistik drucken oder weiter aufbereiten möchten, kopieren Sie einfach die gesamte Tabelle und fügen sie in eine Tabellenkalkulation ein (Abbildung 9.33).

Abbildung 9.33:
Statistik kopieren
und in Excel
einfügen (Commerce)



Sie können bestimmte Ordner – beispielsweise den Storno-Ordner – von der Aggregation ausnehmen, indem Sie die PID des Ordners in der Konfiguration der Extension eintragen. Wechseln Sie dazu in den Extension Manager, wählen Sie Commerce durch Klick auf den Namen aus und nehmen Sie den Eintrag unter Exclude Folder from Statistic vor.

9.1.11 HTML-Templates

Wenn Sie anstelle der Default-Einstellungen ein eigenes HTML-Shop-Template nutzen, finden Sie dieses im Verzeichnis `typo3conf/ext/commerce/pi1` bis `./pi6`. Diese Dateien können Sie nicht wie üblich über den Extension Manager in Ihr Template-Verzeichnis im Fileadmin kopieren. Sie lassen sich aber per FTP vom Webserver (direkt verschieben oder) herunterladen und anschließend ins Template-Verzeichnis übertragen. Um sie zu bearbeiten, müssen eventuell (das hängt von der TYPO3-Konfiguration ab) die Dateierweiterungen geändert werden, und zwar vom Default `.tpl` zu `.tmpl` oder `.html`, oder Sie fügen `.tpl` als erlaubte Endung hinzu (s. Install Tool `[textfile_ext]`). Diese Endung ist eine Altlast in Commerce und soll in den nächsten Versionen umgestellt werden.

Constants

Im Constant Editor finden Sie folgende Einstellungen vor – im Testshop sind die bereits am Anfang des Kapitels zugewiesenen PIDs sowie eines der Templates zugewiesen:

```
plugin.tx_commerce_lib.addressPid = 18
plugin.tx_commerce_lib.userPid = 17
plugin.tx_commerce_lib.basketPid = 11
plugin.tx_commerce_lib.checkoutPid = 12
plugin.tx_commerce_lib.emptyBasketPid = 15
plugin.tx_commerce_lib.emptyCheckoutPid = 16
plugin.tx_commerce_lib.editAddressPid = 13
plugin.tx_commerce_lib.overridePid = 2
plugin.tx_commerce_pi1.templateFile =
    fileadmin/templates/commerce/category_product.html
```

Mit dieser Konfiguration bietet Commerce bereits einen lauffähigen Shop. Weitere Informationen zur Konfiguration von Commerce finden Sie in Foren und Mailinglisten. Eine kurze Installationsanleitung ist der Extension beigelegt. Für den Beispielshop schließen wir die Implementierung dieses sehr umfangreichen Shopsystems an dieser Stelle ab.

Bitte beachten Sie, dass die hier beschriebene Basisinstallation längst nicht alles ist, was Commerce zu leisten vermag. Dieses mächtige Shopsystem, das auf die Verwaltung eines großen Warenbestandes ausgelegt ist, wurde in diesem Kapitel nur beispielhaft installiert. Der Beschreibung des gesamten Funktionsumfangs würde den Rahmen dieses Buches sprengen.

9.2 GSA Shop

Die Extension von Rainer Kuhn basiert auf dem Datenmodell des Warenwirtschaftssystems GS-Auftrag der Sage Software GmbH & Co. KG³ und setzt PHP Version 5.1 oder höher voraus. Die Entwicklung dieser und aller anderen in diesem Kapitel genannten pt_*-Extensions wurde von der punkt.de GmbH⁴ gesponsort. Da GSA Shop derzeit noch an das Warenwirtschaftssystem gekoppelt ist und nicht ohne entsprechendes Hintergrundwissen installiert werden kann, liegt die Extension im TER als Alpha-Version vor. Sie wird aber bereits jetzt auf verschiedenen Websites im Produktionsbetrieb eingesetzt.⁵ In den nächsten Monaten soll eine neue Version des Shops erscheinen, die dann ohne das Warenwirtschaftssystem als TYPO3-Stand-alone-Version betrieben und auf Grundlage des Manuals von Administratoren installiert werden kann. Zu diesem Zeitpunkt müssen wir uns

³ <http://www.sage.de/>

⁴ <http://punkt.de>

⁵ Z. B. <http://www.klickbilderbox.de>

mit einem Ausblick auf dieses sehr umfangreiche Shop-System begnügen und schauen uns eine Testinstallation in Back- und Frontend an.

Doch zunächst ein kurzer Ausblick auf die Shoplösung: Mit dem Shop bereits realisierte Installationen beinhalten eine komplette und automatisierte Auftragsabwicklung, von der Bestellung bis zur Auslieferung und Fakturierung. Die hierfür notwendigen zusätzlichen Extensions werden derzeit noch vom Entwicklerteam überarbeitet und mittelfristig im TER zur Verfügung gestellt.

- Die Preiskalkulation rechnet mit bis zu vier Nachkommastellen. Deshalb ist es notwendig, dass PHP mit `-enable-bcmath` konfiguriert wird.⁶
- Es stehen je nach Artikel verschiedene Steuersätze zur Verfügung.
- Bei der Preisgestaltung können Angebote, Rabatte oder Staffellungen je nach Benutzergruppe sowie kundenspezifische Sonderpreise (individuelle Artikelpreise pro Kunde) eingesetzt werden. Preise können außerdem abhängig von der Bestellmenge definiert werden. Darüber hinaus ist es möglich, für einzelne Artikel Fixkosten zu definieren, die einmalig pro Bestellung und unabhängig von der bestellten Anzahl berechnet werden.
- Der Shop unterscheidet nach Händlern und Endkunden und bietet unterschiedliche Rechnungs- und Lieferanschriften an.
- Die Kreditkartenabrechnung ist derzeit über die Extension `pt_heidelpay` an den Anbieter Heidelbergpay gebunden; künftig sollen weitere Kreditkartenzahlungssysteme zum Einsatz kommen.
- Optional können sogenannte Artikel-Relationen aktiviert werden. Damit ist es beispielsweise möglich, beim Einkaufsvorgang notwendige Abhängigkeiten zu anderen Artikeln oder sich gegenseitig ausschließende Artikel zu definieren oder dem Kunden Bundle-Angebote vorzuschlagen.
- Eine Workflow-Engine mit unterschiedlichen Sichten für Kunden und Shopadministrator ist ebenfalls integriert. Individuelle Workflow-Abläufe können mittels Backend-Konfiguration und/oder durch Programmierung eigener Workflow-Klassen erstellt werden.

Es gibt für dieses Shopsystem vielfältige Möglichkeiten, die Extension an eigene Bedürfnisse anzupassen: So kann über die anpassbaren HTML-Templates für alle Plugins und die überschreibbaren CSS-Klassen das Layout des Shops individuell angepasst werden. Mit den Sprachlabels (Localisation via `locallang`) können Texte geändert werden. Der Shop ist mehrsprachig einsetzbar, vorausgesetzt ist natürlich die grundsätzliche Mehrsprachigkeit

⁶ Siehe <http://de.php.net/de/bc>

des Internetauftritts. GSA Shop bietet eine hohe Konfigurierbarkeit über TYPO3 im TS-Template und viele Hooks und APIs für Entwickler, wodurch eine Vielzahl von Anpassungs- und Erweiterungsmöglichkeiten für eigene Anforderungen zur Verfügung gestellt werden.

Wenden Sie sich nun dem Beispielshop zu.

9.2.1 Registrierung

Der Beispielshop sieht vor, dass sich ein potentieller Kunde spätestens nach dem Auschecken des Warenkorbs registriert und im Shop anmeldet. Für die Registrierung wird die Extension `pt_gsauserreg` von Wolfgang Zenker verwendet, die Sie in Abbildung 9.34 sehen. Für das Login kann ein beliebiges TYPO3-Login eingesetzt werden. Mit der Extension `pt_gsauserreg` ist es möglich, mehrere TYPO3-Frontendbenutzer einem Kunden zuzuordnen. Das ermöglicht z. B. eine Prüfung der Bestellung des einzelnen Kunden-Benutzers, wenn über den Workflow (vgl. S. 221) eingerichtet ist, dass die Bestellung dieses Benutzers durch seinen Chef-Einkäufer zunächst freigegeben werden muss, bevor der Bestellvorgang als abgeschlossen gilt. Die Extension ist derzeit noch nicht im TER verfügbar, wird Interessenten jedoch auf Anfrage von punkt.de zur Verfügung gestellt.

Kunde / Ansprechpartner

Firma: ?

EU Ustid: ?

Anrede:

Titel:

Vorname: *

Nachname: *

Geburtsdatum:

Anschrift

Hausanschrift

Strasse / Hausnr.: *

Adresszusatz:

PLZ: *

Ort: *

Bundesstaat:

Land: *

Postfach

Postfach Nr.:

PLZ:

Telekontakte

Telefon:

Alternative Nr.:

Telefax:

2. Telefax:

Handy:

Alternative Handy-Nr.:

Email Adresse: *

Alternative Email:

Homepage:

* : Markierung für Pflichtfeld

Abbildung 9.34:
Registrierung im
Frontend (GSA Shop)

Je nach Konfiguration der Extension können bei der Registrierung auch die Kontodaten des Benutzers erfragt und überprüft werden (Abbildung 9.35). Zur Überprüfung der Bankdaten wird das externe Tool `ktoblzcheck`⁷ eingesetzt, das auf dem Server installiert sein muss. Sollten Sie das Tool oder die Überprüfung nicht einsetzen wollen, können Sie dies bei der Installation der Extension `pt_gsauerreg` festlegen.

Abbildung 9.35:
Bankdaten eingeben
(GSA Shop)

Auswahl der Zahlungsart

Ihre Zahlungsweise: *

Abbuchung
 Kreditkarte
 Rechnung

Kontodaten

Kontoinhaber:

BLZ:

Bank:

Kontonr.:

BIC:

IBAN:

* : Markierung für Pflichtfeld

9.2.2 Installation

Leider ist es momentan noch nicht so einfach, den GSA Shop zu installieren. Simulieren wir also die Installation.

Installieren Sie GSA-Shop und alle benötigten Extensions. Zur Drucklegung des Buches wurden folgende Versionen eingesetzt:

- `pt_gshop` 0.9.5 (Shop-Core, bereits jetzt im TER)
- `pt_tools` 0.2.0 (allgemeine punkt.de-Bibliothek, bereits jetzt im TER)
- `smarty` 1.0.2 (externe Templating-Engine, im TER verfügbar)
- `static_info_tables` 1.3.0 (externe Bibliothek, im TER verfügbar)
- `pt_gsasocket` 0.1.7 (Anbindung an die GS-Auftrag-Datenbasis)
- `pt_gsauerreg` 0.0.10 (GS-Auftrag-basierte Benutzerregistrierung und -verwaltung)
- `pt_gsauerauth` 0.0.2 (GS-Auftrag-basierter Authentifizierungsservice)
- `pt_heidelpay` 0.1.2 (Kreditkartenzahlung über den Anbieter „Heidelpay“)

⁷ <http://ktoblzcheck.sourceforge.net/>

- pt_euvatcheck 0.0.8 (Prüfung der Umsatzsteuer-Identifikations-Nr. bei der Benutzerregistrierung)
- pt_gsaadmin 0.0.1 (optional: Datenpflege im Backend, wenn GS-Auftrag nicht eingesetzt werden soll)
- newloginbox 2.2.9 (optional: auch jedes TYPO3-Login möglich)

GSA Shop greift momentan direkt auf die Datenbank des Warenwirtschafts-systems zu (Artikelstamm, Kundendaten, Versandkosten, Steuersätze, Vorgangsverwaltung). Dazu muss der Administrator bei der Installation der Extension pt_gsocket die Verbindungsdaten zur GS-Auftrag-Datenbank im TYPO3 Extension Manager festlegen.⁸

Sobald alle notwendigen Daten aus der Warenwirtschaft zur Verfügung stehen, können Sie ein erstes Shop-Plugin in Ihre Shopseite einfügen.

9.2.3 Shopseiten und Plugins

Wie Sie in Abbildung 9.36 sehen, gibt es eine ganze Menge Plugins für den Shop.



Abbildung 9.36:
Liste der Plugins
(GSA Shop)

⁸ Unterstützung bei dieser Integration finden Interessenten bei <http://punkt.de>.

Artikel-Anzeige

Erstellen Sie zunächst eine Seite für die Artikel-Ansicht und wählen Sie das Plugin **GSA Shop: Artikel-Anzeige** aus. Da die Plugins von GSA-Auftrag sich nicht auf eine spezielle Shopseite beschränken, sondern artikelspezifisch sind, kann das Plugin für die Artikel-Anzeige auf jeder beliebigen Seite des Internetauftritts eingefügt werden. Eine automatische Listenansicht sieht GSA Shop derzeit nicht vor, ist aber für künftige Versionen geplant.

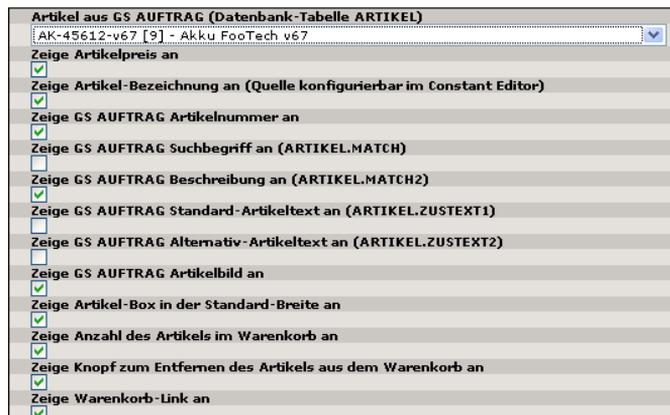
Mit diesem Plugin wird Ihnen die Liste aller im Warenwirtschaftssystem als Online-Artikel markierten Artikel, also der Shop-Warenbestand, angezeigt (Abbildung 9.37⁹). Aus dieser Liste wählen die den Artikel aus, den Sie auf dieser Shopseite darstellen möchten. Je Artikel fügen Sie ein solches Plugin vom Typ **GSA Shop: Artikel-Anzeige** ein.

Abbildung 9.37:
Plugin
Artikel-Anzeige
einfügen (GSA Shop)



Im Beispiel ist der Artikel „Akku“ ausgewählt. Wie Sie in Abbildung 9.38 sehen, können Sie hier unter verschiedenen Optionen für den jeweiligen Artikel auswählen:

Abbildung 9.38:
Artikel-Anzeige
Optionen (GSA Shop)



⁹ Aus Gründen der Vertraulichkeit sind die „echten“ Artikel in dieser Liste ausgegraut.

Zeige Artikelpreis an

Zeige Artikel-Bezeichnung an (Quelle konfigurierbar im Constant Editor)

Im Constant Editor kann der Quellort für die Artikelbezeichnung aus einer Liste ausgewählt werden (Abbildung 9.39). Beachten Sie, dass im Constant Editor verschiedene Kategorien ausgewählt werden können. Die Artikelbezeichnung entstammt der Kategorie GSA Settings.



Abbildung 9.39:
Constant Editor
Quellort
konfigurieren (GSA
Shop)

Zeige GS AUFTRAG Artikelnummer an

Zeige GS AUFTRAG Suchbegriff an (ARTIKEL.MATCH)

Zeige GS AUFTRAG Beschreibung an (ARTIKEL.MATCH2)

Zeige GS AUFTRAG Standard-Artikeltext an (ARTIKEL.ZUSTEXT1)

Zeige GS AUFTRAG Alternativ-Artikeltext an (ARTIKEL.ZUSTEXT2)

Zeige GS AUFTRAG Artikelbild an

Zeige Artikel-Box in der Standard-Breite an

Zeige Anzahl des Artikels im Warenkorb an

Wenn dieser Punkt aktiviert ist, erscheint hinter dem Bestell-Button die Anzahl der derzeit im Warenkorb befindlichen Artikel.

Zeige Knopf zum Entfernen des Artikels aus dem Warenkorb an

Hier wird hinter dem Bestell-Button ein Icon zum Löschen des Artikels aus dem Warenkorb angezeigt – sofern sich dieser Artikel im Warenkorb befindet.

Zeige Warenkorb-Link an

Wird dieser Haken gesetzt, wird ein Link zur Warenkorbseite direkt unterhalb des Artikels angezeigt.

Im Frontend sehen Sie nach dem Einfügen dieses Plugins den Artikel wie in Abbildung 9.40 dargestellt.

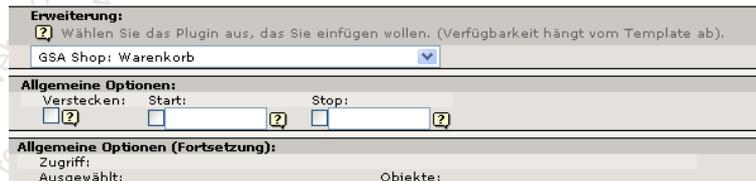
Abbildung 9.40:
Artikel-Anzeige im
Frontend (GSA Shop)



Warenkorb

Erstellen Sie eine Seite für die Warenkorbansicht und fügen das Plugin GSA Shop: Warenkorb ein. Wenn Sie den Haken bei Zeige Warenkorb-Link an gesetzt haben, kann diese Seite im Menü versteckt sein. Wie Sie in Abbildung 9.40 sehen, wird der Link zum Warenkorb damit unterhalb des einzelnen Artikels angezeigt.

Abbildung 9.41:
Plugin Warenkorb
einfügen (GSA Shop)



Im Frontend stellt sich der Warenkorb des Testshops wie folgt dar:

Abbildung 9.42:
Warenkorb im
Frontend (GSA Shop)

[Home / Warenkorb](#)
Shopping cart

| Anzahl | Artikel-Bezeichnung | Art.-Nr. | Preis* [EUR] | Summe* [EUR] |
|--|---|--------------|--------------------------|------------------|
| <input type="text" value="3"/> | Akku FooTech v67 <input type="button" value="Löschen"/> | AK-45612-v67 | 34.99 | 104.97 |
| <input type="text" value="1"/> | Hardware BarTech <input type="button" value="Löschen"/> | HW-3490762-A | 464.00 | 464.00 |
| <input type="button" value="Aktualisieren"/> <input type="button" value="Warenkorb leeren"/> | | | Gesamtsumme [EUR] | 568.97 |
| * Alle Preise sind Brutto-Preise (inkl. MwSt.) | | | | ZUR KASSE |

[Zurück zur letzten Bestell-Seite](#)

Über den Button **Aktualisieren** können Sie die Anzahl der im Warenkorb befindlichen Artikel ändern.

Artikel können übrigens über eine spezielle Konfiguration beim Anlegen des Artikels im Warenwirtschaftssystem (Feld **Webadresse** auf -1 setzen) automatisch im Warenkorb mit der Artikel-Einzelansicht verlinkt werden (S. 219), der Link führt zur jeweiligen Artikel-Einzelansicht (Abbildung 9.43).

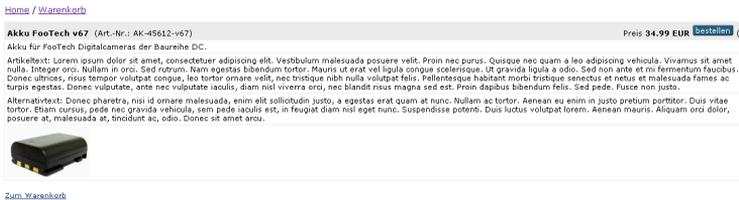


Abbildung 9.43:
Artikel-Einzelansicht
im Frontend (GSA
Shop)

Bestellung

Erstellen Sie nun auf der gleichen Ebene wie den bereits erstellten Shopseiten eine weitere Seite für die Bestellungen-Übersicht. Diese Seite soll im Menü versteckt sein. Fügen Sie dort das Plugin **GSA Shop: Bestellung** ein und wählen Sie bei den TYPO3-Anzeigeoptionen **Bei jedem Login anzeigen**, um diese Seite nur angemeldeten Benutzern zugänglich zu machen.



Abbildung 9.44:
Plugin Bestellung
einfügen (GSA Shop)

Unterhalb dieser Seite erstellen Sie eine weitere Seite **Danke**, die wieder im Menü versteckt sein soll. Die Seite enthält lediglich ein Textelement mit einem kurzen Dankestext für die entgegengenommene Bestellung.

Im Frontend wird nach dem Check-Out des Warenkorbes (Button: **Zur Kasse**) die Übersicht über die Bestellung inklusive der gewünschten Liefer- und Rechnungsanschrift angezeigt (Abbildung 9.45). Sofern der Kunde die beiden Häkchen für die Akzeptanz der AGB und das Widerrufsrecht gesetzt hat, wird die Seite mit dem Dankestext für seine Bestellung aufgerufen. Die Bestellung ist damit abgeschlossen.

Abbildung 9.45:
Bestellübersicht im
Frontend (GSA Shop)

Ihre Bestellung im Überblick

| Ihre Rechnungsdaten | |
|----------------------------------|---------------|
| Gesamtsumme | |
| Summe Artikel brutto [EUR] | 568,97 |
| Summe Versandkosten brutto [EUR] | 10,71 |
| Gesamtsumme brutto [EUR] | 579,68 |
| Mwst. 19,0% [EUR]: 92,55 | |
| Gesamtsumme netto [EUR]: 487,12 | |

Rechnungsadresse

Test Tester
Teststr. 1
11111 Teststadt

Möchten Sie die **Bestellung an mehrere Lieferadressen versenden** ?

Lieferung Artikel 1 -4 von 4

| Artikel | Warenkorb ändern |
|---|------------------|
| Akku FootTech v67 Anzahl: 3 - Preis: 34,99 EUR - Art.Nr.: AK-45612-v67 | 104,97 |
| Hardware BarTech Anzahl: 1 - Preis: 464,00 EUR - Art.Nr.: HW-3490762-A | 464,00 |
| Versandkosten-Pauschale [EUR] Shop Inland (brutto) | 10,71 |
| Summe Lieferung brutto [EUR] | 579,68 |
| Mwst. 19,0% [EUR] 92,55 | |
| Summe Lieferung netto [EUR] 487,12 | |

Lieferadresse

Test Tester
Teststr. 1
11111 Teststadt

**Zahlungsweise
auf Rechnung (Banküberweisung)**

Ändern

Die Kontodaten für die Überweisung erhalten Sie auf der Rechnung.

| | |
|-------------------------------------|---|
| <input checked="" type="checkbox"/> | Ich habe die Allgemeinen Geschäftsbedingungen gelesen und akzeptiere diese hiermit. |
| <input checked="" type="checkbox"/> | Hiermit bestätige ich, daß ich über mein Widerrufsrecht informiert worden bin. |

JETZT BESTELLEN!

