

# **Extension Netcos Renting**

*Von Bernd Warken bei Fa. Netcos AG*

<b>0. Einleitung</b>	<b>3</b>
0.1 Vorwort	3
0.2 Lizenz	3
<b>1. Kickstarter</b>	
1.1 Erstellung der Extension	4
1.2 Neue Datenbanktabelle	4
1.3 Frontend Plugins	4
1.4 Abspeichern und beenden	5
<b>2. Objektmethoden für Termin</b>	<b>6</b>
<b>3. Zugaben</b>	<b>15</b>
3.1 smarty	15
3.2 jscalender	15
3.3 Mehrsprachigkeit	16
<b>4. PHP-Quellcode anpassen</b>	<b>22</b>
4.1 Datenbankzugriff	22
4.2 Erstes Plugin (Vermietungsverwaltung)	22
4.3 Zweites Plugin (vermietet)	31
<b>5. HTML-Templates</b>	<b>38</b>
5.1 Erstes Plugin (Vermietungsverwaltung)	38
5.2 Zweites Plugin (vermietet)	42

# 0. Einleitung

## 0.1 Vorwort

Dieses Dokument beschreibt die Hintergründe für die Erstellung einer TYPO3-Extension, die Zugriff auf die Datenbank gestattet. Dies ist eine offizielle Extension, sie ist auf [typo3.org](http://typo3.org) erhältlich.

Die Extension wird vorab mit dem *Kickstarter* erzeugt und dann manuell geändert. Als Beispiel wird die Vermietung eines Apartments behandelt. Dabei werden zwei Frontend-Plugins angelegt, eine zur Vermietungsverwaltung und eine zum Anschauen der Buchungen für alle Benutzer.

Die Termine werden im Format *yyyymmdd* gespeichert, womit sich das Datum besser vergleichen läßt.

Zur besseren Darstellung des Kalenderformats bei der Eingabe wird das Programm *jscalender* verwendet. Um die Quellcode-Dateien etwas kleiner zu halten, wird zur Auslagerung des HTML-Codes in HTML-Dateien die Extension *smarty* angewendet.

Die Extension besitzt Mehrsprachigkeit, bisher wurden die 2 Sprachen Englisch und Deutsch verwendet. Im Quellcode der Extension werden die mehrsprachigen Begriffe in den Dateien *locallang\*.xml* gespeichert.

Dieses Dokument wurde auf Linux entwickelt und beschreibt auch alle Pfade in Linux-typischer Weise. Dies passt jedoch auch für Mac. Wenn Sie jedoch mit Microsoft Windows arbeiten, müssen Sie sämtliche Pfadangaben entsprechend anpassen. Sonst sind keine Änderungen nötig.

## 0.2 Lizenz

Dieses Dokument gehört zum TYPO3-Projekt über die **Extension netcos\_renting**. Die aktuelle Version dieses Dokuments ist **1.0** vom 3. April 2008. Der Autor ist **Bernd Warken** bei Firma [netcos AG](http://netcos.de).

Alle Elemente des Projekts (dieses Dokument und der Quellcode) stehen unter der Lizenz **GPL (GNU General Public License) Version 3**. Der Originaltext dieser Lizenz ist erhältlich unter <http://www.gnu.org/licenses/gpl.html>, eine deutsche Übersetzung befindet sich unter <http://www.gnu.de/documents/gpl.de.html>.

Copyright 2008 Bernd Warken

# 1. Kickstarter

*Kickstarter* ist eine Typo3-Extension zur Erstellung eigener Extensions. Zunächst muss *kickstarter* über den *Extension Manager* installiert werden.

## 1.1 Erstellung der Extension

Durch *kickstarter* entsteht im Menü des *Extension Managers* ein neues Feld *Neue Extension anlegen*. Wir wählen dieses Feld.

Im entstehenden Formular gibt man zuerst den Namen im Feld *Enter extension key*: an, in diesem Fall **netcos\_renting**. Bestätigung mit dem Button *Update...*

Im Folgenden wird die Optionsliste *KICKSTARTER WIZARD* im Formular bearbeitet.

Man drückt zuerst das Pluszeichen von *General info*. Darin erstellt man den *Title* **Netcos Renting**. Als *Category* wählt man *Frontend Plugins*. Man lässt *State* auf *Alpha*. Für *Dependencies* gibt man die Extensions an, die zusätzlich gebraucht werden, also **cms** und **smarty** an. *Author* hinzufügen. Den Button *Update...* betätigen

Als nächstes das Pluszeichen von *Setup languages* und darin die Sprache *German* wählen. Bestätigung durch *Update...* Die Standardsprache der Extension ist immer Englisch, die gewählte deutsche Sprache ist eine Zusatzsprache.