Warenkorb-Minianzeige

Ein weiteres Plugin ist die GSA Shop: Warenkorb-Minianzeige: Dieses Plugin kann per TypoScript und einem Template-Marker in der HTML-Designvorlage auf allen relevanten Seiten angezeigt werden. Um den aktuellen Warenkorb-Status in Miniatur-Darstellung sichtbar zu machen, wird das Plugin im Backend in einem *General Storage Folder* und im TS-Root-Template-Setup über die Markerdefinition eingebunden.



Abbildung 9.46:
Plugin Warenkorb
einfügen (GSA Shop)

Im Frontend zeigt der Miniwarenkorb dann die Anzahl und die Gesamtsumme aller Artikel im Warenkorb (Abbildung 9.47).



Abbildung 9.47:
Plugin
Warenkorb-Mini
einfügen (GSA Shop)

Artikel-Einzelansicht

Erstellen Sie eine Seite für die Artikel-Einzelansicht, setzen Sie den Haken bei im Menü verstecken und fügen Sie das Plugin GSA Shop: Einzelansicht ein. Auch bei diesem Plugin stehen Ihnen wieder die Optionen, die wir bei der Artikel-Anzeige bereits vorgestellt haben, zur Verfügung (Abbildung 9.38).



Abbildung 9.48:
Plugin
Artikel-Einzelansicht
einfügen (GSA Shop)

Diese Seite ist ein „Template“ für die Detailansicht aller Artikel, d.h. die Einstellungen sollten detaillierter sein als im Plugin Artikel-Anzeige und gelten dann für alle Artikel-Detailansichten. Über eine Konfiguration im Artikel selbst – Genaueres würde hier zu weit führen – können Artikel im Warenkorb automatisch mit einem Link zur jeweiligen Detailseite des Artikels versehen werden, wie bereits beschrieben (Abbildung 9.43).

Bestellarchiv

Erstellen Sie eine Seite namens **Bestellarchiv** für Ihre Bestellungen. Setzen Sie den Zugriff für diese Seite auf **Bei jedem Login anzeigen** und fügen Sie das Plugin **GSA Shop: Workflow / Bestellarchiv** mit dem Typ **Bestellarchiv** ein. Hier werden dann alle Bestellungen gelistet.

Abbildung 9.49:
Plugin Bestellarchiv
einfügen (GSA Shop)

The screenshot shows the configuration form for the 'GSA Shop: Workflow / Bestellarchiv' plugin. It includes a section for 'Plugin Optionen' with a 'FlexForm Konfiguration von Plugins.' field. Below this, there is a 'Plugin Typ' field set to 'Konfiguration (nur fuer Workflow)' and a 'Plugin Typ definieren' dropdown menu set to 'Bestellarchiv'.

Sofern der Workflow aktiviert ist (S. 221) und Sie hier den Plugin-Typ **Workflow** wählen, kann über die Flexforms eingestellt werden, ob die Seite eine Workflow-Sicht für den Shopbetreiber oder den Kunden darstellen soll.

Abbildung 9.50:
Plugin Bestellarchiv
Workflow (GSA Shop)

This screenshot shows the 'Workflow Typ' dropdown menu expanded, displaying three options: 'Kunden-Workflow (nur der dem eingeloggten FE-User zugeordnete Kunde)', 'Kunden-Workflow (nur der dem eingeloggten FE-User zugeordnete Kunde)', and 'Shopbetreiber-Workflow (Bestellungen aller Kunden anzeigen)'. The first option is currently selected.

Das Bestellarchiv ist eine Übersicht für den Kunden. In Abbildung 9.51 sehen Sie die Testbestellung. Hier kann der Benutzer vorhergehende Bestellungen mit wenigen Klicks als neue Bestellung wieder öffnen. Das ist besonders bei komplexen Bestellungen interessant, die sich wiederholen. Preise, Steuern, Adressen usw. werden automatisch, aber mit Hinweis für den Benutzer, an den aktuellen Stand angepasst (z. B. wenn sich der Preis oder der Steuersatz geändert hat). Mit einem Klick auf **Anzeigen** kann der Benutzer die jeweilige Bestellung anzeigen. Daraufhin erscheint diese ähnlich wie in Abbildung 9.45 zu sehen, allerdings ohne detaillierte Einzelpreise. Ein zusätzlicher Button **Diese Bestellung als neue Bestellung öffnen** ermöglicht eine erneute Bestellung dieser Artikel und bietet die Möglichkeit, die Bestellung zu modifizieren, und zwar sowohl was die Artikel als auch was die Rechnungs-/Lieferanschriften und die Zahlungsweise betrifft.

Übersicht über bisherige Bestellungen

| | Bestelldatum | Vorgangs-Nr. | Summe netto | Summe brutto | |
|----------|-------------------|-----------------|-------------|--------------|----------|
| Anzeigen | 11.10.2007, 11:51 | RE-200710/00141 | 749.84 EUR | 892.31 EUR | Anzeigen |

Abbildung 9.51:
Bestellarchiv im
Frontend (GSA Shop)

Workflow

Wenn der Workflow im Constant Editor aktiviert worden ist, erhält man jeweils eine Ansicht für den eingeloggten Frontendbenutzer (bzw. den Kunden) und den Shop-Administrator. Hiermit sind Mehrparteien-Workflows möglich, z. B. löst ein Kunde eine Bestellung aus, die Bestellung muss vom Chefeinkäufer dieses Kunden und danach vom Shop-Administrator freigegeben werden (z. B. für eine Kreditlimit-Überprüfung des Shopbetreibers o.Ä.). Die Einbindung der Seiten für die Workflow-Ansichten erfolgt wie im Zusammenhang mit dem Bestellarchiv beschrieben.

Benutzerverwaltung

Für die Benutzerverwaltung verwendet der GSA Shop eine Extension namens `pt_gsuserreg`. Hier werden die Kundendaten und die Frontendbenutzer-Daten verwaltet. Ein Kunde (z. B. eine Firma) kann über mehrere Frontendbenutzer verfügen. Die möglichen Lieferadressen werden hier ebenfalls – momentan unter dem Arbeitstitel **Adressaufkleber** – verwaltet. Diese Benennung kann aber jederzeit über die lokalen Spracheinstellungen individuell überschrieben werden. Neben Adressdaten können hier ebenfalls die Zahlungsmethoden definiert werden – welche Zahlungsmethoden dem Kunden erlaubt sind, wird vom Shop-Administrator im Constant Editor eingestellt.

Kreditkartenzahlung

Erstellen Sie, falls Sie Kreditkartenzahlungen ermöglichen möchten, eine weitere Seite namens **Kreditkartenzahlung** unterhalb von Bestellungen. Auch diese Seite ist im Menü versteckt. Momentan unterstützt der GSA Shop die Kreditkartenzahlung via **Heidelpay**¹⁰ über die Extension `pt_heidelpay`. Die Extension ist derzeit noch nicht im TER verfügbar, wird Interessenten jedoch auf Anfrage¹¹ zur Verfügung gestellt. Eine Installationsanweisung

¹⁰ <http://www.heidelpay.de/>

¹¹ <http://punkt.de>

liegt der Extension bei. Mittelfristig soll optional die Kreditkartenzahlung mit der Extension `pt_payment` (im TER verfügbar) für den Kreditkartenabrechnungsanbieter Cardia¹² integriert werden. Der Shopbetreiber kann dann wählen, ob er Zahlungen über Heidelpay oder Cardia einsetzen möchte. Später ist geplant auch weitere Zahlungsmöglichkeiten anzubinden.

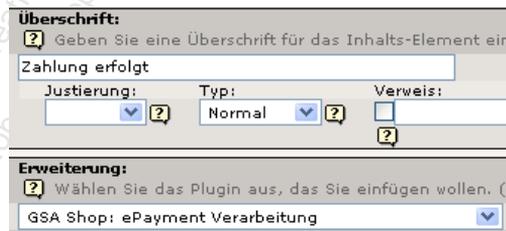
Wählen Sie hier also das Plugin der Extension `pt_heidelpay` aus und fügen es in diese Seite ein.

Abbildung 9.52:
Plugin Heidelpay
(GSA Shop)



Unterhalb dieser Seite erstellen Sie zwei weitere Seiten, die ebenfalls nicht im Menü angezeigt werden sollen: Eine Seite mit der Bezeichnung **Zahlung erfolgt**, und eine namens **Zahlung fehlgeschlagen**. In diese beiden Seiten fügen Sie jeweils das Plugin **GSA Shop: ePayment Verarbeitung** ein. Diese Seiten werden aufgerufen, sofern eine Kreditkartenzahlung erfolgt ist (Abbildung 9.53) bzw. wenn die Bezahlung fehlgeschlagen ist.

Abbildung 9.53:
Plugin epayment
Verarbeitung
einfügen (GSA Shop)



Weitere Bezahlmethoden

Die möglichen Bezahlmethoden werden vom Shop-Administrator konfiguriert. Derzeit unterstützt werden Abbuchung, Rechnung und Kreditkarte. Die Bezahlmethode pro Bestellung wird auf der Bestell-Übersichtsseite vom Kunden (aus den verfügbaren) ausgewählt; wenn er keine Auswahl vornimmt, wird die Default-Zahlungsmethode aus seinen Kundendaten verwendet.

Der Shop verfügt jetzt über eine Anmeldung über die Benutzerregistrierung mit `pt_gsouserreg` und eine Bankdatenüberprüfung, eine Artikel-Standardansicht und eine Einzelansicht der Artikel, über Warenkorb-Funktion, Mini-Warenkorb und verschiedene Bezahlmethoden: Kreditkarte, Rechnung und Bankeinzug. Diese Funktionen von GSA Shop sollen an dieser Stelle genügen. Wenden Sie sich nun dem Template zu.

¹² http://www.cardia.no/View/Cardia_EN/index.html

9.2.4 TS-Template

Im Beispielshop ist das TS-Template in das Template der Rootseite integriert. Hier müssen dann auch die statischen Templates über `Include static from extensions` eingefügt werden (Abbildung 9.54).

Eine sauberere Konfiguration ist natürlich wie immer bei TYPO3, die statischen Templates nur auf den Seiten einzubinden, auf denen die jeweilige Extension eingebunden ist. Dies erfordert aber deutlich mehr Aufwand und ist im Beispielshop der Einfachheit halber im Haupt-Template eingebunden.



Abbildung 9.54:
Statische Templates
zuweisen (GSA Shop)

Alle Einstellungen für die Grundkonfiguration des Shops können über die unterschiedlichen Kategorien der Extension im Constant Editor vorgenommen werden; eine Beschreibung würde den Rahmen dieses Ausblicks sprengen. Die Konfigurationsoptionen sind jedoch bereits jetzt im Constant Editor und im der Extension beiliegenden Manual ausführlich kommentiert. Ein interessanter Punkt ist die Zuweisung der einzelnen Shopseiten im Constant Editor (PT_GSASHOP : : GENERAL): Falls Sie den im Constant Editor bei der jeweiligen Seiten-Einstellung als **Default** vorgeschlagenen Namen als **Alias** für Ihre Seiten verwenden, brauchen Sie hier keine PIDs für die Seiten anzugeben, der Shop findet die Seiten dann automatisch. Wenn Sie keinen Alias oder nicht den Default-Alias für Ihre Shop-Plugin-Seiten verwenden, müssen Sie hingegen die PID oder den Alias der Seiten im Constant Editor angeben.

Man darf gespannt sein, was die künftigen Versionen von GSA Shop an weiteren interessanten Features bieten werden.

9.3 PDF Shop

PDF Shop ist ein kleines Shopsystem, das es Ihnen ermöglicht, PDFs zum Download oder Mail-Versand anzubieten. Wir haben die Version 0.2.1, die als „Stable“ deklariert ist, über den Extension Manager installiert. Es werden keine weiteren Extensions vorausgesetzt. Das Shopsystem ist zur Drucklegung des Buches noch nicht vollständig. So fehlt die Implementierung von Bezahlmethoden und Preisen noch vollständig. Wer allerdings PDF-

Dateien kostenfrei zum Download anbieten möchte, kann diese Extension schon jetzt nutzen.

Erstellen Sie nach der Installation eine Seite in Ihrem Seitenbaum sowie einen Produkte-SysOrdner, in dem Sie später Ihre PDF-Dateien als Produkte verwalten. Fügen Sie das Plugin **PDF Shop** in die eben erstellte Seite ein und weisen Sie als Ausgangspunkt den neuen Produkte-SysOrdner zu.

Erstellen Sie nun in gewohnter Weise ein Extension-Template namens +PDF Shop und weisen Sie es Ihrem Haupt-Template zu.

Fügen Sie folgenden Code in das Setup des Template +PDF Shop ein:

```
plugin.tx_pdforder_pil.pid = 322

plugin.tx_pdforder_pil.templateFile =
    fileadmin/osp/templates/pdfshop.html
#plugin.tx_pdforder_pil.showResults = 10
plugin.tx_pdforder_pil.emailFrom = mail@mytypo3domain.de
plugin.tx_pdforder_pil.emailReply =
plugin.tx_pdforder_pil.emailTo = mail@mytypo3domain.de
plugin.tx_pdforder_pil.emailSubject = PDF Bestellung
plugin.tx_pdforder_pil.defaultErrorMessage = Bitte füllen Sie die
    benötigten Felder aus.
plugin.tx_pdforder_pil.confirmText = Vielen Dank für Ihre Bestellung.
plugin.tx_pdforder_pil.contactFormHeadline = PDF Shop Kontakt
plugin.tx_pdforder_pil.defaultOrderMessage = Hiermit bestellen wir die
    folgenden PDF-Dateien:
```

Mit der ersten Zeile weisen Sie die PID des SysOrdnerns, mit der zweiten das HTML-Template zu. Die folgende Zeile zeigt zukünftig die Anzahl der Suchergebnisse. Dieses Feature ist aktuell noch nicht implementiert, deshalb ist die Zeile auskommentiert. Weiterhin definieren Sie die E-Mail-Adressen des Absenders und des Empfängers sowie die Betreffzeile der E-Mails. Im Anschluss legen Sie eine Fehlermeldung fest, die erscheint, wenn der Kunde nicht alle benötigten Adressfelder ausgefüllt hat. Dieser Text sollte möglichst kurz sein, denn die Meldung erscheint im nicht ausgefüllten Feld. Danach erstellen Sie einen kurzen Dankestext für die erfolgte Bestellung, eine Überschrift für ein Kontaktformular sowie eine weitere Überschrift für die Bestellungen.

Wechseln Sie jetzt in Ihren Produkte-SysOrdner und legen Sie einen neuen Datensatz an. Wählen Sie aus der Liste **PDF Shop**.

Abbildung 9.55:
SysOrdner: neuen
Datensatz anlegen
(PDF Shop)



Füllen Sie die Felder aus, indem Sie zunächst einen Titel und einen Beschreibungstext eintragen. Das Feld PDF-Format – ebenfalls ein Beschreibungsfeld – gibt das Format der Datei (im Beispiel DIN A4) an. Dann weisen Sie die PDF-Datei zu, wahlweise von Ihrem lokalen System oder aus dem Fileadmin-Verzeichnis, geben ein Bild an und speichern den Datensatz ab.

The screenshot shows the 'PDF Shop [1] - 1' administration interface. It contains several sections:

- Verstecken:** A checkbox that is currently unchecked.
- Überschrift:** A text input field containing 'Leseprobe 2'.
- Beschreibung:** A large text area containing the text 'Hier finden Sie eine Leseprobe zum Download.'
- Format der PDF Datei:** A dropdown menu set to 'DIN A 4'.
- PDF:** A section for uploading a PDF file. It shows a file named 'freiekultur_leseprobe.pdf' with a 'No thumb generated!' warning. A 'Durchsuchen...' button is visible.
- Bild:** A section for uploading an image. It shows a file named 'freie_kultur.png' with a thumbnail. A 'Durchsuchen...' button is also present.

Abbildung 9.56:
SysOrdner: Produkt
erstellen (PDF Shop)

Im Frontend sehen Sie jetzt die gelisteten Produkte. Das HTML-Template wurde aus der Extension genommen und nicht angepasst.

The screenshot shows the frontend of the PDF Shop. It displays a list of products under the heading 'PDF Shop':

- Leseprobe 1:** Includes a thumbnail for 'Freie Kultur', the description 'Hier finden Sie eine Leseprobe zum Download.', and the format 'DIN A 4'. Below it, there is a 'PDF-Download' button and an 'Anzahl' field with the value '1'.
- Leseprobe 2:** Includes a thumbnail for 'Nagios', the description 'Hier finden Sie eine Leseprobe zum Download.', and the format 'DIN A 4'. Below it, there is a 'PDF-Download' button and an 'Anzahl' field with the value '1'.

 At the bottom of the list, there is a link 'zum Kontaktformular'.

Abbildung 9.57:
Listenansicht (PDF
Shop)

Über einen Klick auf das PDF können Sie die Datei aufrufen. Ins Feld **Anzahl** geben Sie die gewünschte Anzahl der Dokumente ein, und ein Klick auf den Button leitet Sie zur Bestellung. Dort geben Sie Ihre Adressdaten ein. In einer Textbox steht der Text Ihrer Bestellung. Durch einen Klick auf den Submit-Button wird die Bestellung ausgelöst.

Abbildung 9.58:
Bestellung (PDF
Shop)

Wir versichern Ihnen, dass Ihre Daten nicht an Dritte weitergeleitet werden.

Die im Setup des Extension-Template erstellte Nachricht „Vielen Dank für Ihre Bestellung“ erscheint, und die Bestellung ist damit abgeschlossen. Der Empfänger erhält seine Bestellbestätigung per E-Mail. Wie bereits erwähnt, fehlen in diesem Shop einige wichtige Features und es bleibt abzuwarten, ob diese in absehbarer Zeit implementiert werden.

9.4 Trade

Die Shop-Extension Trade von Steve Ryan und Roger Bunyan wurde in der zur Drucklegung aktuellen Version 0.1.8 installiert. Das HTML-Template basiert aktuell auf einem Tabellen-Layout. Eine kurze Installationsanleitung in englischer Sprache liegt der Extension bei.

Installieren Sie die Extension wie gewohnt über den Extension Manager. Zudem benötigen Sie die Frontendbenutzerregistrierung, da die Adressen der Kunden darüber verwaltet werden. In diesem Testshop werden alle Shopansichten – außer der Einzelansicht des Produkts – über ein Plugin gesteuert, d. h. alle Stationen dieses Shops, Listenansicht, Warenkorb, Adress-

daten eingeben etc. bis zur abgeschlossenen Bestellung, werden über die Seite Trade, die gleich erstellt werden soll, gesteuert. Sofern Sie eine andere Konfiguration benötigen, können Sie für jede Station des Shops eine eigene Seite mit Plugin erstellen und über das Setup des Extension-Templates die entsprechende PID zuweisen.

9.4.1 Plugin einfügen

Erstellen Sie nun die Seite Trade und fügen Sie dort das Plugin Trade ein (Abbildung 9.59).

Abbildung 9.59:
Plugin einfügen und
Flexforms (Trade)

Basic Settings

Über die Flexforms können Sie nun die Basiskonfiguration erstellen. Vergeben Sie eine E-Mail-Adresse für den Shop, tragen Sie die Adresse des Shopbetreibers ein und wählen Sie bereits hier Ihre Bezahlmethoden aus. Trade bietet zunächst drei Methoden an:

- PayPal
- Eway (ein australisches Zahlungssystem)
- Cheque/Money Order/Direct Deposit

Wenn Sie PayPal auswählen, können sie weiter unten in ein entsprechendes Feld Ihre Paypal-E-Mail-Adresse eingeben, für Eway müssen Sie Ihren Kundencode eintragen. Für Eway steht außerdem ein Testmodus zur Verfügung. Für den Test von PayPal können Sie die Sandbox verwenden, wie in Abschnitt 6.6.1 ab Seite 151 beschrieben.

Weitere Einstellung für Lieferung und Zahlung können Sie über TypoScript im Setup des Extension-Template vornehmen.

Sofern Sie bereits eine mehrsprachige Seite nutzen, werden Ihnen die Flexforms für die installierten Sprachen untereinander dargestellt, so dass Sie die Spracheinstellungen in einem Arbeitsschritt vornehmen können (Abbildung 9.60).

Abbildung 9.60:
Plugin in
verschiedenen
Sprachen einrichten
(Trade)

The screenshot displays the configuration interface for the 'Trade' plugin in TYPO3, showing settings for two languages: German (DE) and English (EN). The interface is divided into two main sections, one for each language, with tabs for 'Basic Settings' and 'Advanced Settings'. The 'Basic Settings' tab is selected for both.

DE: (German)

- Shop Owner Email:**
- Shop Owner Details:** Testshop
Teststr 1
11111 Testort
- Payment Options:**
 - Paypal
 - Eway
 - Cheque/Money Order/Direct Deposit
- Paypal Email Address:**
- Eway Merchant Code:**
- Eway Test Mode:**

EN: (English)

- Shop Owner Email:**
- Shop Owner Details:** Testshop
Teststr 1
11111 Testort

Advanced Settings

In die Registerkarte für die erweiterte Konfiguration tragen Sie die Seite(n) des Shops ein, auf der Sie später Ihre Produkte und Kategorien anlegen werden. In dieser Installation ist das ebenfalls die Seite **Trade**.

Eine Checkbox zur Aktivierung des Debug-Modus kann direkt im Plugin aktiviert werden.

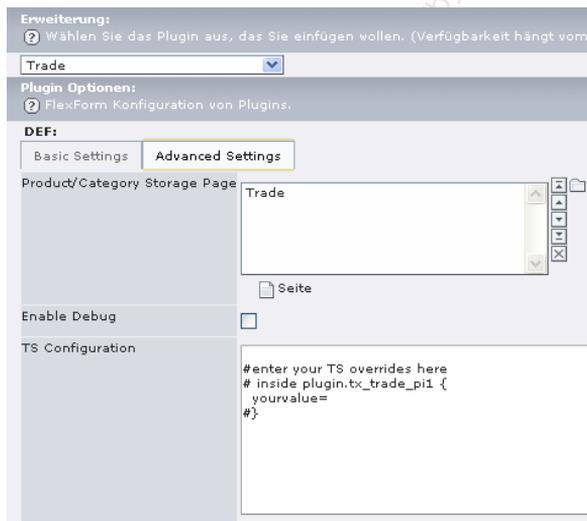


Abbildung 9.61:
Flexforms: Erweiterte
Einstellungen (Trade)

In der erweiterten Konfiguration steht Ihnen ein Feld für die TypoScript-Konfiguration Ihres Shops zur Verfügung. Sofern Sie hier Einträge vornehmen, werden die Default-Einstellungen der TypoScript-Konfiguration in der Extension überschrieben. Dieses Vorgehen ist hier nicht beschrieben. Alle weiteren Einstellungen erfolgen über das Extension-Template `+trade`.

9.4.2 Einzelansicht

Erstellen Sie eine weitere Seite für die Einzelansicht. Nehmen Sie dort die gleichen Basiseinstellungen vor, die Sie auf der Hauptseite des Shops eingefügt haben, und speichern Sie den Datensatz ab. Die PID dieser Seite wird später über Constants und Setup zugewiesen.

9.4.3 Produkte und Kategorien

Wechseln Sie jetzt ins Webmodul **Liste**. Sie können hier Produkte, Kategorien, Orders und Order Status auswählen bzw. anlegen.

Wählen Sie aus der Liste (über Neuen Datensatz anlegen) Products aus.

Abbildung 9.62:
Webmodul Liste:
Neuen Datensatz
anlegen (Trade)



Hier verfahren Sie für die Erstellung der Produkte und Kategorien analog dem Vorgehen bei tt_products. Erstellen Sie zunächst ein Testprodukt und, wenn gewünscht, eine Kategorie für dieses Produkt.

Abbildung 9.63:
Backendformular
Produkt (Trade)

Vergeben Sie einen Namen, einen Infotext, einen Preis, die Artikelnummer und den Steuersatz, fügen Sie ein Bild ein und speichern Sie das Produkt ab. Auch Trade bietet leider keine Möglichkeit, Alternativtexte zu Bildern zu hinterlegen.

Falls Sie Kategorien verwenden, weisen Sie dem Produkt die entsprechende Kategorie zu. Sofern Sie mehrere Preise für unterschiedliche Benutzergruppen definieren möchten, müssen die Preise hier eingetragen und die Feldzuweisung über das Setup des Extension-Template definiert werden.

Sie können noch einige weitere Einstellungen vornehmen, z. B. ein oder mehrere Datenblätter hinzufügen. Das Verfahren ist analog dem, das Sie von `tt_products` kennen. Für die Testinstallation sollen diese Einstellungen jedoch genügen.

9.4.4 Shop mit registrierten Benutzern

Die Extension arbeitet mit der Frontendbenutzerregistrierung zusammen. Sofern sie bereits installiert ist, greift der Shop auf diese Benutzer zu. Über die Constants muss lediglich der SysOrdner mit den Benutzern zugewiesen werden. Die Extension erweitert die Frontendbenutzerregistrierung um eigene Felder für die Lieferanschrift. Der Shop verfügt über ein eigenes Login, das im HTML-Template daktiviert werden kann, wenn Sie ein allgemeines Login für Ihren Internetauftritt nutzen möchten. Sofern Kunden auch ohne vorherige Registrierung Bestellungen durchführen, werden die Kundendaten nach dem Auschecken des Warenkorbs eingegeben. Die Daten werden in der Frontendbenutzerregistrierung gespeichert. Die neu angelegten Benutzer erhalten mit der Bestätigungs-E-Mail die Zugangsdaten zum Shop und können diese für künftige Bestellungen nutzen.

9.4.5 TS-Template

Erstellen Sie ein Extension Template namens `+Trade` im Template SysOrdner und nehmen Sie dort Ihre Shop-Konfiguration vor. Ein statisches Template besitzt Trade nicht. Weisen Sie dieses Extension-Template in gewohnter Weise dem Haupt-Template zu. In diesem einfachen Beispiel wurden in den Constants lediglich die PIDs zugeordnet. Der Constant Editor steht Ihnen für diese Extension nicht zur Verfügung.

Constants

```
tradepid=267
tradestoragepid=267
userstoragepid=52
tradesingleviewpid=324
```

Die PID `trade_pid` ist die der Seite, auf der die Trade-Informationen im Frontend zur Verfügung stehen, in diesem Fall die Seite mit dem Titel **Trade** und der Seiten-ID 267. Der Eintrag `tradestorage_pid` definiert die Seite im Backend, in der die Produkte, Kategorien, Bestellungen etc. angelegt und verwaltet werden. `userstorage_pid` verweist auf den SysOrdner mit den Frontendbenutzern und `tradesingleview_pid` definiert die PID für die Einzelansicht.

Im Frontend können Sie jetzt Ihre ersten Testbestellungen durchführen.

Setup

Das Setup wurde aus dem Beispiel-Setup der Extension kopiert und in einigen Punkten angepasst. Im folgenden Code wird über die Globale (1) zunächst definiert, dass alle Benutzer der Gruppe 1 den Preis erhalten, der im Feld **Preis 2** (Abbildung 9.63) bereits einen Wert enthält. Der Abschnitt (2) definiert die Listenansicht des Shops; es wird eine Shopsuche definiert und der verwendete Subpart des Templates zugewiesen. Das Label für die Suche bleibt in diesem Fall leer, da die Suche in der Standardinstallation sonst doppelt erscheint. Die Sortierung erfolgt nach dem Produkttitel. Als nächstes (3) sollen alle neuen Produkte angezeigt werden, die in den letzten 7 Tagen angelegt wurden, und in Abschnitt (4) sollen Sonderangebote gelistet werden, die über **Special Offer** als Sonderangebote definiert wurden.

Im weiteren Verlauf (5) werden die PIDs zugewiesen. Sie sehen hier, dass eine Variablenliste verwendet wird, um die Seiten oder SysOrdner zu definieren, die über die Constants zugewiesen sind. Die eigentliche Zuordnung der PIDs nehmen Sie über die Constants vor. Eine Liste der Variablen finden Sie im Beispiel-Setup der Extension **Trade**.

```
# (1) Preise für die Benutzergruppe mit der UID 1
[usergroup=1]

plugin.tx_trade_pil.priceField=2
[GLOBAL]
plugin.tx_trade_pil.maxListRows=2

# (2 )Kategorien verwenden
plugin.tx_trade_pil.lists >
plugin.tx_trade_pil.lists {
    default {
        title = Search
        label =
        extraWhere =
        templateSubpart = ITEM_LIST
        search = 1
        orderBy = title
    }
}
# (3) Neue Artikel der letzten 7 Tage
```

```

        recent {
            title = Recently Added Products
            label = Neueste Produkte
            # last 7 days
            extraWhere = DATE_SUB(CURDATE(),INTERVAL 7 DAY)
            <= crdate
            templateSubpart = ITEM_LIST
            search = 0
            orderBy = crdate desc
        }
        # (4) Angebote
        specialoffers {
            title = Special Offers
            label = Sonderangebote
            extraWhere= special=1
            templateSubpart=ITEM_LIST
            search=0
            orderBy=title
        }
    }
# (5 )PIDs zuweisen, s. Constants

plugin.tx_trade_pil.PIDS {
    #storage pages
    productstorage=$tradestoragepid
    categorystorage=$tradestoragepid
    orderstorage=$tradestoragepid
    userstorage=$userstoragepid
    # presentation pages
    list=$tradepid
    list_default=$tradepid
    list_search=$tradepid
    list_recent=$tradepid
    list_specialoffers=$tradepid
    singleview=$tradesingleviewpid
    basket=$tradepid
    basket_overview=$tradepid
    checkout=$tradepid
    user_details=$tradepid
    user_shipping_details=$tradepid
    user_shipping_payment_details=$tradepid
    usersave=$tradepid
    login=$tradepid
    lost_password=$tradepid
    confirm=$tradepid
    thanks=$tradepid
    order_history_list=$tradepid
    order_history_single=$tradepid
    copy_address_to_shipping=$tradepid
    wishlist_list=$tradepid
}

```

Bezahlung und Lieferung

Wie bereits bei der Seitenerstellung beschrieben, müssen einige Einstellungen für die Bezahlmethoden bereits im Plugin vorgenommen und können dort auch über TypoScript definiert werden. In diesem Shop sind im Plugin alle Zahlungsmethoden ausgewählt, der TypoScript-Code jedoch über das Setup des Extension-Template +Trade definiert.

Alle weiteren Einstellungen werden in diesem Beispiel über das Extension-Template +Trade vorgenommen. Der folgende Code wird Ihnen möglicherweise bekannt vorkommen, die Bezahl- und Liefermethoden werden sehr ähnlich wie in `tt_products` behandelt.

Mit `radio = 1` legen Sie fest, dass die Bezahlmethoden im Frontend über einen Radiobutton, nicht über eine Combobox ausgewählt werden.

Zunächst wird die Bezahlmethode für das Payment-Gateway Eway definiert. Hier muss lediglich der Kundencode (`merchantCode`) oder für Testzwecke das Test-Gateway (`useTestGateway`) ausgewählt werden. Alle anderen Einstellungen sind dem Default-Setup entnommen.

Die nächste Bezahlmethode definiert die Bezahlung via PayPal. Ergänzend muss hier die PayPal-E-Mail-Adresse eingegeben werden. Für die Bezahlung mit PayPal wird über `priceTax` außerdem ein Zuschlag von EUR 1,- veranschlagt.¹³

Die nächste Bezahlmethode ist die Vorkasse, auch hier wurde – wie bei Eway und PayPal – ein Bild zugewiesen und für die Vorkasse ein Zuschlag von EUR 3,- verlangt.

Wenden Sie sich nun den Versandmethoden zu.

Dem **Postversand** wurde wieder ein Bild zugewiesen und eine Versandkostenpauschale von EUR 5,- erhoben. Die **Luftfracht** hat eine etwas höhere Gebühr, für den **Selbstabholer** fallen keine zusätzlichen Kosten an. Ihm steht lediglich die Bezahlmethode **Vorkasse** zur Verfügung. Die Kreditkartenzahlung ist mit `excludePayment`, das Sie bereits von `tt_products` kennen, ausgeschlossen. Der Punkt `hideDetails` bestimmt, ob im Frontend Datainformationen über den Versand im Trade-Menü angezeigt werden können oder nicht.

```
# Beispielkonfiguration Payment
plugin.tx_trade_pil.payment {
    radio = 1
    10.title = Eway
    10.description = Eway-zahlung über Payment Gateway
    10.image.file = EXT:trade/res/creditcard.gif
    10.priceTax =
    10.detailsOK = if ($this->validateCreditCard($this->pi
Vars['card_number'], $this->payment['card_exp_month'], $this->payment['c
```

¹³ Ein Redirect zurück zum Shop muss separat eingestellt werden.

```

ard_exp_year'],$this->payment(['card_name'])) $detailsOK=true; else $de
tailsOK=false;

    10.detailsError = Fehler bei der Eingabe. Bitte
        versuchen Sie es erneut.
    10.handleScript = EXT:trade/lib/eway_payment.php
    10.merchantCode =
    10.useTestGateway =

    20.title = PayPal
    20.description = PayPal-zahlung über Payment Gateway
    20.image.file = EXT:trade/res/paypal.gif
    20.priceTax = 1
    20.handleScript = EXT:trade/lib/paypal.php
    20.paypalEmail =

    40.title = Vorkasse
    40.description = Sobald Ihre Zahlung eingegangen ist,
        wird die Ware versendet.
    40.image.file = EXT:trade/res/money.gif
    40.priceTax = 3

    ## Beispielkonfiguration Shipping:
plugin.tx_trade_pil.shipping >
plugin.tx_trade_pil.shipping {
    10.title = Postversand
    10.image.file = EXT:trade/res/pakettrans.gif
    10.priceTax = 5
    10.hideDetails = 0

    20.title = Luftfracht
    20.image.file = EXT:trade/res/postdanmark.gif
    20.priceTax = 10
    20.hideDetails = 0

    40.title = Selbstabholer
    40.priceTax = 0
    40.excludePayment = 10, 20
    40.hideDetails = 1
}
}

```

Diese Konfiguration soll für den Beispielshop genügen.

9.4.6 Der Shop im Frontend

In Abbildung 9.64 sehen Sie die Listenansicht des Shops. Oberhalb der Produktliste, die hier lediglich aus einem Testprodukt besteht, finden Sie

das Trade-Menü. Über dieses können Sie navigieren, und – sofern Sie die `hideDetails` nicht auf 1 gesetzt haben – weitere Details Ihrer Bestellung ansehen. Unterhalb dieses Menüs wird in der Standardansicht eine Login-Box angezeigt. Im Bild ist der Benutzer bereits angemeldet. Der angemeldete Benutzer gehört der Benutzergruppe 1 an. Wie im Setup und im Produkt eingestellt, wird ihm damit das Preisschema 2 zugewiesen. Dazu erfolgt hier eine Textausgabe. Die Übersetzungen der Texte nehmen Sie ebenfalls im Setup vor. Weitere Informationen dazu finden Sie im entsprechenden Abschnitt zu `tt_products`. Über das Suchfeld können Sie innerhalb des Shops nach Produkten suchen.

Abbildung 9.64:
Listenansicht im
Frontend (Trade)



Weiterhin sehen Sie innerhalb der Listenansicht einen Miniwarenkorb, der im Standardshop integriert ist. Durch einen Link auf die Punkte **Neueste Produkte** und **Sonderangebote** erhalten Sie eine Übersicht über alle Produkte, die in den letzten 7 Tagen angelegt wurden, bzw. die, bei denen der Haken **Special Offer** gesetzt ist (Abbildung 9.63). Damit in diesem Menü der Punkt **Suche** nicht angezeigt wird, ist das Label für die Suche leer (s. Setup). Ferner finden Sie in dieser Ansicht das Eingabefeld für die Bestellmenge und die entsprechenden Links, um die Produkte in den Warenkorb zu legen bzw. in den Warenkorb zu wechseln (**add** und **checkout**). Der Link **Exchange Rates** leitet auf eine externe Seite, die über Umrechnungskurse für Währungen informiert.

Ein Klick auf den Titel oder den **more info**-Link des Produkts leitet zur Einzelansicht. Die Einzelansicht verfügt in der Standardkonfiguration ebenfalls über eine Warenkorbfunktion, zeigt die Produktinformationen und die Artikelnummer an. Ein Link leitet zurück zur Listenansicht. Da der Beispielshop über nur eine Listenansicht mit nur einem Produkt verfügt, ist der Trade-Menüpunkt **Select Products** erst jetzt verfügbar.

Sobald Sie den Warenkorb auschecken, erhalten Sie eine Übersicht über die Produkte im Warenkorb und können zwischen den im TS-Template definierten Bezahl- und Liefermethoden auswählen (Abbildung 9.65).

| Basket Contents: | Price: | Qty: | Total: |
|---|---------|------|----------------|
| Ein Testprodukt | \$20.00 | 1 | \$20.00 |
| <input type="button" value="Update"/> | | | |
| Delivery: | | | |
| Ground | | | \$5.00 |
| Select payment method: | | | |
| <input type="radio"/> Eway  | | | |
| <input type="radio"/> PayPal  | | | |
| <input type="radio"/> Vorkasse  | | | |
| <input type="radio"/> Rechnung  | | | |
| Eway-zahlung über Payment Gateway | | | \$0.28 |
| TOTAL (AUD \$): | | | \$25.28 |
| <input type="button" value="Go to payment/ordering"/> | | | |

Abbildung 9.65:
Warenkorb
auschecken (Trade)

Die Menüpunkte **Payment/Shipping Details** und **Customer Details** können ausgewählt werden, sobald der Warenkorb ausgecheckt wird. **Payment/Shipping Details** löst die Ansicht für die Auswahl der Bezahl- und Liefermethode aus und **Customer Details** zeigt die hinterlegten Adressdaten an. Hier kann der Kunde seine Adressdaten ändern. Diese Ansicht erhält auch der nicht registrierte Kunde, um seine Daten einzugeben (Abbildung 9.67).

| Qty x Product: | Price: | Total: |
|--|---------|----------------|
| 1 x Ein Testprodukt | \$20.00 | \$20.00 |
| Shipping: | | |
| Ground | | \$5.00 |
| Payment Method: | | |
| Number: <input type="text"/> | | |
| Credit Card Details: Expiry Date: 01 - 2007 | | |
| Card name: <input type="text"/> | | |
| Payment Processing Fee: | | \$0.28 |
| Total amount in AUD \$: | | \$25.28 |
| Invoice Address: | | |
| Testster Teststr. 1 Teststadt, 11111 Germany Phone: 2222222 Email: tester@mytypo3domain.de | | |
| Delivery Address: | | |
| Testster Teststr. 1 Teststadt, 11111 Germany | | |
| Comments and requests: | | |
| <input type="text"/> | | |
| I agree to the Terms and Conditions <input type="checkbox"/> | | |
| I would like to be sent email including more information about our products. <input checked="" type="checkbox"/> | | |
| <input type="button" value="Finalise order"/> | | |

Abbildung 9.66:
Warenkorbübersicht
(Trade)

Durch einen Klick auf **Go to payment/Ordering** wechseln Sie in die Bestellung (Abbildung 9.66) und erhalten eine Gesamtübersicht. Dort bestätigen Sie die AGB und lösen dann die Bestellung aus.

In den Beispielbildern kauft ein bereits angemeldeter Benutzer im Shop ein. Sofern der Benutzer noch nicht registriert ist, muss er seine Adresdaten nach dem Auschecken des Warenkorbs eingeben. Sofern sich Rechnungs- und Lieferadresse nicht unterscheiden, kann der Kunde durch Klick auf den Button **Copy** seine Rechnungsanschrift in die Lieferanschrift kopieren (Abbildung 9.67). Beide Anschriften werden in der Frontendbenutzerregistrierung verwaltet. Trade erweitert die Frontendbenutzerregistrierung um die entsprechenden Felder.

Abbildung 9.67:
Rechnungs- und
Lieferanschrift
(Trade)

| Invoice Details | Shipping Details |
|---|---|
| Company <input type="text"/> | <input type="button" value="Copy"/> If shipping details are the same as the invoice detail, click to copy |
| Title <input type="text"/> | |
| *Full Name <input type="text" value="Tester"/> | *Contact Name <input type="text" value="Tester"/> |
| *Address <input type="text" value="Teststr. 1"/> | *Address <input type="text" value="Teststr. 1"/> |
| *City <input type="text" value="Teststadt"/> | *City <input type="text" value="Teststadt"/> |
| *Postcode/zip <input type="text" value="11111"/> | *Postcode/ZIP <input type="text" value="11111"/> |
| *State <input type="text"/> | *State <input type="text"/> |
| *Country <input type="text" value="Germany"/> | *Country <input type="text" value="Germany"/> |
| *Telephone <input type="text" value="2222222"/> | <input type="button" value="Checkout"/> |
| *Email <input type="text" value="tester@mytypo3domain.de"/> | |
| Homepage <input type="text"/> | |

9.4.7 Order Status

Im Webmodul **Liste** der Shopseite **Trade** finden Sie die Bestellungen und können diesen Bestellungen einen Bestellstatus zuweisen. Für diese einfache Bestellverwaltung planen die Entwickler die Implementierung eines Tracking-Moduls, das zur Drucklegung noch nicht fertiggestellt ist. Über **Order Status** legen Sie den Zustand der Bestellungen an. Hier stellen Sie alle Status ein, die Ihr Produkt nach der Bestellung annehmen kann: „Produkt wurde versendet“, „Produkt wurde bezahlt“ oder Ähnliches. Diesen Status können Sie den Bestellungen zuordnen und dort weitere Kommentare und einen Trackingcode eintragen. Erstellen Sie zunächst einige Status. Wechseln Sie ins Webmodul **Liste** und legen Sie mit **Neuen Datensatz anlegen** mehrere **Order Status** an (Abbildung 9.68). Durch einen Klick auf das Diskettensymbol mit dem grünen Kreuz können mehrere Status hintereinander angelegt und gespeichert werden.

Abbildung 9.68:
Order Status
erstellen (Trade)

Diese Statusmeldungen können Sie nun den Testbestellungen zuweisen, die in dieser Testinstallation der Einfachheit halber ebenfalls auf dieser Seite gespeichert werden. Rufen Sie die Bestellung auf und weisen über die Combobox den eben erstellten Status **Working** zu.

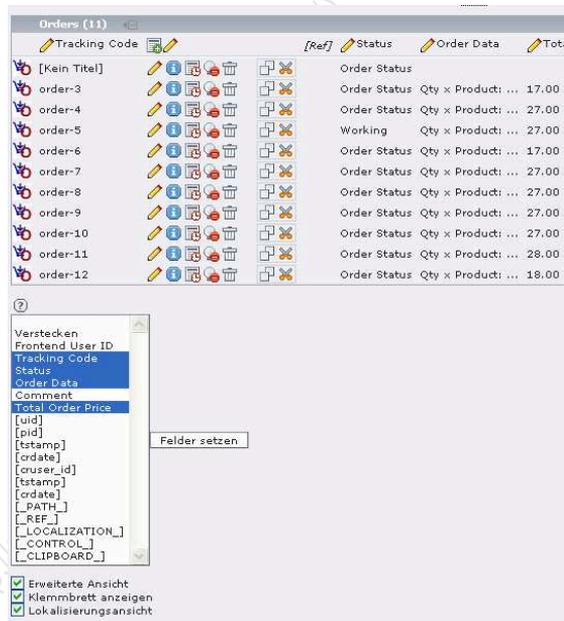
Abbildung 9.69:
Order Status
zuweisen (Trade)

In Abbildung 9.69 sehen Sie die Bestellung und können den Status weiter ändern. Im Testshop wurde das HTML-Templete nicht angepasst, wie Sie im Editorfeld sehen. Dort werden die Bestellinformationen angezeigt. Über die E-Mail-Adresse wird die Bestellung dem Kunden zugeordnet.