## 1.2 Neue Datenbanktabelle

Wir werden die Buchungen in einer neuen Datenbanktabelle speichern. Dazu wird das Pluszeichen von *New Database Tables* betätigt. Als *Tablename* geben wir an: **fromto**. Mit Hilfe dieses Namens wird eine neue Tabelle innerhalb der Datenbank für die TYPO3-Installation angelegt. Der richtige Tabellennamen lautet: **tx\_netcosrenting\_fromto**. Als *Title of the table* geben wir an **rented** und auf Deutsch **vermietet**.

Nun werden noch die Felder angegeben, die in der Tabelle verwendet werden. Ein Buchungstermin hat zwei Felder zur Speicherung der Zeiträume: **from\_date** und **to\_date**.

In *NEW FIELD* tragen wir daher erst **from\_date** als *Field name* ein, mit *Field title* **From** in English und **Von** in German. Der *Field type* ist **String input**. *Field width* und *Max. characters* sind jeweils 8. *Update...*

In dem neu entstehenden *NEW FIELD* tragen wir dann **to\_date** als *Field name* ein, mit *Field title* **To** in English und **Bis** in German. Der *Field type* ist wieder **String input**. *Field width* und *Max. characters* sind wieder jeweils 8. *Update...*

## 1.3 Frontend Plugins

Nun das Pluszeichen von *Frontend Plugins* zweimal betätigen. Dadurch entstehen zwei neue Felder zur Erstellung von Plugins, das eine für die Vermietungsverwaltung und das andere zum Anschauen der Buchungen.

Im ersten dieser Pluginformulare wird auf Englisch der *title*: **Renting Administration** angegeben und in Deutsch **Vermietungsverwaltung**. Auswählen des Buttons *Add to 'Insert Plugin' list in Content Elements* und darin *Add icon to 'New Content Element' wizard*. *Update*.

Im anderen Pluginformular wird auf Englisch der *title*: **rented** angegeben und in Deutsch **vermietet**. Anklicken *Add to 'Insert Plugin' list in Content Elements* und darin *Add icon to 'New*

*Content Element' wizard.* Update.

## **1.4 Abspeichern und beenden**

Die Eingabe ist jetzt abgeschlossen. Daher den Feldbutton *View result* wählen. Dort auf *WRITE* klicken. Die Extension wird dann auf *typo3conf/ext/netcos\_renting/* abgespeichert. Diese Extension wird nun im Menü-Feld *Install extensions* des *Extension Managers* angezeigt und kann durch Klicken des Pluszeichens installiert werden.

Die Quellcode-Dateien der Extension können angesehen werden durch Klicken auf den Extension-Namen im *Extension Manager*. Im entstehenden Menü muss *Edit files* ausgesucht werden. Dies stellt die Verzeichnisstruktur des Quellcodes der Extension dar. Da zwei Plugins eingerichtet wurden, enthält der Extension-Quellcode auch zwei Unterverzeichnisse *pi1* und *pi2* mit dem Quellcode für die Plugins.

## 2. Objektmethoden für Termine

Die Termine für die Buchungen werden in der Datenbank im Format *yyyymmdd* abgespeichert. Der 23. Januar 2008 würde dann also als Wert *20080123* gespeichert. Bei der Erstellung der Extension werden wir mit diesem Format in PHP rechnen.

Zum Umwandeln und Handhaben der verschiedenen Datumsformate ist es sinnvoll, eine PHP-Klasse zu erstellen. Wir nennen diese Klasse **Termin**. Den Code für diese Klasse speichern wir in der Datei *class\_termin.php* im Quellcode-Verzeichnis der Extension.

Da sich die Sprachmeldungen in dieser Datei nicht so ohne weiteres erstellen lassen, werden alle Fehlermeldungen in Englisch ausgegeben. Für den normalen Betrieb reicht das. Die noch vorhandenen deutschen Strings sind nicht mehr nötig, z.B. **\$Monatsnamen**, bleiben aber vorläufig noch in der Datei.

Da diese Datei sehr lang ist, haben wir sie in viele Teile zerlegt. Wenn Sie die gesamte Datei haben wollen, müssen Sie alle diese Abschnitte in dieser Sektion zusammensetzen oder schauen Sie im Quellcode der Extension nach.