Wenn Sie nun die Tabelle aller Bestellungen durch Klick auf das Pluszeichen anklicken und z. B. die Felder **Tracking Code**, **Status**, **Order Data**, **Total Order Price** über Felder setzen hinzufügen (Abbildung 9.70), können Sie die Bestelltabelle exportieren. Wählen Sie dazu das Icon für den CSV-Import aus, speichern Sie die Daten und exportieren Sie z. B. ins Excel-Format. Sie

können natürlich beliebige Felder der Bestelltabelle verwenden, um noch ausführlichere Ergebnisse zu erhalten.

Abbildung 9.70:
Bestellungen im
Webmodul Liste
(Trade)



In Excel können Sie die Daten wie in Abbildung 9.71 darstellen und weiterverarbeiten, indem Sie z. B. nur die Datensätze summieren, die den Orderstatus mit der UID 3 besitzen und damit als abgeschlossen gelten.

Abbildung 9.71:
Tabelle der
Bestellungen in Excel
(Trade)

| | A | B | C | D | E | F | G |
|----|---------------|----------|------------|------|--------|------------|-----------------|
| 1 | tracking_code | CONTROL_ | CLIPBOARD_ | REF_ | status | order_data | price_total tax |
| 2 | | | | | | | |
| 3 | | | | | 1 | | |
| 4 | order-3 | | | | 1 | | 17.00 |
| 5 | order-4 | | | | 1 | | 27.00 |
| 6 | order-5 | | | | 3 | | 27.00 |
| 7 | order-6 | | | | 1 | | 17.00 |
| 8 | order-7 | | | | 1 | | 27.00 |
| 9 | order-8 | | | | 1 | | 27.00 |
| 10 | order-9 | | | | 1 | | 27.00 |
| 11 | order-10 | | | | 1 | | 27.00 |
| 12 | order-11 | | | | 1 | | 28.00 |
| 13 | order-12 | | | | 1 | | 18.00 |
| 14 | | | | | | | |
| 15 | | | | | | | |

Auch für Trade sind in naher Zukunft einige neue Features geplant; so sollen in nächsten Releases folgenden Funktionalitäten implementiert sein:

- realURL-Unterstützung
- Vereinfachung der Templates für die Konfiguration
- Möglichkeit, multiple Plugins auf einer Seite einzufügen und dann die entsprechenden Seitenfunktionen wie Warenkorb, Listenansicht etc. über ein Code-Feld auszuwählen
- jquery-Features – AJAX-Anweisungen, flyouts etc.
- Nutzen der TS-Constants für allgemeine Zuweisungen wie HTML-Template, E-Mail-Adressen etc.
- Adresdatensätze nutzen, um Kunden-Details zu speichern; mehrere Adressen pro Frontendbenutzer sollen erlaubt sein.
- Integration der static_info_tables zur Länderauswahl
- diverse Lieferpreis-Berechnungen: nach Gewicht, Sperrgut, Zusatzkosten etc.
- verschiedene Preisoptionen nach Kategorien, Rabatt nach Anzahl, Benutzergruppen etc.

9.5 Webformat Shop System

Dieser Abschnitt widmet sich einer einfachen Installation des Webformat-Shops von Mauro Lorenzutti.¹⁴ Das Shopsystem ist auch unter dem Namen Extended Shop bekannt. Eine Dokumentation in englischer Sprache liegt bereits vor. Dieser Shop verfügt über eine Listen- und Einzelansicht, alle Warenkorbfunktionalitäten sowie eine Bestellverwaltung und ist mehrsprachig einsetzbar.

Der Shop soll in der Version 3.0 über die folgenden, zusätzlichen Features verfügen:

- Unterstützung von realUrl
- Sicherheitsoptimierung gegen SQL-Injection
- viele Hooks zur Erweiterung des Shops, z. B. neue Felder ohne den Quellcode zu modifizieren

¹⁴ <http://www.webformat.com/>

- Implementierung der Kategorien mit der Extension `toi_category`
- Optimierung der Bezahlmethoden; Unterstützung der Payment-Gateways durch entsprechende Extensions. Zunächst werden PayPal und Banca Sella integriert, weitere sollen folgen.
- verbesserte Unterstützung der Mehrsprachigkeit
- Lagerverwaltung
- Verwendung mehrerer Bilder je Produkt, auch in verschiedenen Größen
- neue Felder für das Produkt wie verwandte Dokumente, verwandte Seiten etc.
- Herstellerverwaltung
- Integration der Extension `sr_feuser_registration` und der TYPO3-Authentifizierung
- Detailseiten zum Produkt ähnlich den Detailseiten von `tt_news`
- Verwendung von Benutzergruppen- und Benutzerrabatten
- Miniwarenkorb, der über TypoScript eingebunden wird
- Verbesserung der Versandmethoden; es wird nicht mehr notwendig sein, die Methoden und Preise über TypoScript zu definieren. Für jedes Land können die Versandmethoden und -kosten als separate Eigenschaften auch von Nichtadministratoren (**Neuen Datensatz anlegen**) definiert werden.

Installieren Sie die Extension wie gewohnt über den Extension Manager. Die Extension finden Sie unter dem Extension Key `extendedshop` im TER. Die Extension setzt die Erweiterung `tt_address` voraus. Die Installation des Shops in der Version 2.0.7 (Stable) erfolgte im Beispiel unter TYPO3 4.2, die Extension warnt aber, dass sie nur bis Version 4.0.5 kompatibel sei. Gleiches gilt für die verwendete PHP-Version 5.1.2; die Extension laufe nur bis PHP Version 5.0.5. Diese Informationen haben wir in unserem Test ignoriert und dennoch erfolgreich installieren können.

9.5.1 Shopseiten erstellen und Plugin einfügen

Erstellen Sie zunächst eine Seite für die Listenansicht des Shops namens **Webformat Shop** und fügen Sie das Plugin **Webformat Shop System** ein.



Abbildung 9.72:
Plugin auswählen
(Webformat)

Listenansicht

Die Hauptseite Ihres Shops soll der Darstellung der Listenansicht dienen. Wählen Sie aus der Combobox **Products List** aus. Für diesen Shop muss ein SysOrdner mit Produkten als Ausgangspunkt gesetzt werden. Den Ausgangspunkt definieren Sie, sobald Sie einen SysOrdner für die Produkte erstellt haben.

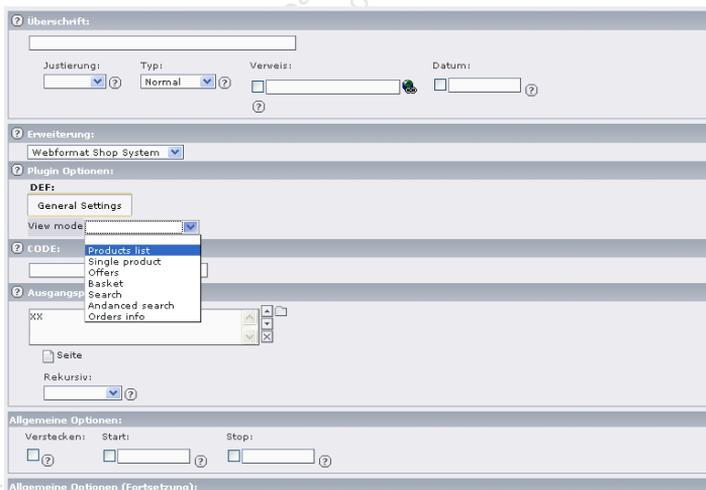


Abbildung 9.73:
Plugin für die
Listenansicht
zuweisen
(Webformat)

Angebote

Erstellen Sie eine weitere Seite für die Angebote Ihres Shops und fügen Sie das Plugin ein. Wählen Sie aus der Combobox **Offers** aus und speichern Sie den Datensatz. Auf dieser Seite stellt der Shop später Ihre Sonderangebote dar.

Suche

Für die Seite **Suche** fügen Sie ebenfalls das Plugin ein und weisen über die Combobox **Search** die Suche zu.

Warenkorb

Die letzte Seite, die zu erstellen ist, ist die Seite für den **Warenkorb**. Fügen Sie wieder das Plugin ein und weisen Sie über die Combobox den **Basket** zu.

9.5.2 SysOrdner

Webformat verwaltet Produkte, Adressen und Bestellungen in SysOrdnern.

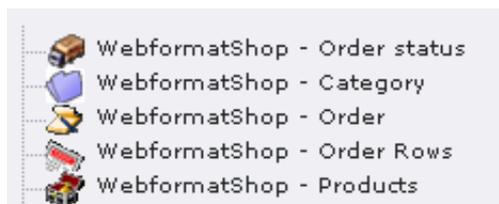
Erstellen Sie zunächst einen SysOrdner für Ihre Bestellungen, der später alle Bestellungen für die Bestellverwaltung zur Verfügung stellen wird. Ein weiterer SysOrdner ist für die Lieferanschriften der Kunden zu erstellen. In diesem Ordner wird die Lieferadresse des neuen Kunden bei seiner ersten Bestellung gespeichert.

Legen Sie nun einen dritten SysOrdner namens **Produkte** für Ihre Produkte an und weisen Sie diesen SysOrdner den eben erstellten Shopseiten für die Listenansicht, die Angebotsansicht, die Warenkorbanzeige und die Suche zu.

9.5.3 Produkte

In Ihrem Produkte-SysOrdner erstellen Sie jetzt Ihre Produkte. Klicken Sie auf den Link **neuen Datensatz anlegen** und wählen Sie **Webformatshop - Products** aus.

Abbildung 9.74:
SysOrdner: neuen
Datensatz anlegen
(Webformat)



Das Backendformular für das Produkt stellt folgende Felder zur Verfügung:

Itemnumber

Artikelnummer

Title

Produkttitel

Page Title

Detailseite des Produkts. Wenn dieses Feld leer ist, wird der Produkttitel verwendet. Diese Funktion steht nur zur Verfügung, wenn die entsprechende TS-Konfiguration vorliegt.

Summary

Textfeld für eine kurze Zusammenfassung

Description

Editorfeld für ausführliche Produktinformationen.

Image

Sie können diesem Produkt wie gewohnt ein Bild hinzufügen. Auch dieses Shopsystem bietet keine Möglichkeit, Alternativtexte für Bilder darzustellen.

Price

Preis des Produktes. Der Shop verwendet in den Default-Einstellungen den Dezimalpunkt. Hier gibt es ein kleines Problem: Bei Preisen wie 10.50 müssen Sie 10.5 setzen, sonst erscheint eine Fehlermeldung.

Price no tax, In Stock, WWW, Ordered, Weight, Volume, Correlated

Accessories und Offert price no tax

stehen derzeit noch nicht zur Verfügung. Die Kategorie-Behandlung ist laut Auskunft des Autors ebenfalls noch nicht komplettiert.

Offer price

gibt den Angebotspreis an. Diese Werte definieren die Sonderangebote. Wenn dieser Wert gesetzt ist, wird der Prozentsatz für den Discount-Preis automatisch kalkuliert. Der Discountpreis wird in % angegeben. Diese Funktionalität steht nur zur Verfügung, wenn dem Feld **Offer** kein fixer Preis zugeordnet wurde.

Sizes und Colors

Größe und Farbe des Produkts ein. Wenn hier mehr als ein Wert eingetragen werden soll, trennen Sie die Werte mit einem Ausrufezeichen. Mit dieser Funktion werden die Varianten des Produkts, also Artikel erstellt.

Correlated products

verweist auf verwandte oder ähnliche Produkte. Sie können die entsprechenden Produkte mit **durch Datei browsen** auswählen und diesem Produkt zuweisen.

Abbildung 9.75:
Backendformular des
Produktes
(Webformat)

The screenshot shows the TYPO3 backend interface for editing a product. The page title is 'Webformatshop - Producte [1] - 123456'. The breadcrumb trail is 'Webformatshop - Producte'. The form is organized into several sections:

- Versteckten:** Contains 'Start:' and 'Stop:' checkboxes, both currently unchecked.
- Language:** A dropdown menu is set to 'Standard'. Below it is a 'Itemnummer:' field containing the value '123456'.
- Title:** A text field containing 'Sweatshirt', with 'Page Title' and 'Sweatshirt' listed below it.
- Summary:** A text area containing 'Eine Zusammenfassung'.
- Description:** A rich text editor with a toolbar (bold, italic, underline, list, link, unlink, image, table, undo, redo) and a text area containing 'Die Produktinformationen'.
- Image:** A file manager showing a selected file 'gif.jpg' with other formats listed: 'JPEg', 'TIFF', 'BMP', 'PCX', 'TGA', 'PNG', 'PDF', 'AI'. A 'Durchsuchen...' button is present.
- Price:** A text field with '10', and a 'Price No Tax' field with '0'.
- In Stock:** A checkbox that is currently checked.
- Kategorie:** A dropdown menu with 'WYWIW' selected.
- WYWIW:** A text field.
- Ordered:** A text field.
- Weight:** A text field with '200'.
- Volume:** A text field with '0'.
- offset price:** A text field with '0', and a 'Discount' field with '0'.
- Offert price no tax:** A text field with '0'.
- Sizes:** A text field.
- Colors:** A text field.
- Comrelated Products:** A section with a 'Comrelated Products' label and a list of related products, currently empty.

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://www.typo3.com>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.com>)

9.5.4 Shop mit registrierten Benutzern

Wenn Sie mit registrierten Benutzern arbeiten, können die Benutzerdaten aus der Frontendbenutzerregistrierung verwendet werden. Die Lieferadressen werden, wie bereits am Anfang erwähnt, in der Extension `tt_address` gespeichert und im SysOrdner **Adressen** verwaltet. Wird ein neuer Benutzer angelegt, werden die allgemeinen Daten und die Rechnungsanschrift des Kunden in der Frontendbenutzerregistrierung gespeichert, die Lieferadresse aber in `tt_address`, da die Frontendbenutzerregistrierung eigentlich keine zwei Adressen verwalten kann – es sei denn, sie wird durch eine Extension erweitert, wie es beim Shopsystem Trade der Fall ist. Bei der ersten Bestellung erhält der neu angelegte Kunde eine separate E-Mail mit seinen Zugangsdaten. Bei seinem nächsten Login werden die Adressdaten dann automatisch eingelesen.

9.5.5 TS-Template

Erstellen Sie ein Extension-Template namens `+extended shop` im Template-SysOrdner und nehmen Sie dort Ihre Shop-Konfiguration vor. Ein statisches Template besitzt der Shop nicht. Weisen Sie das Extension-Template in gewohnter Weise dem Haupt-Template zu. In diesem einfachen Beispiel sind in den Constants lediglich die PIDs, das HTML-Template mit CSS und die Dezimalzeichen zugeordnet. Über den Constant Editor können Sie weitere Konfigurationen vornehmen

Constants

```
# PIDs zuweisen
plugin.tx_extendedshop_pi1.pidBasket = 326
plugin.tx_extendedshop_pi1.pidOrders = 327
plugin.tx_extendedshop_pi1.pidUsers = 52
plugin.tx_extendedshop_pi1.pidDelivery = 328

# Default HTML-Template und CSS-Datei zuweisen
plugin.tx_extendedshop_pi1.file.templateFile = EXT:extendedshop/pi1/shop.html
plugin.tx_extendedshop_pi1.file.cssFile = EXT:extendedshop/pi1/styleShop.css

# Dezimalzeichen zuweisen
plugin.tx_extendedshop_pi1.priceDec = 2
plugin.tx_extendedshop_pi1.priceDecPoint = ,
plugin.tx_extendedshop_pi1.priceThousandPoint = .
```

`pidBasket` ist die der Warenkorbseite, `pidOrders` definiert den SysOrdner für die Bestellungen, `pidUsers` verweist auf den SysOrdner mit den Front-

endbenutzern und `pidDelivery` definiert die PID des SysOrdnerns, in dem die Lieferadressen gespeichert werden.

Im Frontend können Sie jetzt Ihre ersten Testbestellungen durchführen.

Setup

Das Setup wurde dem Beispiel-Setup der Extension entnommen und in einigen Punkten angepasst. Im folgenden Code werden zunächst das Template und die CSS-Datei über Variablen zugewiesen. Die eigentlichen Einstellungen wurden in den Constants vorgenommen. Die Rekursivität wurde gesetzt und die Pflichtfelder des Adressformulars definiert. Alle weiteren Instanzen des Setup finden Sie im Codebeispiel in den Kommentaren erklärt.

```
includeLibs.tx_extendedshop_pil =
    EXT:extendedshop/pil/class.tx_extendedshop_pil.php

plugin.tx_extendedshop_pil= USER
plugin.tx_extendedshop_pil.userFunc = tx_extendedshop_pil->main

plugin.tx_extendedshop_pil{

    # The main function
    userFunc = tx_extendedshop_pil->main

    # Template und CSS zuweisen
    templateFile = {$plugin.tx_extendedshop_pil.file.templateFile}
    cssFile = {$plugin.tx_extendedshop_pil.file.cssFile}

    recursive.field = recursive

    # Suchfelder
    searchFields = code,title

    # Felder des Adressformulars
    requiredFields =name,address,city,zip,state,phone,email,authorization,
conditions
    requiredFieldsSymbol = *

    # Keine Lieferkosten ab diesem Bestellwert
    freeDelivery = 129,00

    # Mindestbestellwert ab
    minOrder = 5,00

    # E-mail Adressen des Shops
    orderEmail_from = tester@mytypo3domain.de
```

```

orderEmail_fromName = mytypo3domain.de

# BCC Empfänger der Bestätigungs E-mail
orderEmail_to = tester@mytypo3domain.de

# HTML Mails disabled
orderEmail_htmlmail = 0
email_notify_default = 1
}

# Produkttitel im Seitentitel anzeigen
config.titleTagFunction = tx_extendedshop_pil->product_title

```

Bezahlung und Lieferung

Wenden Sie sich nun den Methoden für Bezahlung und Lieferung zu. In diesem einfachen Shop soll zunächst die Bezahlung gegen Rechnung ermöglicht werden. Für diese Bezahlmethode wurde lediglich ein Titel vergeben, zusätzliche Kosten fallen nicht an. Die zweite Bezahlmethode soll die Nachnahme sein. Dafür werden zusätzlich EUR 4,- Gebühren berechnet. Für die Bezahlung über PayPal müssen die entsprechenden Zuweisungen erfolgen. Weitere Informationen zur PayPal-Bezahlung finden Sie im Kapitel 6.

Bei den Liefermethoden wurde die Zahlung in die Länder Deutschland, Frankreich und Italien erlaubt und die entsprechenden Versandkosten definiert.

```

## Bezahlmethoden
plugin.tx_extendedshop_pil.payment {
    # Auswahl über Radiobuttons
    radio = 1

    10.title = Rechnung
    10.image.file =
    10.priceTax =
    10.priceNoTax =

    20.title = Nachnahme
    20.image.file =
    20.priceTax = 4,00
    20.priceNoTax =

    50.title = PayPal
    50.bankcode = paypal
    50.UICCODE = EUR
    50.ShopLogin = myPayPalLogin
    50.bankname = PayPal

```

```
50.banklink = www.paypal.com
50.return = http://www.mysite.com/index.php?id=11
50.paylink = https://www.paypal.com/cgi-bin/webscr?cmd=_xclick&upload=1

}

## Liefermethoden
plugin.tx_extendedshop_pil.shipping {

    10.title = Deutschland
    10.priceTax = 4,50

    20.title = Italien
    20.priceTax = 8,00

    30.title = Frankreich
    30.priceTax = 8,00

}
```

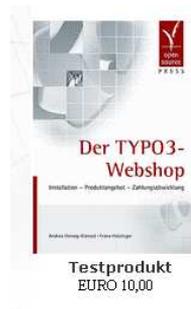
Ergänzen Sie das Setup um obigen Code für die Bezahl- und Liefermethoden. Diese Konfiguration soll für den Beispielshop genügen.

9.5.6 Der Shop im Frontend

Wie bei den anderen Shop-Extensions, die in diesem Kapitel beschrieben sind, wurden auch bei Webformat keinerlei Anpassungen an der HTML- und der CSS-Datei vorgenommen. Die Bilder stellen die Voreinstellungen des Shops dar. Natürlich können Sie den Shop an Ihre Bedürfnisse anpassen.

In Abbildung 9.76 sehen Sie die Listenansicht des Shops. Da der Shop lediglich aus einem Testprodukt besteht, ist hier nicht viel zu sehen.

Abbildung 9.76:
Listenansicht im
Frontend
(Webformat)



Lediglich das Bild, der Preis und der Produkttitel werden dargestellt. Sofern Sie verschiedene Produkte angelegt und die entsprechende Konfiguration

im Constant Editor gewählt haben, können Sie die Produkte auch neben- und untereinander darstellen. Die Anzahl der Produkte, die pro Seite angezeigt werden dürfen, können ebenfalls über den Constant Editor definiert werden.

Ein Klick auf den Titel leitet zur Einzelansicht (Abbildung 9.77). Diese verfügt in der Standardkonfiguration über eine Warenkorbfunktion und zeigt die Artikelnummer sowie die hinterlegten Produktinformationen an. Ein Link zurück zur Listenansicht ist nicht voreingestellt.



Abbildung 9.77:
Detailansicht im
Frontend
(Webformat)

Sobald Sie einen Artikel in den Warenkorb gelegt haben leiten sowohl der grafische wie auch der Textbutton in den Warenkorb (Abbildung 9.78).



Abbildung 9.78:
Warenkorbansicht
(Webformat)

In dieser Ansicht können Sie den Warenkorb aktualisieren oder leeren.

Durch einen Klick auf **Weiter** wechseln Sie zur Adressdateneingabe. Da der Kunde in diesem Fall bereits ein registrierter Benutzer ist und bei einer vorhergehenden Bestellung seine Lieferadresse angegeben hat, werden seine Adressdaten automatisch in die Eingabefelder übernommen.

In dieser Ansicht müssen die Checkboxes für die AGB und die Einverständniserklärung für die Speicherung der Adressdaten aktiviert werden. Sobald der Kunde seine Angaben über den Button bestätigt hat, wechselt der Shop zur Auswahl der Bezahl- und Liefermethoden (Abbildung 9.79).

Abbildung 9.79:
Bezahl- und
Liefermethode
auswählen
(Webformat)

Versandkosten

Deutschland Preis: EURO 4,00

Bezahlung

Rechnung
 Nachnahme
 PayPal
 Credit card with Banca Sella
 Credit card with Authorize.net

Preis: EURO 4,00

Ihre bestellten Artikel



Testprodukt
 Artikelnr. 11111
 Einzelpreis: EURO 10,00

Preis: EURO 10,00

Gesamtpreis: EURO 18,00

Ihre persönlichen Daten
 Bitte füllen Sie alle mit * markierten Felder aus.

| | |
|---------------------------------------|----------------------------------|
| Name und Vorname | Firma |
| Adresse Teststr. 1 | |
| Stadt Teststadt | Person oder Firma? Person |
| PLZ 11111 | |
| Landkreis | Steuernummer |
| Land | Mobil |
| Telefon | Fax |
| E-mail tester@mytypo3domain.de | Website |

Ich bin mit der Speicherung meiner Daten einverstanden.
 Ich akzeptiere die AGB.

Lieferadresse
 Hier können Sie eine abweichende Lieferadresse angeben.

| | |
|---------------------------------------|---------------------------------------|
| Name und Vorname tester tester | E-mail tester@mytypo3domain.de |
| Adresse Teststr. 1 | PLZ 11111 |
| | Stadt Teststadt |
| Landkreis Testkreis | Land Deutschland |
| Telefon 11111 | Mobil |

Kommentar

Im Bild wurde die Bezahlmethode Nachnahme und die Lieferung innerhalb Deutschlands ausgewählt. Dabei fallen zusätzliche Gebühren von insgesamt EUR 8,- an. Dieser Betrag setzt sich aus der Versandkostenpauschale und dem Preis für die Nachnahme zusammen.

Ein Klick auf den Link **Bestellung abschließen** löst zuletzt die Bestellung aus. Die gesamten Bestelldaten werden in einer Übersicht angezeigt. Diese Bestätigung kann vom Kunden ausgedruckt werden. Die hier angezeigten Informationen werden auch in der Bestellverwaltung, die im nächsten Abschnitt beschrieben wird, angezeigt.

9.5.7 Bestellungen

Wechseln Sie nun im Backend ins Webmodul **Liste** und den SysOrdner mit den Bestellungen, der nun die getätigten Testbestellungen enthält. Der Shop hat einen Unterordner **2007** generiert und darin einen Ordner **10** für die Bestellungen des Monats Oktober (Abbildung 9.81). Die einzelnen Bestellungen dieses Monats werden in diesem SysOrdner abgelegt.

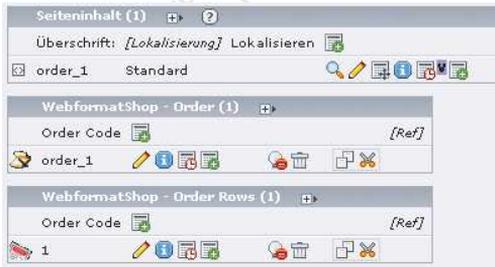


Abbildung 9.80:
Ordner Bestellungen
(Webformat)

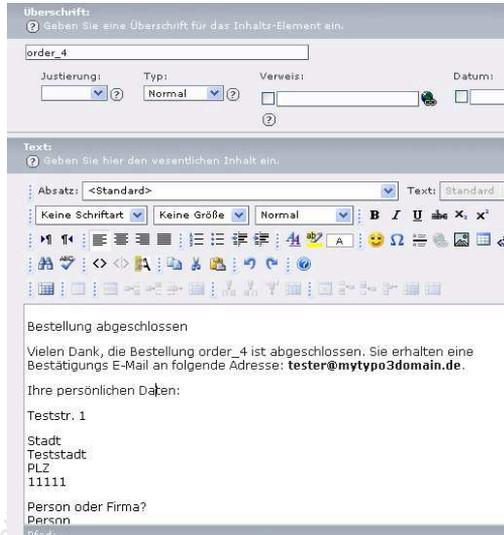


Abbildung 9.81:
Bestellungen
(Webformat)

Die Bestellungen werden in drei Ansichten gezeigt:

Die erste ist die Bestätigungs-E-Mail mit allen Daten der Bestellung (Abbildung 9.82). Sie sehen die vollständigen Daten, die in der Bestellbestätigung im Frontend angezeigt wurden.

Abbildung 9.82:
Bestellungen:
Bestätigungs E-Mail
(Webformat)



Die zweite Ansicht enthält die Bestelldaten, den Tracking-Code und die IP-Adresse des Kunden, die im Bild aus Gründen der Vertraulichkeit ausgegraut ist. Die Abbildungen 9.83 und 9.84 zeigen die Ansicht dieser Bestelldaten.

Abbildung 9.83:
Bestelldatenansicht
Order 1. Teil
(Webformat)

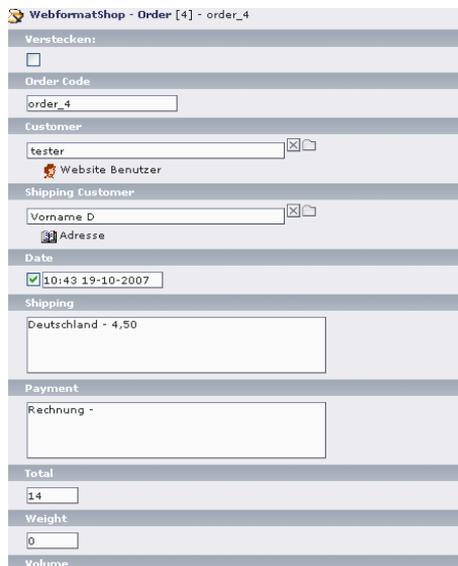


Abbildung 9.84:
Bestelldatenansicht
Order 2. Teil
(Webformat)

Die dritte Ansicht enthält das bestellte Produkt mit der Bestellnummer, der Artikelnummer, der Bestellmenge und dem Preis (Abbildung 9.85). Über phpMyAdmin können Sie aus diesen Datenbanktabellen statistisches Material ermitteln. In Zukunft soll Webformat jedoch über eine komplette Bestellverwaltung im TYPO3-Backend verfügen.

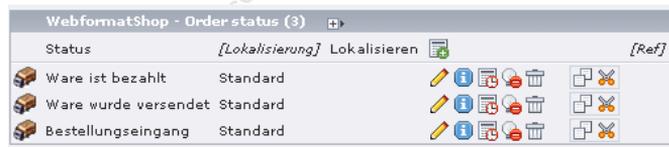
Abbildung 9.85:
Daten des bestellten
Produktes
(Webformat)

Statuscode

Obwohl, wie mehrfach erwähnt, die Bestellverwaltung noch nicht vollständig implementiert ist, können Sie Ihren Bestellungen bereits jetzt einen entsprechenden Bestellstatus zuweisen, wie bereits in den Shopsystemen Commerce und Trade beschrieben.

Wechseln Sie im Webmodul Liste in den SysOrdner Bestellungen und legen Sie mit **Neuen Datensatz anlegen** die Bestellstatus an. Dazu wählen Sie aus der Liste **Order status** aus und erstellen die entsprechenden Statusinformationen.

Abbildung 9.86:
Order-Status
erstellen
(Webformat)



Den entsprechenden Status können Sie dann der jeweiligen Bestellung über **Order** und dort die Combobox **Order status** zuweisen (Abbildung 9.87).

Abbildung 9.87:
Order-Status
zuweisen
(Webformat)



Trotz dieser Minimalkonfiguration wird deutlich, dass auch diese Shop-Extension über einige interessante Eigenschaften verfügt, und man darf gespannt bleiben, mit welchen Features neue Releases aufwarten werden.

Anhang

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/3.0/de/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

Übersicht TYPO3-Shopsysteme

Zum Abschluss eine kurze Übersicht über die in diesem Buch beschriebenen TYPO3-Shopsysteme. Wir haben dazu einen Fragebogen an die jeweiligen Entwickler versendet, deren Antworten Sie im Folgenden zusammengefasst finden.

| Feature | Com | GSA | Tr | tt | Web |
|--------------------------|------------|------------|-----------|-----------|------------|
| Verschiedene Steuersätze | • | • | – | • | – |
| Rabattsystem | – | • | • | • | • |
| Preisskalierung | • | • | • | • | –/• |
| Mehrsprachigkeit | • | • | – | • | • |
| Gutscheinsystem | – | – | – | • | – |
| Lieferdatum | – | • | – | • | • |
| Sonderangebote | • | • | • | • | – |

Tabelle A.1:
Features

Fortsetzung:

| Feature | Com | GSA | Tr | tt | Web |
|-------------------------------|-----|-----|----|----|-----|
| Alt- und Titeltex t f. Bilder | – | – | – | • | –/• |
| Download-Artikel | – | – | – | – | –/• |
| Tracking | – | – | • | • | • |
| Merzettelfunktion | – | – | – | • | – |
| Bestellverwaltung/-Management | • | • | • | • | • |
| Artikel-Abonnement | – | – | – | – | – |
| Permanenter Miniwarenkorb | • | • | • | • | –/• |
| Produkt-Kategorien | • | – | • | • | –/• |

Com = Commerce (0.9.3) / GSA = GSA Shop / Tr = Trade / tt = tt_products (2.6.0) / Web = Webformat Shop (2.0.7/3.0)

Welche Zielgruppe spricht das Shopsystem an?

Commerce

Größere Firmen mit einer sehr variablen Artikelstruktur auf der Suche nach einer flexiblen Lösung, die an die eigenen Bedürfnisse angepasst werden kann; leicht durch Hooks anzupassen.

GSA-Shop

Unternehmen, technisch versierte Administratoren (komplexe Installation).

Trade

TYPO3-Anfänger und alle TYPO3-Nutzer, die eine schnell integrierbare Shoplösung benötigen.

tt_products (2.6.0)

Kleinere Unternehmen.

Webformat Shop (2.0.7/3.0)

Kleine und mittelständische Unternehmen, die eine schnell integrierbare Shoplösung benötigen.

Gibt es eine empfohlene Begrenzung für die Anzahl der Artikel im Shop?

Commerce

Keine Limitierung; es wurden bereits Shops mit mehr als 100.000 Artikeln umgesetzt.

GSA Shop

Bzgl. Datenverwaltung keine, bzgl. Handling im TYPO3-Backend (Pull-Down-Menü) etwas unübersichtlich.

Trade

Keine Einschränkungen.

tt_products 2.6.0

Keine Einschränkungen (empfohlen bis ca. 5000 Artikel).

Webformat Shop (2.0.7/3.0)

Keine Einschränkungen.

Einsatzschwerpunkte des Shopsystems

Commerce

ERP_Kopplung, Artikel-Attribute, Order-Handling, Mehrsprachigkeit.

GSA Shop

Professioneller Einsatz mit WaWi-Anbindung und automatisierter Abwicklung kompletter (digitaler) Geschäftsprozesse im Zusammenspiel mit weiteren Extensions.

Trade

Einfache Installation.

tt_products 2.6.0

Kleine Shops mit wenigen Produkten, aber diversen Produktvarianten und vielen Bildern.

Webformat Shop (2.0.7/3.0)

Einfache Installation, volle Integration in TYPO3-Extensions zur Benutzerverwaltung, einfache Erweiterbarkeit.

Besondere Highlights

Commerce

Leicht erweiterbar und an eigene Bedürfnisse anzupassen; Gutscheine und Rabattsystem sind derzeit in Arbeit.

GSA Shop

Automatisierte Abwicklung kompletter Geschäftsprozesse, nahtlose Integration in WWS/ERP GS-Auftrag Professional, präzise Fließkommaabrechnung (bis 4 Nachkommastellen), kundenspezifische Preise (pro Kunde und Artikel), Workflow-Engine und Bestellungen-Archiv (reusable Orders).

Trade

Flexform-basierte Basiskonfiguration, einfache HTML-Anpassungen.

tt_products 2.6.0

Kreditpunkte und Gutscheinsystem, Tracking, Mehrsprachigkeit, Bildershops, Categoriesystem.

Webformat Shop (2.0.7/3.0)

Erweitertes Bestellmanagement (3.0), Integration von Payment-Gateways, anspruchsvolle Versandoptionen, Rabattsystem.

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/3.0/de/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

B

Anhang

Der Extension Manager

Obwohl der Extension Manager für Aufbau und Pflege eines TYPO3-Systems ein zentrales Werkzeug ist, bereitet dessen Bedienungskonzept gerade Einsteigern oft Schwierigkeiten. Da er auch für die Einrichtung des Webshops unentbehrlich ist, wollen wir uns das Tool in diesem Zusammenhang ein wenig genauer anschauen.

B.1 Sichtbarkeit von Extensions

Sie sollten im Extension Manager zunächst den Haken bei `Display shy extensions` setzen, damit *alle* in Ihrem System installierten Extensions unter der Auswahl `Loaded extensions` erscheinen. „Shy extensions“ sind solche, die zum Basissystem gehören und vom Administrator in der Regel nicht wieder „angefasst“ werden (müssen). Der Übersicht halber lassen sie sich darum von der Anzeige ausnehmen.

Im Menü **Settings** sollten Sie zudem den Haken bei **Display unsupported extensions** oder **Enable extensions without review (basic security check)** setzen. Andernfalls finden Sie viele Extensions im Repository (TER) einfach nicht und können sie somit auch nicht installieren. Allerdings birgt die Installation einer Extension auch eine gewisse Gefahr; beispielsweise könnten sich bei einer Sicherheitslücke Unbefugte über SQL-Injection bei MySQL einen Backend-Admin-Zugang zur TYPO3-Installation verschaffen.

Seit dem 15.02.06 gibt es einige wichtige Änderungen im sog. TER2: So werden ins offizielle Repository nur solche Extensions aufgenommen, die einen einfachen Sicherheitscheck durchlaufen haben, der von jeweils zwei Mitgliedern des Security-Teams durchgeführt wird. Sobald eine Extension ins TER geladen wird, werden die Mitglieder des Security-Teams informiert. Im Laufe der Zeit versucht das Team die Extensions zu sichten und als „security checked“ zu bewerten. Dieser Prozess ist bei der Flut vorhandener und neu hinzukommender Extensions natürlich sehr aufwendig und nimmt viel Zeit in Anspruch. Bis dahin sind diese Extensions – und das sind derzeit fast alle – als „unsupported“ oder ohne Review deklariert. Allerdings haben auch zukünftig nur solche Extensions eine Chance ins offizielle TER aufgenommen zu werden, die mindestens den BETA-Status erreicht haben.

Ein Grund für dieses Vorgehen ist die generelle Strategie, TYPO3-Anwender auf mögliche Sicherheitsrisiken aufmerksam zu machen, denn TYPO3-Extensions setzen Sie stets auf eigene Gefahr ein. Melden Sie Sicherheitslücken sofort an das TYPO3-Security-Team – keinesfalls sollten Sie sich darüber zuvor in öffentlichen Foren und Mailinglisten austauschen, damit Sie potentiellen Eindringlingen keine wertvollen Tipps geben. Abonnieren Sie zudem die TYPO3-Security-Mailingliste, damit Sie sofort über entdeckte Lücken informiert werden und gegebenenfalls rasch reagieren können.

B.2 Sprachauswahl

Auch die **Translation Settings** werden im EM gesetzt. Wählen Sie unter **Translation Handling** die Sprachen Ihrer Webseite. Wenn Sie künftig Extensions in verschiedenen Sprachen einsetzen, müssen Sie diese gegenüber dem Repository updaten, damit die Übersetzungen auch funktionieren. Installieren Sie eine Extension nach, müssen Sie das Update erneut durchführen.

Bei Fehlern oder schlechten Formulierungen in anderen Sprachen sollten Sie die entsprechende Projektseite besuchen.¹ Der Autor einer Extension hinterlegt selbst evtl. nur englische Texte, und für jede andere Sprache sind freiwillige Übersetzer gefragt. Übersetzungsfehler bzw. -vorschläge sollten Sie im Bugtracker² unter **Translation & Internationalization** melden.

¹ <http://translation.typo3.org/translation/>

² http://bugs.typo3.org/my_view_page.php

B.3 Editieren von Extension-Dateien

Im Verlaufe dieses Buches wurden einige ursprüngliche Dateien der Extension `tt_products` verändert und gespeichert. Beim ersten Versuch dieser Art über den Extension Manager erschien unter Umständen die folgende Meldung:

```
[SAVING IS DISABLED - can be enabled by the
$TYPO3_CONF_VARS['EXT']['noEdit']-flag]
```

In diesem Fall ist also über das Installtool zunächst die grundsätzliche Möglichkeit zur Manipulation dieser Dateien zu geben. Entfernen Sie dazu unter **All Configuration** das Häkchen bei `[noEdit]`, um die Schreibrechte für die Extension-Dateien zu erhalten. Eventuell muss anschließend der Konfigurationscache gelöscht werden, um die Änderung wirksam zu machen.³

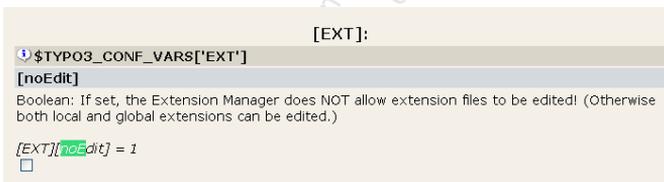


Abbildung B.1:
Editieren erlauben

B.4 Backup und Update

B.4.1 Versionen von Extensions

Falls Sie eine neuere Version einer bereits installierten Extension über den Extension Manager aus dem TER laden möchten oder sich nur informieren wollen, ob es bereits Updates gibt, kann es sein, dass manche der neuesten Versionen gar nicht gefunden werden. Damit Ihnen diese Enttäuschung erspart bleibt, können Sie wie folgt vorgehen.

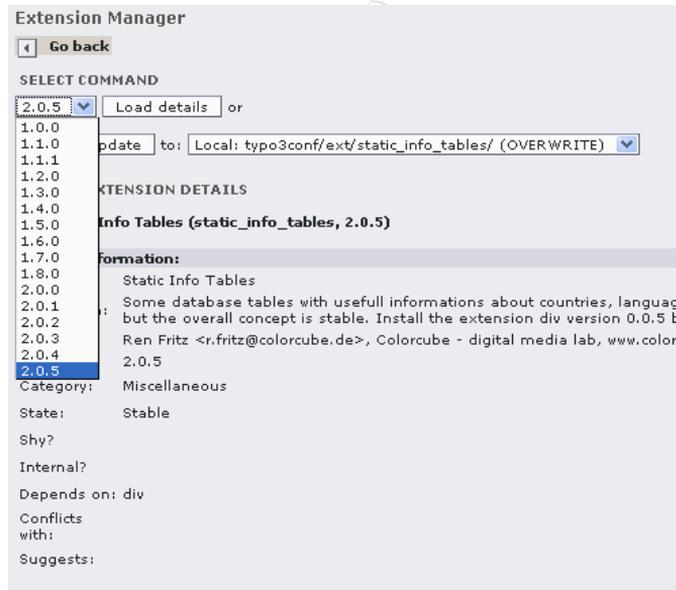
Wählen Sie zunächst **Import Extensions** aus, klicken Sie auf **Retrieve/Update** und aktualisieren Sie damit das TER für Ihr System. Hinter diesem Button erscheint das Datum des letzten Updates gegenüber dem TER.

Nun tragen Sie den Namen der Extension in das Textfeld ein, z. B. `static_info_tables`. Klicken Sie auf den **Look up**-Button, erhalten Sie eine Liste aller Extensions, in deren Namen `static_info_tables` vorkommt. Die

³ Die Lese-/Schreibrechte auf Betriebssystemebene müssen natürlich ebenfalls korrekt gesetzt sein, was aber der Fall sein sollte. Treten nach der soeben beschriebenen Prozedur noch Probleme auf, müssten Sie den Systemadministrator Ihres TYPO3-Servers zu Rate ziehen.

verfügbaren Versionen einer Extension ermitteln Sie per Klick auf deren Namen. Erst hier tauchen dann auch die neuesten Versionen auf.

Abbildung B.2:
Versionen einer
Extension auflisten



Nehmen wir an, es gibt eine neuere Version einer in Ihrer Webseite installierten Shop-Extension. Deren Features möchten Sie natürlich nutzen und darum updaten. Für den Fall der Fälle sollten Sie zunächst die laufende Version sichern. Im Extension Manager klicken Sie dazu die Extension an, um sie auszuwählen, und wählen in der daraufhin in der oberen rechten Ecke erscheinenden Combobox **Backup/Delete**.

Abbildung B.3:
Backup/Delete



Über den Link **Download extension "tt_products" as a file** im Bereich **Backup** speichern Sie die Extension lokal als .t3x-Datei. Danach installieren Sie die neue Version; sollte diese nicht zufriedenstellend funktionieren, können Sie das Backup jederzeit wieder einpflegen, auch in einer anderen TYPO3-Umgebung.

Ein erneuter Upload einer Extension mit derselben Versionsnummer kann beispielsweise notwendig sein, wenn ein schwerwiegender Fehler in der Extension zu spät bemerkt und die Extension mit einem Patch wieder funk-

tionsfähig gemacht wurde, ohne dass darum eine neue Version ins TER hochgeladen wurde.

B.4.2 MD5-Fehler

Beim Download einer Extension aus dem TER kann es zu einer Fehlermeldung in Bezug auf die MD5-Prüfsumme kommen.⁴ Um das zu beheben, wählen Sie im Extension Manager wieder den Menüpunkt **Import extensions** und klicken den Button **Retrieve/Update**. Dadurch wird die Datei mit den möglichen TYPO3-Extensions wieder mit jenen aus dem TER abgeglichen. Durch einen neuen Upload einer Extension könnte sich inzwischen die MD5-Summe geändert haben. Der Extension Manager bemerkt das aber nicht und verwendet beim Vergleich der Extensions noch die alte Prüfsumme.

B.4.3 Installation als .t3x-Datei

Wenn Sie ein Backup einer Extension zurückspielen oder eine Extension aus dem TER direkt als .t3x-Datei installieren möchten, wählen Sie in der Combobox **Menü** des Extension Managers **Import extensions**. Der Zugriff auf Extensions erfolgt dann nicht nur in gewohnter Weise über das Repository, sondern weiter unten auch auf in Ihrem System befindliche .t3x-Dateien.