Start und Lizenz:

```
<?php
/*****
* Copyright notice
*
* (c) 2007,2008 Bernd Warken <be.warken@netcos.de>
* All rights reserved
*
* This script is part of the TYPO3 project. The TYPO3 project is
* free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* The GNU General Public License can be found at
* http://www.gnu.org/copyleft
*
* This script is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* This copyright notice MUST APPEAR in all copies of the script!
*****/

class Termin {
```

Private Variablen:

```

private $value;
private $max;
private $error = "";
private $timestamp = 0;
static private $max_Tage = array(0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
static private $Monatsnamen = array(0, 'Januar', 'Februar', 'März', 'April', 'Mai', 'Juni',
    'Juli', 'August', 'September', 'Oktober', 'November', 'Dezember');
static private $Monatsnummern = array(
    'Januar' => 1, 'januar' => 1, 'Jan' => 1, 'Jan' => 1, 'JAN' => 1,
    'Februar' => 2, 'februar' => 2, 'Feb' => 2, 'feb' => 2, 'FEB' => 2,
    'März' => 3, 'märz' => 3, 'Maerz' => 3, 'maerz' => 3,
    'Mär' => 3, 'mär' => 3, 'MÄR' => 3, 'Mae' => 3, 'mae' => 3, 'MAE' => 3,
    'April' => 4, 'april' => 4, 'Apr' => 4, 'APR' => 4, 'apr' => 4,
    'Mai' => 5, 'mai' => 5,
    'Juni' => 6, 'juni' => 6, 'Jun' => 6, 'jun' => 6, 'JUN' => 6,
    'Juli' => 7, 'juli' => 7, 'jul' => 7, 'Jul' => 7, 'JUL' => 7,
    'August' => 8, 'august' => 8, 'Aug' => 8, 'aug' => 8, 'AUG' => 8,
    'September' => 9, 'september' => 9, 'Sep' => 9, 'sep' => 9, 'SEP' => 9,
    'Oktober' => 10, 'oktober' => 10, 'Okt' => 10, 'okt' => 10, 'OKT' => 10,
    'November' => 11, 'november' => 11, 'Nov' => 11, 'nov' => 11, 'NOV' => 11,
    'Dezember' => 12, 'dezember' => 12, 'Dez' => 12, 'dez' => 12, 'DEZ' => 12
);
static private $char2int = array('0' => 0, '1' => 1, '2' => 2, '3' => 3, '4' => 4, '5' => 5,
    '6' => 6, '7' => 7, '8' => 8, '9' => 9);

```

Konstruktion eines Termin-Objekts aus einem Datumstring:

```

public function __construct($value) {
    $str = trim((string)$value);
    $this->value = "";
    if ($str == "") {
        $this->append_error('empty Termin data. ');
        return(FALSE);
    }
    if ($this->_is_Termin($str)) {
        $this->value = $str;
        $m = $this->Monat();
        if ($m <= 0 || $m > 12) {
            $this->append_error('wrong month number: '.$m.'. ');
            $this->value = "";
            return(FALSE);
        }
        $t = $this->Tag();
        $max = $this->max_Tage_des_Monats();
        if ($t <= 0 || $t > $max) {
            $this->append_error('wrong number of days: '.$t.', maximal number of days of the month is '.
$max.'. ');
            $this->value = "";
            return(FALSE);
        }
        $this->max = $max;
        return(TRUE);
    }
    $len = strlen($str);
    $a = explode("/", $str);
    $c = count($a);
    $english = 'yes';
    switch($c) {
    case 3:
        $j = $a[0];
        $m = $a[1];
        $t = $a[2];
        break;
    case 2:
        $j = date('Y');
        $m = $a[0];
        $t = $a[1];
        break;
    default:
        $english = 'no';
        break;
    }
    if ($english == 'yes') {
        $s = "";
        if ($this->_is_int($j)) {
            $j = (int)$j;
            if ($j < 100) $j += 2000;