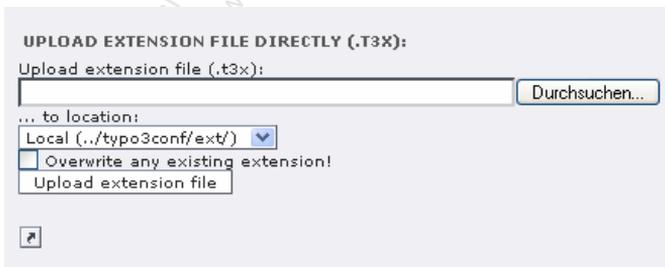


Abbildung B.4:
Upload

Vergessen Sie bei einem Update nicht, den Haken bei **Overwrite any existing extension!** zu setzen – damit ist natürlich lediglich die bereits installierte Version derselben Extension gemeint. Folgen Sie anschließend dem Dialog des Extension Managers, um die Installation abzuschließen.

Sollten Sie eine Datei der Extension editiert haben (wie zum Beispiel im Verlauf der Arbeit an dem Beispielshop), so wären diesen Änderungen jetzt allerdings verloren. Am besten notieren Sie sich irgendwo – am einfachsten

⁴ Siehe dazu im Bug-Tracker: <http://bugs.typo3.org/view.php?id=6269> Fehler: Error: MD5 hash of downloaded file not as expected

am Anfang des Template-Setups zur Extension –, dass es Änderungen am Programmcode gegeben hat, z. B. in Form von Kommentaren wie

```
### tt_products: Datei class.tx_ttproducts_order_view.php bearbeitet
```

Oder legen Sie sich eigens eine Seite im Seitenbaum zu Dokumentationszwecken dieser Art an. Damit wissen auch andere Personen, worauf in Ihrer TYPO3-Umgebung zu achten ist.

B.5 Fehlermeldung

Wenn Sie die Extension `tt_products` im Extension Manager anklicken und die Informationen dazu anzeigen, sehen Sie im unteren Bereich eine Fehlermeldung bzw. Warnung:

```
No XCLASS inclusion code found in file "lib/class.tx_ttproducts_email_div.php"  
No XCLASS inclusion code found in file "lib/class.tx_ttproducts_form_div.php"
```

Diese Warnungen können Sie getrost ignorieren, denn es handelt sich lediglich um den Vorschlag an die Entwickler, XCLASSES zu verwenden, wie wenig sinnvoll das auch im Einzelfall sein mag. Eine solche Meldung hat jedenfalls keinerlei negative Auswirkungen auf die Funktionsweise einer Extension.

B.6 static_info_tables

Wenn Sie die Static Info Tables verwenden, die Sie z. B. für die Länder(gruppen)auswahl im Shop benötigen, müssen Sie zunächst die Extensions `lib` und `div`, dann die `static_info_tables` und die `static_info_tables_de` (in dieser Reihenfolge!) installieren. Ab `static_info_tables` Version 2.0.6 werden `lib` und `div` nicht mehr benötigt. Am besten setzen Sie hier stets die aktuellsten Versionen ein, weil Fehlerkorrekturen häufig sind.

Trotz erfolgreicher Installation kann es vorkommen, dass die Tabellen leer sind. Klicken Sie also zur Sicherheit die `static_info_tables` im Extension Manager noch einmal an und wählen Sie aus der oberen rechten Combobox **UPDATE!** (nicht zu verwechseln mit dem regulären Update-Button bei der Installation). Anschließend wählen Sie den von Ihnen bevorzugten ISO-Code; leider bietet die Extension kein UTF-8 an. Beachten Sie bitte die Hinweise bei der Installation.

B.7 Erweiterung des Seiten-Moduls

Es gibt eine Erweiterung des Seiten-Moduls im TYPO3-Backend, die die Darstellung der Produktdatensätze übersichtlicher macht. Sie sorgt ebenfalls dafür, dass die Datensätze der Kategorien, alternative Sprachen und künftig noch weitere Informationen angezeigt werden. Dazu muss der Code des TYPO3-Kerns erweitert werden; den Patch finden Sie auf der TYPO3-Webseite.⁵

Voraussetzung ist natürlich, dass Sie der Administrator des Systems sind und grundlegende Eingriffe in die Installation vornehmen dürfen – und, ganz wichtig, sich diese auch zutrauen!

Unter Linux vergeben Sie zunächst die Schreibrechte auf das TYPO3-Verzeichnis bzw. die Dateien, die aus der Patch-Datei hervorgehen.

```
linux: # cd typo3installationsverzeichnis
linux: # patch -p1 < apiPageModule.diff
```

Nehmen Sie den letzten Patch, der dort hinzugefügt wurde. Angaben zum genauen Pfad zur Datei, nach dem Sie unter Umständen gefragt werden, und andere wichtige Informationen entnehmen Sie der Patch-Datei und dem Verzeichnisbaum Ihres Servers.

Im Extension Manager finden Sie unter tt_products diese Option: Use patch 1822.

Nach Installation des Patches und Aktivierung dieser Einstellung steht Ihnen das erweiterte Seiten-Modul unter tt_products zur Verfügung. Die Produkte sind nun übersichtlicher aufgelistet und Sie müssen für die anderen Tabellen nicht mehr extra auf das Listenmodul wechseln (Abbildung B.5).

The screenshot shows the TYPO3 Backend interface. On the left is a sidebar with navigation menus. The main area is divided into two panes. The top pane shows a tree view of the tt_products extension settings, with 'Erweitert' (Advanced) selected. The bottom pane shows a list of products with columns for 'Titel', 'Preis', 'Farbe', and 'Größe'. The products listed are 'Brilliant Blue' in various sizes and 'Sublime Green'.

| Titel | Preis | Farbe | Größe |
|--------------------------|-------|-----------|-------|
| Brilliant Blue - Large | 4,99 | Dark Blue | L |
| Brilliant Blue - Medium | 0,00 | Dark Blue | M |
| Brilliant Blue - XLarge | 0,00 | Dark Blue | XL |
| Brilliant Blue - XXLarge | 0,00 | Dark Blue | XXL |
| Sublime Green | 4,99 | Green | L |

Abbildung B.5:
Seitenbaum mit
Patch 1822

⁵ <http://bugs.typo3.org/view.php?id=1822>

C

phpMyAdmin

In verschiedenen Kapiteln wurde bereits das Programm phpMyAdmin erwähnt,¹ ein mächtiges Tool zur Administration von MySQL-Datenbanken. Diese Applikation können Sie als TYPO3-Extension installieren und über das Webmodul aufrufen. Sie benötigen für phpMyAdmin nur geringe SQL-Kenntnisse, denn die Bedienung ist weitgehend intuitiv. In der übersichtlichen Oberfläche können Sie einzelne Datenbanktabellen auswählen und ansehen. Bei `tt_products` z. B. erhalten Sie so eine Übersicht der Feldnamen, die erfreulicherweise genauso heißen wie die Endungen der Marker im Shop-Template.

Der Marker für das Feld `title` in der Tabelle `tt_products` heißt demnach `PRODUCTS_TITLE`. Nur beim Preis gibt es Abweichungen, weil hier

¹ phpMyAdmin wurde von Tobias Ratschiller initiiert und basiert – ebenso wie phpPgAdmin – auf Peter Kuppelwiesers MySQL-Webadmin. Die Projektbetreuung haben momentan Michal Čihař, Marc Delisle, Garvin Hicking, Michael Keck, Sebastian Mendel, Olivier Müller, Alexander M. Turek und Robin Johnson inne.

die MwSt., Rabatte, Zwischensummen und Gutscheine berücksichtigt werden müssen. Mit phpMyAdmin können Sie natürlich auch Ihre Datenbank bearbeiten, um z. B. zum Löschen markierte Datensätze tatsächlich zu entfernen oder wiederherzustellen. Das geschieht über das Feld `deleted`, das in jeder TYPO3-Tabelle vorkommt. Ist es auf 1 gesetzt, ist der Datensatz als gelöscht markiert, bei 0 ist er aktiv. Ähnlich verfahren Sie mit dem Feld `hidden`, um versteckte Datensätze wieder sichtbar zu machen.

Einige wichtige Operationen seien im Folgenden kurz beschrieben; starten Sie dazu phpMyAdmin nach der Installation der entsprechenden TYPO3-Extension gleichen Namens im Modul Tools des TYPO3-Backends.² Es sei jedoch ausdrücklich darauf hingewiesen, dass Eingriffe in die Tabellen oder gar Datenbankstruktur Ihres Systems weitreichende Folgen haben können. Die unter Umständen vom Shop-Administrator zu erledigenden Aufgaben wie Datensicherung oder `-export` werden ab Seite 276 behandelt.

Abbildung C.1:
phpMyAdmin
Startseite



Auf der phpMyAdmin-Startseite können Sie einige grundlegende Einstellungen vornehmen:

Neue Datenbank anlegen

Voraussetzung für diese Operation sind entsprechende (Admin-)Rechte im System.

Prozesse anzeigen

Hier finden Sie alle aktuellen Zugriffe auf die Datenbank – üblicherweise nur ein einziger Prozess, und zwar jener, der den SQL-Befehl `SHOW PROCESSLIST` zur Anzeige dieser Prozessliste abgesetzt hat. Im Falle einer Überlastung des Servers wäre es möglich, hier Prozesse manuell zu beenden. Allerdings kann dies zu Inkonsistenzen in der Datenbank führen, da Schreiboperationen auf die Datenbank unvollständig bleiben. Sie müssten anschließend dafür sorgen, dass die Datenbank wieder in einen lauffähigen Zustand zurückversetzt wird,

² Über Ihren Webhoster können Sie phpMyAdmin ggf. auch als selbständige PHP-Anwendung ohne TYPO3 nutzen, was sich insbesondere bei komplexen Abfragen und großen Datenmengen empfiehlt.

und beispielsweise halb gefüllte Tabellen löschen und neu aus einer Datensicherung importieren. Kurz: Hier – wie bei den meisten direkten Datenbankmanipulationen – sollten Sie nicht nur über ein Backup, sondern auch Erfahrung im Umgang mit Datenbanken und Tabellen verfügen.

Zeichensätze und Kollationen

Hier erhalten Sie Beschreibungen der installierten Zeichensätze (Abbildung C.2). Sie können Daten in den MySQL-Tabellen in einem dieser Zeichensätze speichern; es ist jedoch empfohlen, nur die Voreinstellung `latin1_swedish_ci` oder den UTF-8 Zeichensatz `utf8_general_ci` zu verwenden. Mit UTF-8 können Sie nicht nur deutsche Umlaute, sondern nahezu alle wichtigen Schriftzeichen abspeichern.

Tabellenformate

Hier finden Sie eine Kurzbeschreibung der MySQL-internen Tabellenformate, die je nach Anforderung (Geschwindigkeit, Speicherverbrauch, Transaktionsmöglichkeit etc.) mehr oder weniger geeignet sind.

Datenbanken

Hier wählen Sie eine Datenbank für die Bearbeitung mit phpMyAdmin aus. Bei mehreren TYPO3-Instanzen können Sie zwischen diesen einen Datenaustausch durchführen.

Exportieren

Siehe Seite 274.

Language

Stellen Sie über **German (de-utf-8)** als Sprache Deutsch im UTF-8 Zeichensatz ein. Dies entspricht der Anzeige von phpMyAdmin im Browser-Fenster.

MySQL-Zeichensatz

Der Zeichensatz **UTF-8 Unicode (utf8)** wird intern für die Metadaten von MySQL verwendet.

Zeichensatz/Kollation der MySQL-Verbindung

Hier legen Sie fest, welcher Zeichensatz für Daten in allen neu angelegten Tabellen verwendet wird. Die Kollation bestimmt die sprachspezifischen Sonderzeichen, die bei Zeichenoperationen zu berücksichtigen sind. Über TYPO3 ist es aber auch möglich, im Frontend UTF-8 zu verwenden, auch wenn Sie im Backend oder in MySQL mit einem anderen Zeichensatz arbeiten.

Oberflächendesign

Verschiedene Ansichten des Backends von phpMyAdmin.

phpMyAdmin-Dokumentation

Hier finden Sie genauere Ausführungen zu den häufigsten Problemen.

Offizielle phpMyAdmin-Homepage

Führt Sie zur Website des Projekts.³

[ChangeLog]

Eine Übersicht über die zuletzt am PHP-Programmcode durchgeführten Verbesserungen und Korrekturen.

[CVS]

Link auf ein Verzeichnis unter `sourceforge.net`, wo Sie den aktuellen Programmcode finden.

[Lists]

Hier können Sie in den Archiven der phpMyAdmin-Mailinglisten suchen und sich dort registrieren, um Fragen zu stellen.

Abbildung C.2:
phpMyAdmin
Zeichensätze und
Kollationen

| Zeichensätze und Kollationen | | | |
|--|---|---|---|
| Kollation | Beschreibung | Kollation | Beschreibung |
| armSCII8 (ARMScii-8 Armenian) | | latin2 (ISO 8859-2 Central European) | |
| armSCII8_bin | Armenisch, Binär | latin2_bin | Mitteleuropäisch (multilingual), Binär |
| armSCII8_general_ci | Armenisch, case-insensitive | latin2_croatian_ci | Kroatisch, case-insensitive |
| ascii (US ASCII) | | latin2_czech_cs | Tschechisch, case-sensitive |
| ascii_bin | Westeuropäisch (multilingual), Binär | latin2_general_ci | Mitteleuropäisch (multilingual), case-insensitive |
| ascii_general_ci | Westeuropäisch (multilingual), case-insensitive | latin2_hungarian_ci | Ungarisch, case-insensitive |
| big5 (Big5 Traditional Chinese) | | latin5 (ISO 8859-9 Turkish) | |
| big5_bin | Traditionelles Chinesisch, Binär | latin5_bin | Türkisch, Binär |
| big5_chinese_ci | Traditionelles Chinesisch, case-insensitive | latin5_turkish_ci | Türkisch, case-insensitive |
| binary (Binary pseudo charset) | | latin7 (ISO 8859-13 Baltic) | |
| binary | Binär | latin7_bin | Baltisch (multilingual), Binär |

C.1 Daten sichern

Wichtig beim Umgang mit ständig sich verändernden Daten – und somit auch bei Ihrem Webshop – sind tägliche Datensicherungen, auch wenn Ihr Web-Hoster diese bereits vornimmt. Jede TYPO3-Extension, die Sie installieren, birgt ein Sicherheitsrisiko, denn über eine Sicherheitslücke könnten Unbefugte Zugang zum TYPO3-Backend und somit Zugriff auf sämtliche Daten im Shop erlangen,⁴ und spätestens dann ist ein Backup Gold wert.

Löschen Sie zunächst die TYPO3-Cache-Tabellen, um Platz von überflüssigen temporären Dateien einzusparen.

³ http://www.phpmyadmin.net/home_page/index.php

⁴ Darum nochmals die dringende Aufforderung an Sie als Administrator, sich über TYPO3-Newslisten über entdeckte Sicherheitslücken zu informieren.

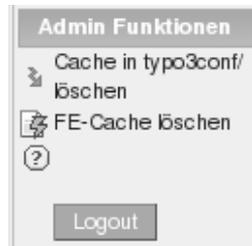


Abbildung C.3:
TYPO3-Cache
löschen

Wählen Sie den Menüpunkt **Exportieren** und darin **Alle auswählen**. Unter **SQL-Optionen** sind Mit **'DROP TABLE'** und Mit **'IF NOT EXISTS'** sinnvoll, wenn Sie später einen Import in eine bereits vorhandene TYPO3-Umgebung vornehmen.

Es muss unbedingt die Option **Senden** (eine merkwürdige Übersetzung des originalen und besser verständlichen **Save as file**) aktiviert sein. Danach geben Sie in **Dateinamenskonvention** eine Bezeichnung für Ihre TYPO3-Umgebung und ein geeignetes Komprimierungsformat an. Starten Sie nun den Export mit Klick auf den **OK-Button**. Die so generierte SQL-Datei sollten Sie zuletzt auf einem von der Webpräsenz unabhängigen Medium sichern. Machen Sie eine Kontrolle, ob Sie diese Datei in einem Editor öffnen können. Überprüfen Sie stichprobenartig das Vorhandensein einiger Daten aus Ihrer TYPO3-Seite. Eine komprimierte Datei müssen Sie mit einer Archivierungssoftware zuvor entpacken.

Wollen Sie ausschließlich die Shop-Tabellen sichern, wählen Sie diese aus; welche das sind, erfahren Sie über die Datei `ext_tables.php`, die zu jeder Extension gehört.

Über den Extension Manager können Sie das noch einfacher erreichen: Wählen Sie darin `tt_products` und in der Select-Box rechts oben **Dump DB** aus. „Dump“ bedeutet hier: Sicherung der Tabellenstruktur, bei der keine Daten gesichert werden. Hier finden Sie sämtliche Tabellennamen von `tt_products`. Es ist aber auch über den Extension Manager möglich, auf einfache Weise alle Daten einer Extension zu sichern. Wählen Sie im Menü rechts oben **Backup/Delete** und klicken Sie danach auf den Link hinter dem Text **Download all data from**. Das startet den Download einer SQL-Datei mit allen Daten aus den Tabellen von `tt_products`. Das Ergebnis ist eine `t3x-`, keine `t3d-` Datei. Das hat den Nachteil, dass eine solche Datei nicht über phpMyAdmin, sondern über die Import/Export-Extension importiert werden müsste. Einen solchen Import können Sie über das Kontext-Menü auf jeder Seite des Seitenbaumes starten (**Weitere Einstellungen | Importieren aus t3d**). Sie können jedoch auf eine einzelne Tabelle, z. B. `tt_products` klicken und somit einen Export dieser Datei durchführen. Das Ergebnis ist dann die Datei `T3X_tt_products-tt_products+adt.sql`.

Damit erhalten Sie den gewünschten Dump der Daten:

```
# TYPO3 Extension Manager dump 1.1
#
# Host: localhost    Database: TYPO3
#-----

#
# Table structure for table "tt_products"
#
DROP TABLE IF EXISTS tt_products;
CREATE TABLE tt_products (
  uid int(11) unsigned auto_increment,
  pid int(11) unsigned default '0',
  ...

```

Eine Sicherung ist aber nur dann von Nutzen, wenn sich daraus die ursprünglichen Daten auch wiederherstellen lassen. Machen Sie die Probe mit einer leeren TYPO3-Instanz und importieren Sie dorthin die gesicherten SQL-Daten, wie ab Seite 277 beschrieben.

C.2 Tabellen leeren und löschen

Wie bereits mehrfach erwähnt, sollten auf Ihrem Server keine Kreditkarten- oder Kontodaten gespeichert sein. Sollte Ihre Shop-Anwendung dies aber verlangen und eine entsprechende Zertifizierung vorliegen, sichern Sie solche Daten regelmäßig (per Cron-Job) zusammen mit den Daten der Bestellungen (`sys_products_orders`) über den zuvor beschriebenen Export und löschen Sie anschließend die Daten aus dem Online-Shop.

Wählen Sie beispielsweise die entsprechende Tabelle aus `tt_products` namens `sys_products_cards` aus und klicken Sie anschließend **Leeren**. Wählen Sie versehentlich **Löschen**, würde der Shop im Frontend von nun an Fehlermeldungen produzieren.

Ein weiteres Beispiel für das Leeren einer Tabelle sei `sys_products_orders`, in der die Bestellvorgänge gespeichert sind. Diese Tabelle sollte gut gefüllt sein, denn sie enthält alle im Verlauf des Buches getätigten Testbestellungen. Bevor der Shop produktiv eingesetzt wird, sollten die Testeinträge natürlich gelöscht werden. Rufen Sie dazu die Tabelle über phpMyAdmin auf und löschen Sie alle Datensätze, die die Testbestellungen enthalten – entweder über die Registerkarte **Leeren** oder Auswahl einzelner Einträge. Markieren Sie die Datensätze mit **Alle auswählen** und klicken Sie anschließend auf das (rote) Kreuz. Sie werden daraufhin aufgefordert, die SQL-Anweisung zu bestätigen.



Abbildung C.4:
Datensätze manuell
löschen

Wenn Ihnen dieser Weg zu aufwendig ist, können Sie in der oberen Leiste die Registerkarte **SQL** auswählen und die Anweisung direkt als SQL-Query eingeben.



Abbildung C.5:
Datensätze löschen
mit Delete

Sie können die Query natürlich über Bedingungen einschränken.

C.3 Tabellen wiederherstellen

Liegt Ihnen ein Backup Ihrer Shop-Datenbank in Form einer `.sql`-Datei vor, können Sie damit eine vollständige oder teilweise Rekonstruktion in Angriff nehmen, denn diese Datei enthält im Klartextformat alle zur Wiederherstellung von Struktur und Inhalten der Datenbank notwendigen SQL-Befehle; jeder dieser – meist in Großbuchstaben hinterlegten – Befehle endet mit einem Semikolon, also z. B.:

```
CREATE TABLE tt_products (
  uid int(11) unsigned NOT NULL auto_increment,
  pid int(11) unsigned DEFAULT '0' NOT NULL,
  ...
);
```

So erzeugt `CREATE TABLE` eine neue Datenbanktabelle, während `INSERT INTO` deren Felder mit Werten befüllt., z. B.:

Diese ist darauf zurückzuführen, dass die per INSERT übergebenen Werte nicht zu den Spalten der in dieser Installation vorhandenen Tabelle `tt_products` passen. Der Grund: Die Export-Datei, aus der der INSERT-Befehl übernommen wurde, enthält keine Informationen über die Tabellenspalten, und bei einer Neuinstallation von `tt_products` in einer leeren TYPO3-Instanz entspricht die Reihenfolge der Datenbankfelder nicht derjenigen der TYPO3-Umgebung Ihres Shops.

Um ein solches Problem zu vermeiden, empfiehlt es sich, beim Export stets **Data: Complete inserts** zu setzen; dadurch werden auch die Tabellenfelder exportiert.

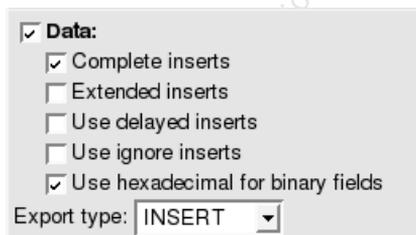


Abbildung C.9:
Data: Complete
inserts

Das Ergebnis in der SQL-Exportdatei hat dann ungefähr die folgende Form:

```
INSERT INTO `tt_products` (
    `uid`,
    `pid`,
    `tstamp`,
    ...
);
```

Aber auch mit diesem vollständigen INSERT könnte es Fehlermeldungen ähnlich der von Abbildung C.10 geben.



Abbildung C.10:
Fehler 2 beim Import

Der Grund könnten Unterschiede in der Tabellenstruktur bei verschiedenen Versionen von `tt_products` sein. Sie müssen dann durch das Anlegen neuer oder das Löschen nicht gemeinsamer Spalten dafür sorgen, dass die Tabellenstruktur von Ausgangs- und Zielsystem identisch ist.

Sämtliche beim Import durchgeführten SQL-Anweisungen werden angezeigt. Klicken Sie nun nochmals auf `tt_products` in der linken Spalte mit den Tabellennamen und anschließend oben auf **Anzeigen**.

Zeige Datensätze 0 - 17 (18 insgesamt, die Abfrage dauerte 0.0008 sek)

SQL-Befehl:

```
SELECT *
FROM tt_products
LIMIT 0, 30
```

[Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

Zeige : 30 Datensätze, beginnend ab 0
 untereinander angeordnet und wiederhole die Kopfzellen nach 100 Datensätzen.
 Nach Schlüssel sortieren: keine OK

| uid | pid | tstamp | hidden | starttime | endtime | title | note | price | price2 | image |
|-----|-----|------------|--------|-----------|---------|---------------------------|---|--------|--------|------------------------|
| 1 | 236 | 1192881922 | 0 | 0 | 0 | Ein Testprodukt | Das erste Produkt im Shop ist das Testprodukt. Das... | 39.90 | 34.90 | HC_Titel_T3webshop.png |
| 2 | 236 | 1189511221 | 0 | 0 | 0 | Geschenkgutschein 20 Euro | Diesen Gutschein können Sie an Freunde verschenken... | 20.00 | 0.00 | gutschein.gif |
| 134 | 236 | 1187880480 | 1 | 0 | 0 | T-Shirt | Ein wunderschönes T-Shirt | 18.50 | 0.00 | produkt.gif |
| 135 | 254 | 1191694055 | 0 | 0 | 0 | Spezielles Produkt | Dieses Spezialprodukt für registrierte Benutzer, d... | 159.00 | 0.00 | dummy_02.gif |
| 136 | 236 | 1192878439 | 0 | 0 | 0 | Spezialanfertigung | ein ganz großes und sperriges Produkt | 19.00 | 0.00 | dummy.gif |

Abbildung C.14:
`tt_products` Tabelle
 importieren

Damit ist der Import abgeschlossen. Auf diese Weise können Sie jede SQL-Datei, die über einen Datenbank-Export erzeugt wurde, auch wieder importieren. Ggf. müssen Sie dabei auftretende Datenbankfehlermeldungen beheben. Um diese zu vermeiden, empfiehlt es sich, mit einer völlig leeren Datenbank beim Import zu arbeiten und damit automatisch Tabellen neu anzulegen.

In diesem Beispiel des Imports der Tabelle `tt_products` kann diese nicht im TYPO3-Seitenbaum angezeigt werden, weil der Seitenbaum nicht importiert wurde und die TYPO3-Umgebung noch leer ist. Sie müssten also auch `pages` mit den Seiten und `tt_content` mit den Seiteninhalten und ggf. weitere Tabellen aus Ihrer produktiven Umgebung exportieren und hier wieder importieren.

Sie müssten also auch die Tabelle `pages` aus Ihrer produktiven Umgebung exportieren und hier wieder importieren.

Wenn Sie nicht Daten bzw. Werte, sondern ausschließlich leere Tabellen wiederherstellen wollen, ist es deutlich einfacher, die fehlenden über den Extension Manager anzulegen. Diesen Anwendungsfall gibt es bei jeder Neuinstallation einer Extension: Sie haben das Verzeichnis `tt_products` in das Hauptverzeichnis `typo3conf/ext` über einen Dateitransfer kopiert. Damit sind zwar die PHP-Dateien von `tt_products` vorhanden, aber noch keine Eintäge in der Datenbank. Klicken Sie auf die Extension `tt_products` und installieren Sie diese neu, indem Sie auf **UPDATE** der Datenbanktabellen klicken.

Alternativ können Sie auch im Install Tool unter **2: Database Analyser** den Punkt **Update required tables COMPARE** aufrufen. Damit werden ebenso alle Korrekturen an der Datenbank durchgeführt, wie sie die aktuell installierten TYPO3-Extensions benötigen. Diese Vorgehensweise hat den Vorteil, dass alles in einem Durchgang erledigt wird. Sie müssen nicht mehr jede Extension einzeln im Extension Manager installieren, sondern können eine fertige Datei `typo3conf/localconf.php` mit den bereits als installiert eingetragenen Extensions kopieren. Sie können diese Datei auch editieren und unterhalb von `$TYPO3_CONF_VARS['EXT']['extList'] =` durch Komma getrennt anhängen. Damit installieren Sie mehrere Extensions in einem Ruck ohne Umweg über den Extension Manager. Hier entfernen Sie auch Extensions, wenn zum Beispiel das TYPO3-Backend durch einen PHP-Fehler nicht mehr erreichbar ist. Es müssen anschließend aber noch die `tempCACHED...`-Dateien im Verzeichnis `typo3conf` gelöscht werden.

C.4 Weitere Export-Formate

Der Datenexport gibt Ihnen darüber hinaus die Möglichkeit, anstelle von SQL Formate wie Word, Excel oder LaTeX zu verwenden, um die Daten weiterzuverarbeiten, Berechnungen durchzuführen oder sich Statistiken zu generieren.

Abbildung C.15:
Export nach Excel

The screenshot shows the 'Exportieren' (Export) dialog in phpMyAdmin. The 'Exportieren' section is expanded, showing options for SQL, LaTeX, Microsoft Excel 2000, Microsoft Word 2000, CSV-Daten für MS Excel, CSV-Daten, and XML. The 'Microsoft Excel 2000' option is selected. The 'CSV-Optionen' (CSV Options) section is also expanded, showing settings for field delimiters, quoting, escaping, line delimiters, and NULL replacement. The 'Senden' (Send) section is expanded, showing the file name convention and compression options. The 'Exportiere' field shows '202' and 'Datensätze ab Zeile' (Records from line) shows '0'. An 'OK' button is located at the bottom right of the dialog.

Beispielhaft wollen wir `sys_products_orders` für statistische Zwecke exportieren und die gesamte Tabelle in ein Excel-Format exportieren: Wählen

Sie dazu die Registerkarte **Export** und definieren Sie die Exportdatei. Klicken Sie den Radiobutton **Microsoft Excel 2000**, wählen Sie mit der Checkbox **Feldnamen** in die erste Zeile setzen und vergeben Sie einen Dateinamen.

Wenn Sie das Tracking nutzen und dort die Bestellungen als bezahlt definiert haben, können Sie auch nur die als bezahlt markierten Datensätze exportieren:

```
SELECT *
FROM sys_products_orders
WHERE STATUS IN ( 60, 13 )
```

Der Status-Wert kommt aus dem Tracking. Die 13 steht für „Order has been paid“, die 60 für den Versand eines Geschenkgutscheins. Sofern Sie keine Gutscheine nutzen, können Sie diese Zahl weglassen. Wenn Sie allerdings das Gutscheinsystem nutzen, muss dieser Wert zugewiesen werden. In der Tabelle `sys_products_orders` wird nämlich der jeweils zuletzt vergebene Status verwendet. Wenn also andere Statusmeldungen nach „Order has been paid“ folgen, wie z. B. der Gutscheinversand durch den Gutschein-Käufer, gilt dieser Wert für den Zustand der Bestellung, und nicht mehr „Order has been paid“.

Die Liste der selektierten Datensätze können Sie nun exportieren, indem Sie alle Datensätze auswählen und auf das Icon **Exportieren** klicken. Sofern Sie mehr als 30 Datensätze selektiert haben, müssen Sie die Anzeige aller Datensätze veranlassen (**Anzeigen**). Andernfalls würden Sie nur die aktuell angezeigten Datensätze exportieren. Wählen Sie, wie oben beschrieben, Ihr Export-Format aus, vergeben Sie einen Dateinamen und klicken Sie auf **OK**.

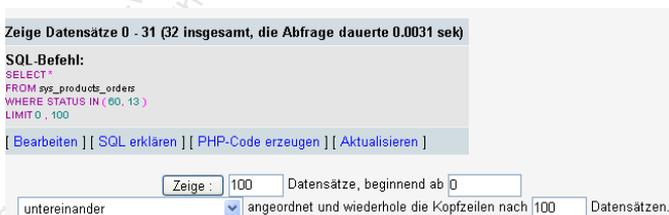


Abbildung C.16:
Export im
Excel-Format

C.5 Daten suchen und ändern

Klicken Sie auf die Tabelle, welche die gesuchten Informationen enthält. Sie können die Daten über **Anzeigen** überprüfen und danach über **Suchen** exakte Suchbegriffe (=) oder ähnliche (LIKE) übergeben. Beispiel:

```
name LIKE '%gesuchtername%'
```

Die Prozentzeichen (%) hier sind wichtig: Es sind Platzhalter für beliebige andere Zeichen. In der Ergebnisliste können Sie über das Stift-Symbol einen Datensatz editieren und die Daten ändern.

Herzog-Kienast/Holzinger: "Der TYPO3-Webshop" (<http://creativecommons.org/licenses/by-sa/3.0/de/>)
Original Edition (C) Open Source Press (<http://www.opensourcepress.de>)

D

Anhang

tt_products im Constant Editor

Beachten Sie, dass einige der aktuell im Constant Editor möglichen Einstellungen bald überholt sein werden (im Manual der Extension als „deprecated“ markiert). Durch die fortschreitende Entwicklung von tt_products werden komplexere Konfigurationen notwendig,¹ die sich nur noch über das Setup oder die Constants vornehmen lassen, wo z. B. auch der Code/Flexform und die verwendete Tabelle mit angegeben werden muss, denn nur so erhält man genügend Flexibilität. Die zu verwendenden Datentypen bei der Eingabe werden im Constant Editor angezeigt. Fast alle Konstanten können 1:1 auch im Setup verwendet werden. Nur bei der file-Einstellung müssen Sie die Syntax beachten. Einige Konstanten können im Setup eine komplexere Form mit zusätzlichen Einstellungen haben. Es sei auch nochmals darauf hingewiesen, dass ein Setup- immer Vorrang vor einem Constants-Eintrag hat.

¹ Die Extension mbf_products_pricecalc von Esben Fest bietet beispielsweise ein Backend-Interface zum Setup von Preisberechnungen.

D.1 Referenz

`file.templateFile`

HTML-Template-Datei; eine einfache Textdatei, standardmäßig im ISO-8859-1-Zeichensatz² mit mehreren Subparts für die verschiedenen Shop-Plugins. In jedem Haupt-Subpart steht HTML-Code gemischt mit weiteren Subparts und Markern. Eine Beispieldatei ist die voreingestellte `template/example_template_bill_de.tpl`

`file.noImageAvailable`

Ein alternatives Bild, das verwendet wird, wenn es für ein Produkt kein Bild gibt.

`file.datasheetIcon`

Bild für das Datenblatt-Symbol.

`wrap1`

Wrap Nr. 1, der über das Marker-Paar GW1B/GW1E eingefügt wird. Standardmäßig ist hier ` | ` voreingestellt. Es wird zum einheitlichen Design der Texte eingesetzt. Verwenden Sie für das Design besser CSS-Templates, wo Sie solche Wraps nicht mehr benötigen.

`wrap2`

Wrap Nr. 2 für GW2B/GW2E

`wrap3`

Wrap Nr. 3 für GW3B/GW3E

`n12brNote`

Umbruch für das Bemerkungsfeld in der Bestellung: Zeilenumbrüche im Feld Bemerkung werden in `
`-HTML-Tags umgewandelt.

`useArticles`

Artikel-Tabelle verwenden: Hier muss eine 1 stehen, wenn die Artikel-Tabelle verwendet werden soll. Sie benötigen diese, wenn Produktvarianten einen eigenen Preis oder andere Textbeschreibungen haben sollen. Dann übernimmt die Produktvariante den Preis oder den Text aus dem Artikel, der dem Produkt zugeordnet ist und mit ihm in den Attributen der Variante (Farben, Größen etc.) übereinstimmen muss. Setzen Sie 2, wenn die Artikel keinerlei Übereinstimmung mit den Varianten eines Produktes haben sollen, z. B. in einem Shop, wo grundsätzlich Artikel statt Produkten gepflegt werden.

² Über verschiedene Einstellungen kann man auch einen anderen Zeichensatz wie UTF-8 verwenden.

`color1`

Farbe 1; Marker GC1 für die Farbe. Verwenden Sie besser eine CSS-Datei und CSS-Klassen. Standardmäßig auf #cccccc gesetzt.

`color2`

Farbe 2; Marker GC2 für die Farbe. Standard: #999999.

`color3`

Farbe 3; Marker GC3 für die Farbe. Standard: #333333.

`clickEnlarge`

Klickvergrößerung; beim Klick auf ein Bild in der Einzelansicht öffnet sich ein neues Fenster mit vergrößertem Bild.

`clickEnlarge_list`

Wie `clickEnlarge`, aber für die Listenansicht.

`clickEnlarge_listHasChilds`

Wie `clickEnlarge`, aber in der Listenansicht, wenn die Produkte nach Kategorien gefiltert sind und es eine Unterkategorie zur aktuellen Kategorie gibt.

`clickEnlarge_listRoot`

Klickvergrößerung bei Root-Kategorie; wie `clickEnlarge`, aber in der DAM-Listenansicht, wenn die aktuelle Kategorie eine Root-Kategorie ist.

`clickEnlarge_basket`

Klickvergrößerung der Bilder im Warenkorb.

`clickEnlarge_listcat`

Klickvergrößerung der Kategorie Bilder.

`selectColor`

Farbe ist über Select-Boxen auswählbar.

`selectColor2`

Farbe 2 ist über Select-Boxen auswählbar.

`selectColor3`

Farbe 3 ist über Select-Boxen auswählbar.

`selectSize`

Größe ist über Select-Boxen auswählbar.

`selectSize2`

Größe 2 ist über Select-Boxen auswählbar.

`selectSize3`

Größe 3 ist über Select-Boxen auswählbar.

- `selectDescription`
Beschreibung ist über Select-Boxen auswählbar.
- `selectGradings`
Staffelung ist über Select-Boxen auswählbar.
- `selectMaterial`
Material ist über Select-Boxen auswählbar.
- `selectQuality`
Qualität ist über Select-Boxen auswählbar.
- `displayListCatHeader`
Anzeige des Kategorie-Kopfteils in der Listenansicht; die Kategorie-Überschriften (Seitentitel) werden automatisch in der Produktliste angezeigt. Das ist nicht immer ein Vorteil, denn vielleicht möchten Sie lieber ein Inhaltselement als Überschrift einfügen.
- `displayBasketCatHeader`
Anzeige des Kategorie-Kopfteils im Warenkorb; (siehe `displayListCatHeader`).
- `displayCatListType`
Typ der Anzeige in der Kategorieliste; HTML-Tag, das für die Kategorieliste verwendet wird. Standard: ``.
- `displayCurrentRecord`
`cObj`-Einstellungen; Darstellung eines Einzelprodukts über den Datensatz aus `$cObj->data` der aktuellen Seite.
- `basketPic`
URL des Bildsymbols für den Warenkorb.
- `clickIntoBasket`
Beim Klick in den Warenkorb; wenn gesetzt, wird nach Klick auf den Button **in den Warenkorb** das Produkt in den Warenkorb gelegt, und dann die Seite sofort auf den Warenkorb gewechselt. `PIDbasket` muss gesetzt sein.
- `clickIntoList`
Beim Klick in der Einzelansicht; wenn gesetzt, wird nach Klick auf den Button **in den Warenkorb** das Produkt in den Warenkorb gelegt und dann die Seite sofort auf die Listenansicht gewechselt. `PIDlistDisplay` muss gesetzt sein, damit der Shop zur Listenansicht findet.
- `clickIntoSubmenu`
Beim Klick ins Untermenü; die Untermenü-Einträge einer Kategorieliste werden nur nach einem Klick auf deren Vaterkategorie angezeigt.

`clickItemsIntoSubmenu`

Beim Klick auf ein Bild in der Listenansicht wird auf die Unterkategorie gewechselt, wenn es sie gibt.

`basketMaxQuantity`

Maximale Anzahl an Artikeln, die in den Warenkorb gelegt werden dürfen.

`quantityIsFloat`

Anzahl an Produkten ist eine Fließkommazahl; es können also auch Nachkommastellen in der Anzahl verwendet werden. Dies ist beim Verkauf von Obst und Gemüse denkbar, wenn es Mengenangaben in kg sind.

`CSSRowEven`

Gerade Zeilen mit CSS; CSS für gerade Zeilen in der Listenansicht.

`CSSRowUneven`

Ungerade Zeilen mit CSS; CSS für ungerade Zeilen in der Listenansicht.

`CSSListDefault`

Standard-CSS; CSS für Standardzeilen in der Listenansicht.

`CSSListCurrent`

Aktuelle CSS; CSS für aktuelle Zeilen in der Listenansicht.

`NoSingleViewOnList`

Normalerweise wird auf der Listenansicht auch die Einzelansicht angezeigt, wenn Sie auf den Link zur Einzelansicht geklickt haben. Es wird also entweder die Listenansicht oder die Einzelansicht dargestellt, aber man bleibt immer auf derselben Seite. Wenn Sie dafür aber getrennte Seiten haben wollen, müssen Sie hier 1 setzen.

`limit`

Maximale Anzahl an Produkten, die in der Listenansicht gezeigt werden.

`limitImage`

Maximale Anzahl an Bildern eines Produkts, die in der Listenansicht angezeigt werden.

`limitImageSingle`

Maximale Anzahl an Bildern eines Produkts, die in der Einzelansicht angezeigt werden.

`usePageContentImage`

Bilder von den Inhaltselementen auf der Seite mit dem Produkt werden als Bilder des Produkts verwendet.

`separateImage`

Standardmäßig werden alle Bilder eines Produkts aneinandergereiht und als 1 Bild dargestellt. Wenn 1 gesetzt ist, werden die Bilder getrennt. Nun können Sie die einzelnen Bildermarker mit Nummern verwenden: `PRODUCT_IMAGE3`.

`maxW_single`

Maximale Bildbreite in der Einzelansicht; maximale Pixelanzahl der Breite eines Bildes eines Produkts in der Einzelansicht.

`maxH_single`

Maximale Bildhöhe in der Einzelansicht; maximale Pixelanzahl der Höhe eines Bildes eines Produkts in der Einzelansicht.

`maxW_list`

Maximale Bildbreite in der Listenansicht; maximale Pixelanzahl der Breite eines Bildes eines Produkts in der Listenansicht und bei Suchergebnissen.

`maxH_list`

Maximale Bildhöhe in der Listenansicht; maximale Pixelanzahl der Höhe eines Bildes eines Produkts in der Listenansicht und bei Suchergebnissen.

`maxW_listHasChilds`

Maximale Bildbreite in der DAM-Listenansicht; maximale Pixelanzahl der Breite eines Bildes eines Produkts in der DAM-Listenansicht, wenn es noch eine Unterkategorie zur ausgewählten Kategorie gibt.

`maxH_listHasChilds`

Maximale Bildhöhe in der DAM-Listenansicht; maximale Pixelanzahl der Höhe eines Bildes eines Produkts in der DAM-Listenansicht, wenn es noch eine Unterkategorie zur ausgewählten Kategorie gibt.

`maxW_listRoot`

Maximale Bildbreite in der DAM-Listenansicht; die Kategorie ist eine Root-Kategorie.

`maxH_listRoot`

Maximale Bildhöhe in der DAM-Listenansicht; die Kategorie ist eine Root-Kategorie.

`maxW_basket`

Maximale Bildbreite in der Warenkorbansicht; maximale Pixelanzahl der Breite eines Bildes eines Produkts im Warenkorb.

`maxH_basket`

Maximale Bildhöhe in der Warenkorbansicht; maximale Pixelanzahl der Höhe eines Bildes eines Produkts im Warenkorb.

maxW_listcat

Maximale Bildbreite der Kategorie in der Listenansicht; maximale Pixelanzahl der Breite eines Bildes einer Kategorie in der Listenansicht und bei Suchergebnissen.

maxH_listcat

Maximale Bildhöhe der Kategorie in der Listenansicht; maximale Pixelanzahl der Höhe eines Bildes einer Kategorie in der Listenansicht und bei Suchergebnissen.

maxW_popup

Maximale Bildbreite des Bildes im Popup-Fenster; maximale Pixelanzahl der Breite eines Bildes im Fenster, das sich beim Klick auf ein Bild geöffnet hat.

pid_list

Liste der Seite mit erlaubten Produkten; Setzen Sie das auf die Seite oberhalb der SysOrdner/Seiten mit den Produkten und verwenden Sie außerdem `recursive`. Oder geben Sie hier durch Komma getrennt die Seitennummern ein, aus denen Produkte, Artikel, Kategorien usw. für diesen Shop ausgelesen werden sollen. Wenn Sie aber einen Ausgangspunkt im Plugin setzen, so wird dieser stattdessen verwendet, aber nur für die Seite, auf der sich das Shop-Plugin befindet.

PIDinfo

Seiten-ID für die Seite mit dem Plugin Warenkorb: Eingabe Kundendaten.

PIDfinalize

Seiten-ID für die Seite mit dem Plugin Warenkorb: Bestellung abschließen.

PIDthanks

Seiten-ID für eine Dankesseite, auf die nach Abschluss der Bestellung statt auf `PIDfinalize` gewechselt werden soll.

PIDtracking

Seiten-ID für die Seite mit dem Plugin Bestellung: Verwaltung.

PIDbilling

Seiten-ID für die Seite mit dem Plugin Bestellung: Rechnung.

PIDdelivery

Seiten-ID für die Seite mit dem Plugin Bestellung: Lieferschein.

PIDpayment

Seiten-ID für die Seite mit dem Plugin Warenkorb: Kontrolle und Bezahlung.

`PIDitemDisplay`

Seiten-ID für die Seite mit dem Plugin Produkte: Einzelansicht. Bei verschiedenen Einzelansichten je nach Produkt müssen Sie auf ein komplexeres Setup ausweichen.