```

```

if ($j < 2000 || $j >= 3000) {
    $this->append_error('Wrong year number: '.$j.'.');
    return(FALSE);
}
$s = (string)$j;
}
if (strlen($s) != 4) {
    $this->append_error('wrong year description: '.$j.'.');
    return(FALSE);
}
if ($this->_is_int($m)) {
    $m = (int)$m;
    if ($m > 0 && $m < 10) {
        $s .= '0'.(string)$m;
    } elseif ($m <= 12) {
        $s .= (string)$m;
    } else {
        $this->append_error('wrong month number: '.$m.'.');
        return(FALSE);
    }
} else {
    $this->append_error('month must be a number: '.$m.'.');
    return(FALSE);
}
if ($this->_is_int($t)) {
    $t = (int)$t;
    $max = $this->max_Tage_des_Monats($m, $j);
    if ($t <= 0 || $t > $max) {
        $this->append_error('wrong number of days: '.$t.', maximal number of days of the month is
'.$max.'.');
        return(FALSE);
    }
    if ($t > 0 && $t < 10) {
        $s .= '0'.(string)$t;
    } else {
        $s .= (string)$t;
    }
} else {
    $this->append_error('day must be a number: '.$t.'.');
    return(FALSE);
}
$this->value = $s;
return(TRUE);
}

$i = 0;
$s = "";
while ( $this->_is_int( $str{$i} ) && $i < $len ) {
    $s .= $str{$i};
    $i++;
}

```

```

if ($s) {
    $t = (int)$s;
} else {
    $this->append_error('no day number. ');
    return(FALSE);
}
while ($str{$i} == '' && $i < $len) $i++;
while ($str{$i} == '.' && $i < $len) $i++;
while ($str{$i} == '-' && $i < $len) $i++;
if ($i == $len) {
    $m = date('n');
    $j = date('Y');
} else {
    $s = "";
    while ($this->_is_int($str{$i}) && $i < $len) {
        $s .= $str{$i};
        $i++;
    }
    if ($s) {
        $m = (int)$s;
        if ($m <= 0 || $m > 12) {
            $this->append_error('wrong month number: '.$m.' ');
            return(FALSE);
        }
    } else {
        while ($str{$i} != '' && $str{$i} != '.' && $i < $len) {
            $s .= $str{$i};
            $i++;
        }
    }
    if (!$s) {
        $this->append_error('no information for month. ');
        return(FALSE);
    }
    if (self::$Monatsnummern[$s]) {
        $m = self::$Monatsnummern[$s];
    } else {
        $this->append_error('wrong month name: '.$s.' ');
        return(FALSE);
    }
}
while ($str{$i} == '' && $i < $len) $i++;
while ($str{$i} == '.' && $i < $len) $i++;
while ($str{$i} == '-' && $i < $len) $i++;
if ($i == $len) {
    $j = date('Y');
} else {
    $s = "";
    while ( $this->_is_int( $str{$i} ) && $i < $len ) {
        $s .= $str{$i};
        $i++;
    }
}

```



```

private function _is_Termin($str) {
    $s = (string)(int)$str;
    $len = strlen($s);
    if ($len != 8) return(FALSE);
    if (!$this->_is_int($str)) return(FALSE);
    if ($str{0} != '2') {
        $this->append_error('wrong year. ');
        return(FALSE);
    }
    return(TRUE);
}

```

Weitere Methoden:

```

private function _char2int($c) {
    $n = self::$char2int[$c];
    if (isset($n)) return($n); else return(-1);
}

private function _is_int($str) {
    $len = strlen($str);
    for ($i = 0; $i < $len; $i++) {
        if ($this->_char2int($str{$i}) < 0) return(FALSE);
    }
    return(TRUE);
}

public function is_false() {
    return($this->string() == "");
}

public function is_true() {
    return(!$this->is_false());
}

public function int() {
    return (int)$this->value;
}

public function string() {
    return (string)$this->value;
}

private function append_error($str) {
    if ($this->error) $this->error .= '$str; else $this->error = $str;
    return($this->error);
}

public function get_error() {
    return($this->error);
}

```

Ausgabe des aktuellen Termins als Datumstring:

```
public function Datum() {
    if ($this->is_false()) return("");
    $str = $this->string();
    $s = "";
    for ($i = 6; $i <= 7 ; $i++) $s .= $str{$i};
    $s .= '!';
    for ($i = 4; $i <= 5 ; $i++) $s .= $str{$i};
    $s .= '!';
    for ($i = 0; $i <= 3 ; $i++) $s .= $str{$i};
    return($s);
}

public function date_en() {
    if ($this->is_false()) return("");
    $str = $this->string();
    $s = "";
    for ($i = 0; $i <= 3 ; $i++) $s .= $str{$i};
    $s .= '/';
    for ($i = 4; $i <= 5 ; $i++) $s .= $str{$i};
    $s .= '/';
    for ($i = 6; $i <= 7 ; $i++) $s .= $str{$i};
    return($s);
}
```