`PIDlistDisplay`

Seiten-ID für die Seite mit dem Plugin Produkte: Liste. Bei verschiedenen Listenansichten je nach Produkt müssen Sie auf ein komplexeres Setup ausweichen.

`PIDsearch`

Seiten-ID für die Seite mit dem Plugin Produkte: Suche. Wenn alle Suchergebnisse auf einer speziellen Seite ausgegeben werden sollen, geben Sie hier eine Seiten-ID ein.

`PIDbasket`

Seiten-ID für die Seite mit dem Plugin Warenkorb: Inhalt. Wenn Sie nach dem Einfügen von Produkten in den Warenkorb auf die Warenkorb-Seite wechseln wollen, geben Sie hier die PID ein.

`PIDstoreRoot`

Oberste Seite des Shops; lassen Sie dies besser leer und verwenden Sie `pid_list`. Damit läuft es schneller, weil weniger Seiten nach Produkten durchsucht werden müssen.

`PIDmemo`

Seiten-ID für die Seite mit dem Plugin Allgemein: Memo.

`PID_sys_products_orders`

Seiten-ID für die Seite, in der die eingegangenen Bestellungen abgespeichert werden. Wenn das nicht gesetzt ist, wird alles auf der Seite mit dem Plugin Warenkorb: Inhalt abgespeichert.

`PIDgiftsTable`

Seiten-ID für die Datensätze der Geschenkgutscheine. Standardmäßig werden Sie die Seiten-ID von `PIDfinalize` erhalten.

`PIDuserFolder`

Seiten-ID des SysOrdners für neu angelegte Benutzer.

`pidsRelatedProducts`

Seiten-IDs, in denen verwandte Produkte gefunden werden.

`pidsAddresses`

Seiten-IDs, in denen Adressen gefunden werden.

`paymentActivity`

Definiert den Zeitpunkt im Bestellablauf, zu dem die Bezahlung durchgeführt wird. Es gibt 2 mögliche Werte:

`payment` Für Bezahlung nach der Kontrolle der Daten und vor dem Abschluss der Bestellung. Das hat den Vorteil, dass eine Bestellung nicht durchgeführt wird, wenn der Käufer noch nicht bezahlt hat.

`finalize` Wenn die Bezahlung erst nach dem Abschluss der Bestellung erfolgt. Dies hat den Vorteil, dass der Käufer auch später noch bezahlen kann.

`advanceOrderNumberWithInteger`

Wenn gesetzt, wird die Bestellnummer immer um eine Zufallszahl zwischen 1 und dieser Zahl erhöht. Damit können Sie bei den Bestellnummern etwas mogeln und mehr eingegangene Bestellungen vortauschen, als die angezeigte Bestellnummer vorgibt. Wenn eine Person 2 Bestellungen innerhalb einer gewissen Zeitspanne abgibt, kann sie die Bestellnummern überprüfen und Rückschlüsse auf die Kaufaktivitäten im Shop ziehen. Um das zu verhindern, muss hier eine Erhöhung der Bestellnummer per Zufall aktiviert sein. So hat nur noch der Shop-Administrator Zugriff auf die tatsächlichen Zahlen.

`alwaysAdvanceOrderNumber`

Eine Bestellnummer wird normalerweise für den nächsten Kunden wiederverwendet, wenn der erste eine Bestellung nicht zu Ende geführt, sondern vorher abgebrochen hat. Wenn gesetzt, wird die Bestellnummer immer um wenigstens 1 erhöht. Das wird für manche Bezahl-Gateways benötigt. Wenn dort eine Bestellung mit derselben Bestellnummer zweimal bezahlt wird, wird das nicht akzeptiert. Manche Gateways verlangen, dass die Bestellnummer bei jedem Aufruf eine andere sein muss, selbst wenn zuvor abgebrochen wurde; sie wollen immer eine eindeutige Nummer haben. Das wird bei `paymentActivity= payment` für manche Gateways benötigt.

`alwaysUpdateOrderAmount`

Wenn das gesetzt ist, wird die Anzahl der Produkte im Warenkorb immer auf die Zahl gesetzt, die Sie im Eingabefeld eingegeben haben. Wenn Sie also zunächst 2 Produkte A eingegeben haben und dann noch 3 Produkte A dazugeben, werden Sie insgesamt nur 3 Produkte im Warenkorb haben. Die letzte eingegebene Zahl wird immer für das Produkt A genommen. Das ist auch die Standard Einstellung. Das hat den Vorteil, dass jemand, der 5 Produkte A in den Warenkorb gelegt hat, aber eigentlich nur 3 haben wollte, nun in der Listenansicht einfach 3 schreiben kann. Er muss nicht extra in den Warenkorb wechseln, um die Anzahl auch reduzieren zu können. Alternativ müssten Sie Löschen oder + / - Buttons in der Listenansicht einfügen.

`defaultArticleID`

Artikel-ID (uid aus der Tabelle `tt_products_articles`), die in der Einzelansicht eines Artikels gezeigt wird, wenn kein Parameter `tx_ttproducts_pi1[article]` vorhanden ist.

`defaultProductID`

Produkt-ID (uid aus der Tabelle `tt_products`), die in der Einzelansicht gezeigt wird, wenn kein Parameter `tx_ttproducts_pi1[product]` vorhanden ist.

`defaultCategoryID`

Kategorie-ID (uid aus der Tabelle `tt_products_cat`), deren Produkte in der Listenansicht gezeigt werden, wenn kein Parameter `tx_ttproducts_pi1[product]` vorhanden ist.

`defaultDAMCategoryID`

DAM-Kategorie-ID (uid aus der Tabelle `tt_dam_cat`), von der Produkte in der Listenansicht gezeigt werden, wenn kein Parameter `tx_ttproducts_pi1[dam_cat]` vorhanden ist.

`rootCategoryID`

Root-Kategorie; die oberste Kategorie, die für Kategorielisten verwendet wird.

`rootDAMCategoryID`

Root-DAM-Kategorie; die oberste DAM-Kategorie, die für DAM-Kategorielisten verwendet wird.

`rootPageID`

Root-Seiten-ID; die oberste Seiten-ID bzw. Kategorie, die für Kategorielisten bei Seiten als Kategorie verwendet wird.

`recursive`

Anzahl an Unterebenen im TYPO3-Seitenbaum, nach denen rekursiv Produkte etc. gesucht werden sollen; Standard 99. Sie müssen das nur dann ändern, wenn die Produkte auf den Unterseiten nicht im Shop verwendet werden sollen. Wenn Sie im Plugin `rekursiv` ausgewählt haben, hat das für die Seite des Plugins Vorrang vor dieser Constants/Setup-Einstellung.

`domain`

Domäne Ihres Shops; der globale Marker `DOMAIN` wird damit ersetzt, z. B. `www.mytypo3domain.de`. Das macht es einfacher, eine Shop-HTML-Template-Datei auch auf Ihrem Testserver unter einer anderen Domäne laufen zu lassen.

`TAXpercentage`

Geben Sie hier den ersten Steuersatz ein. Er wird für alle Produkte verwendet, in denen Sie nichts im Steuerfeld eingetragen haben.

`TAXincluded`

Wenn das gesetzt ist, ist die Steuer in den Preisen der Produkte enthalten. Sie müssen entscheiden, welche Preise Sie im Backend eintra-

gen wollen. Bedenken Sie, dass der Endkunde „runde“ Preise bevorzugt und keine Zahlen wie 23,61 als Preis. Daher ist es auch standardmäßig auf 1 gesetzt und aktiv. So geben Sie im Backend beim Produkt die runden Endkundenpreise an.

TAXmode

Dieser Steuermodus soll immer auf der Voreinstellung 1 bleiben. Damit ist die in Deutschland übliche Steuerberechnung aktiv: Zuerst werden die Nettosummen (Preise ohne Steuer) aller Produkte und Gebühren für Versand und Bezahlung addiert. Es wird in den internen Berechnungen nirgends gerundet. Erst auf die Endsumme und ebenso die angezeigten Teilsummen wird die Steuer aufgeschlagen. Bei 1 wird sofort die Steuer auf das Einzelprodukt aufgeschlagen und auch auf 2 Nachkommastellen gerundet. Am Ende werden die Summen der Einzelprodukte addiert.

TAXrates

Steuersätze; durch Komma getrennte Liste von Steuersätzen in Prozent. Der erste wird als Steuersatz A, der zweite als B genommen usw. Diese werden bei der Marke PRICE_TAXRATE_TAX1 usw. angezeigt.

priceDec

Anzahl an Nachkommastellen im Preis.

priceDecPoint

Dezimaltrennzeichen bei der Ausgabe des Preises. Standard ist der Punkt; in Deutschland können Sie das Komma setzen.

priceThousandPoint

Trennpunkt bei der Ausgabe der Tausender-Stelle im Preis.

currencySymbol

Währung im Shop; Standard EUR.

lockLoginUserInfo

Aktivieren eines eingeloggten Frontendbenutzers für den Shop; wenn gesetzt, werden die Adressdaten des eingeloggten Benutzers automatisch für den Shop verwendet und es können keine Rechnungsdaten mehr eingegeben werden, sie sind also gesperrt.

editLockedLoginInfo

Dasselbe wie lockLoginUserInfo, aber der eingeloggte Benutzer darf einige Felder seiner Daten ändern. Dazu muss aber auch das HTML-Template angepasst werden, damit die statischen Ausgabertexte in EDIT-Felder umgewandelt werden. Der Subpart BILLING_ADDRESS_LOGIN, der innerhalb eines Warenkorb-Subparts liegt, muss dafür überarbeitet werden.

`loginUserInfoAddress`

Hier werden die Datenbank-Tabellenfelder Adresse, Land, Postleitzahl und Stadt des eingeloggten Frontendbenutzers in einem einzigen Adressfeld zusammengefasst. `lockLoginUserInfo` muss aktiviert sein. Sie benötigen das nur, wenn Sie nur ein einziges Adressenfeld im Shop haben wollen.

`orderByItemNumberSg`

Sortierung der Produkte in der Einzelansicht. In der Einzelansicht erhalten Sie über die Marker `LINK_NEXT_SINGLE` und `LINK_PREV_SINGLE` Buttons **vor** und **zurück** angezeigt, um ohne Rückkehr in die Listenansicht zu den nächsten Produkten zu gelangen. Mit dieser Einstellung können Sie die Sortierung angeben, in der die Produkte gereiht werden.

`orderNumberPrefix`

Präfix für die Bestellnummer in der Ausgabe. In der Ausgabe können Sie die Bestellnummer kennzeichnen: `MEIN_SHOP_343222` statt bisher `343222`. Geben Sie Ihr Wunschpräfix ein, also z. B. `MEIN_SHOP_`

`orderEmail_from`

E-Mail des Absenders, also des Shopbetreibers, in der E-Mail an den Kunden, z. B. `shop@mytypo3domain.de`.

`orderEmail_fromName`

Name des Absenders, also des Shopbetreibers, z. B. `MY TYPO3 DOMAIN GERMANY`.

`orderEmail_to`

E-Mail des Empfängers, also des Shopbetreibers für die Bestellbestätigung, z. B. `shop@mytypo3domain.de`. Sie können hier eine durch Komma getrennte Liste mehrerer E-Mail-Adressen eingeben.

`orderEmail_toDelivery`

Wenn 1 gesetzt, werden die E-Mail-Benachrichtigungen an die Lieferadressen, nicht an die Rechnungsadresse geschickt.

`orderEmail_subject`

Betreff-Zeile der E-Mail in der Bestellbestätigung; wenn leer, wird die erste Zeile aus dem Subpart `EMAIL_PLAINTEXT_TEMPLATE` verwendet.

`orderEmail_htmlmail`

Wenn 1 gesetzt, werden die E-Mails im HTML-Format versendet.

`orderEmail_apostrophe`

E-Mail-Header-Zeilen in Apostrophen einbinden. Wenn auf Anführungszeichen " gesetzt, wird dies in der Betreffzeile vorne und hinten angehängt: "Ihre Bestellung bei `mytypo3domain.de`"

email_notify_default

Wenn gesetzt, sind die E-Mail-Benachrichtigungen an die Kunden aktiviert. Das wird für das Tracking verwendet. Der Kunde kann das anschließend aber selbst auf der Seite mit dem Plugin **Bestellung: Verwaltung** für jede Bestellung einstellen.

update_code

Passwort des Shop-Administrators; wenn dieser im Backend eingeloggt ist und auf die Seite mit dem Plugin **Bestellung: Verwaltung** wechselt, muss er im Frontend dieses Passwort eingeben und erhält dann die Liste aller Bestellungen.

usePriceTag

Wenn 1 gesetzt, können Sie den Preis als Grafik statt als Text anzeigen lassen. Verwenden Sie dafür das Setup `priceTagObj cObject`.

useStaticInfoCountry

Wenn gesetzt, wird die Länder-Select-Box über das Static Info Country Feld `static_info_country` des Frontendbenutzers `fe_users` angezeigt. Das geschieht über den Marker `PERSON_COUNTRY_CODE`. Das ersetzt das `country`-Feld.

bulkilyWarning

Warnmitteilung bei Produkten, die als sperrig gekennzeichnet sind.

bulkilyAddition

Zusatzkosten für ein Produkt, das als sperrig gekennzeichnet ist.

bulkilyFeeTax

Steuer in Prozent der Zusatzkosten für ein Produkt, das als sperrig gekennzeichnet ist.

specialPreparation

Hinweistext für Produkte, die als Spezialanfertigung gelten. Sie können nicht direkt im Shop bestellt werden. Es muss eine Anfrage an den Shop gestellt werden. Der Preis ist hier unverbindlich und hängt von den tatsächlichen Anforderungen ab. Beispiel `Special preparation possible! Anfrage wegen Spezialanfertigung`.

max_note_length

Maximale Anzahl an Zeichen, die im Bemerkungsfeld einer Bestellung eingegeben werden können.

createUsers

Wenn gesetzt, werden die Daten eines Erstbestellers nach dem Abschluss der Bestellung in die `fe_users`-Tabelle eingetragen. Der Besteller erhält Benutzername und Passwort per E-Mail zugesendet. Als Benutzername wird immer die E-Mail Adresse verwendet.

outputFolder

Name inkl. relativem Pfad zum Verzeichnis, in dem die im Shop erzeugten Dateien für die Rechnung und den Lieferschein abgespeichert werden. Das ist standardmäßig `fileadmin/data`. Setzen Sie noch die Schreibrechte für den Apache auf dieses Verzeichnis.

memberOfGroup

Frontendbenutzer-Gruppe, mit der ein Kunde neu in der Tabelle `fe_users` angelegt wird (siehe `createUsers`).

PIDagb

ID der Seite mit den AGB des Shops.

AGBtarget

HTML-Target der AGB-Seite `PIDagb` ist auf `_blank` voreingestellt.

AGBattachment

AGB-Datei als Anhang. Das wird benötigt, damit der Kunde in der Bestellbestätigung diese Datei als Anhang erhält.

generateCSV

Erzeugen einer CSV-Datei mit den Bestelldaten. Diese wird auch an den Shop-Administrator in der Bestellbestätigung als Anhang mitgesendet.

CSVdestination

Verzeichnis für CSV-Dateien; relatives Verzeichnis, in dem die CSV-Dateien der Bestellungen gespeichert werden. Achtung: Diese Dateien dürfen nicht über einen Webbrowser anzeigbar sein! Wählen Sie ein Verzeichnis außerhalb des Apache-Root-Verzeichnisses oder setzen Sie eine `.htaccess` Datei. `../csvorders/` ist voreingestellt.

CSVfields

Felder, die in der CSV-Datei verwendet werden.

CSVinOneLine

Wählen Sie das Format der CSV-Datei; wenn auf 1 gesetzt, werden alle zusätzlichen Informationen in einer Zeile hinter dem ersten bestellten Produkt geschrieben. Das hat Vorteile beim Dateimport. Lassen Sie das leer, ist es optimal auf MS Excel abgestimmt.

stdSearchFieldExt

Suchfelder; in diesen Datenbank-Tabellen Feldern wird nach den eingegebenen Begriffen gesucht.

alwaysInStock

Wenn 1 gesetzt, wird die Zahl bei auf Lager eines Produkts bei einem Kauf nicht reduziert. Das gilt global für alle Produkte. Die Produkte müssen also immer lieferbar sein.

showNotInStock

Anzeige der Produkte, die nicht auf Lager sind; wenn 1 gesetzt, werden auch jene Produkte in den Listenansichten angezeigt, bei denen 0 auf Lager eingetragen sind.

notInStockMessage

Meldung, die bei Produkten angezeigt wird, die nicht auf Lager sind.

inStockPieces

Stückerheit (pieces)

warningInStockLimit

Limit für eine Warnmeldung wegen zu geringe Zahl auf Lager bei einem Produkt. Wenn die Anzahl eines Produkts im Shop nach einem Verkauf diese Schwelle unterschritten hat, wird eine E-Mail mit einem Warnhinweis und dem Text aus `instock_warning` an den Shop-Administrator gesendet.

newItemDays

Anzahl an Tagen, die ein neu eingestelltes Produkt als neu gilt. Das beeinflusst die Anzeige des Plugins **Produkte: Liste Neue Artikel**.

substitutePageTitle

Den Seitentitel ersetzen; wenn 1 gesetzt, wird der Titel eines Produkts in der Einzelansicht **Produkte: Einzelansicht** als Seitentitel angezeigt. Wenn auf 2 gesetzt, wird analog der Untertitel des Produkts als Seitentitel angezeigt.

whereGift

SQL-Where-Bedingung, über die alle Produkte gefunden werden, die als Geschenkgutscheine verwendet werden können. Diese Produkte erhalten die Funktionalität eines Geschenkgutscheins.

debug

Debug-Modus ist aktiviert, wenn 1 gesetzt. Sie sollten das für Testzwecke aktivieren. Damit wird der Warenkorb nach dem Abschluss einer Bestellung nicht gelöscht. Das soll in einem Online-Shop nicht gesetzt sein, denn sonst müsste der Kunde seine Produkte manuell aus dem Warenkorb löschen. Außerdem bleiben auch die eingegebenen Adressdaten erhalten, die sonst nach dem Abschluss einer Bestellung immer gelöscht werden.

D.2 clickIntoBasket

Wenn Sie möchten, dass durch Anklicken des Warenkorb-Icons ein direkter Aufruf des Warenkorbs erfolgt, müssen Sie in den Constants oder über den Constant-Editor Folgendes definieren:

```
plugin.tt_products.clickIntoBasket = 1
```

Falls Sie `clickIntoBasket` verwenden, müssen Sie den Zurück-Link von der Einzel- zur Listenansicht über `history.back()` definieren. Der Zurück-Link `LINK_ITEM` ist in diesem Kontext nicht möglich, weil laut Definition ja sofort auf die Seite des Warenkorbs gewechselt werden muss.

D.3 showNotInStock

In diesem Beispielshop sollen auch jene Produkte angezeigt werden, die aus verschiedenen Gründen nicht lieferbar sind. Ein Produkt, also ein Buch, kann vergriffen und bis zur nächsten Auflage nicht lieferbar sein; oder es ist vergriffen und wird nicht wieder aufgelegt. Die Anzeige der Produkte kann aber dennoch sinnvoll sein: Es könnte ja später eine Neuauflage geben, oder der Shopbesucher soll informiert werden, dass es dieses Buch einmal gegeben hat. Er kann es sich evtl. über ein Antiquariat oder eine Bibliothek besorgen. Für den Verlag ist das Anklicken des Produkts zudem ein statistischer Wert, der evtl. für oder gegen eine Neuauflage des Produkts entscheiden kann. Der Shop selbst verfügt nicht über diese Möglichkeit, allerdings können Sie ein Statistikmodul wie die Extension `awstats` einsetzen und die Informationen aus den Statistiken beziehen.

```
showNotInStock = 0
```

Wird in **Am Lager** der Wert 0 eingetragen, wird das Produkt normalerweise im Frontend nicht angezeigt. Im Plugin **Produkte: Einzelansicht** wird der Subpart `ITEM_SINGLE_DISPLAY` für die Anzeige verwendet.

Wenn Sie jedoch in den Konstanten

```
showNotInStock = 1
```

setzen, wird das Produkt angezeigt, aber als nicht lieferbarer Artikel deklariert. Der Subpart `ITEM_SINGLE_NOT_IN_STOCK` wird damit im Plugin **Produkte: Einzelansicht** statt des normalen Subpart `ITEM_SINGLE_DISPLAY` für alle jene Produkte verwendet, die nicht auf Lager sind. Wenn aber **Am Lager** einen Wert größer Null 0, also beispielsweise 17 hat, dann wird für diese Produkte auch weiterhin wie oben der Subpart `ITEM_SINGLE_DISPLAY` verwendet.

Die Nachricht `notInStockMessage` wird im Marker `PRODUCT_INSTOCK` ausgegeben. Dieser kann im HTML-Template des Shops beim Produkt eingefügt werden.

Weitere detaillierte Informationen über die Lieferung können in der Registerkarte **Versand** gesetzt werden. Hier stehen Ihnen noch **auf Kundenwunsch, In Kürze lieferbar und sofort lieferbar** zur Verfügung. Sofern Sie diese Werte zuweisen, erscheint im Frontend das entsprechende Ampellicht.

Der Marker `PRODUCT_DELIVERY` wird durch ein Symbol (`availableShort.gif`, `availableDemand.gif` oder `availableImmediate.gif`) dargestellt. Sie können das über das Setup einstellen:

```

delivery      {
    0.image < .image
    0.image {
        file =
EXT:tt_products/res/icons/fe/availableDemand.gif
        file.maxW = $plugin.tt_products.maxW_single
    }
    1.image < plugin.tt_products.delivery.0.image
    1.image {
        file =
EXT:tt_products/res/icons/fe/availableImmediate.gif
    }
    2.image < plugin.tt_products.delivery.0.image
    2.image {
        file =
EXT:tt_products/res/icons/fe/availableShort.gif
    }
}

```

Ähnlich gibt es eine `instock_warning` als Sprachtext in der Datei `pi1/locallang.xml`. Dies wird verwendet, um einen Shopbetreiber per E-Mail zu benachrichtigen, dass die Anzahl auf Lager eines Produkts eine gewisse Schwelle unterschritten hat.