Ersetze Tag des aktuellen Termins durch 01:

```
public function erster_des_Monats() {
    if ($this->is_false()) return("");
    $str = $this->string();
    $str[6] = 0;
    $str[7] = 1;
    return(new Termin($str));
}
```

Anzahl der Tage innerhalb des aktuellen Monats:

```
public function max_Tage_des_Monats() {
    if (func_num_args() == 2) {
        $m = func_get_arg(0);
        $j = func_get_arg(1);
    } else {
        $m = $this->Monat();
        $j = $this->Jahr();
    }
    if ($m == 2 && $this->_is_Schaltjahr($j)) return(29);
    return(self::$max_Tage{$m});
}
```

Aktueller Tag als Zahl:

```
public function Tag() {
    if ($this->is_false()) return(0);
    $str = $this->string();
    $s = $str{6}.$str{7};
    return((int)$s);
}
```

Aktueller Monat als Zahl:

```
public function Monat() {
    if ($this->is_false()) return(0);
    $str = $this->string();
    $s = $str{4}.$str{5};
    return((int)$s);
}
```

Aktuelles Jahr als Zahl:

```
public function Jahr() {
    if ($this->is_false()) return(0);
    $str = $this->string();
    $s = $str{0}.$str{1}.$str{2}.$str{3};
    return((int)$s);
}
```

Name des aktuellen Monats:

```
public function Monatsname() {
    if ($this->is_false()) return(0);
    $n = $this->Monat();
    if ($n <= 0 || $n > 12) return("");
    return(self::$Monatsnamen[$n]);
}
```

Setze Variable *timestamp* aus dem aktuellen Datum und gib den Variablenwert zurück:

```
public function timestamp() {
    if ($this->timestamp == 0) $this->timestamp = strtotime($this->Datum());
    return($this->timestamp);
}
```

Ausgabe des aktuellen Wochentags als ganzes Wort:

```
public function get_Wochentag() {
    return(strftime('%A', $this->timestamp()));
}
```

Ausgabe der Abkürzung des aktuellen Wochentags mit 2 Buchstaben:

```
public function get_Wochentag_Abk() {  
    return(strftime('%a', $this->timestamp()));  
}
```

Aktueller Wochentag als Zahl, 1=Montag, 7=Sonntag:

```
public function get_Wochentag_Zahl() {  
    return(strftime('%u', $this->timestamp()));  
}
```

Aktuelle Woche des Jahres als Zahlenwert (z.B. 26), mögliche Werte sind 00 bis 53, beginnend mit dem ersten Montag der ersten Woche (also 1 als letzte Woche des Vorjahres):

```
public function get_Kalenderwoche() {  
    return(strftime('%W', $this->timestamp()));  
}
```

Strftime angewandt auf aktuellen Termin:

```
public function strftime($str) {  
    return(strftime($str, $this->timestamp()));  
}
```

Ende der Datei:

```
}  
?>
```

## 3. Zugaben

### 3.1 smarty

*Smarty* ist eine TYPO3-Extension. Sie muss also zuerst installiert werden.

Sie wird in beiden Plugins der erstellten Extension ausgiebig verwendet. Es gibt einige wenige Regeln, die bei der Verwendung beachtet werden müssen. Wir benutzen *smarty*, um aus PHP-Code Teile in Templates auszulagern. Diese Templates sind HTML-Codes, die zusätzlich spezielle Variablen verwenden; diese funktionieren einfach als Abkürzungen, die vom PHP-Code gesetzt werden können.

Als erstes muss die Dateien der Extension *smarty* eingebettet werden. Dies geschieht in der Datei **pi1/class.tx\_netcosrenting\_pi1.php**:

```
require_once(t3lib_extMgm::extPath('smarty').'class.tx_smarty.php');
require_once(t3lib_extMgm::extPath('smarty').'Smarty/libs/Smarty.class.php');
```

Im PHP-Code wird erst einmal eine lokale Instanz der Klasse **Smarty** erstellt. Wir nennen diese Instanz auch **smarty**.

```
$this->smarty = new Smarty;
```

Dann müssen wir festlegen, wo die *smarty*-Templates zu finden sind. Dieses Verzeichnis ist das Unterverzeichnis **tmpl** im Extension-Verzeichnis, das wir mit dem Argument **template\_dir** festlegen. Außerdem wird **tmpl\_c** intern genutzt; wir legen es fest mit dem Argument **compile\_dir**. In beiden Konfigurationen wird das Extension-Verzeichnis durch die PHP-Konstruktion **t3lib\_extMgm::extPath(\$this->extKey)** ermittelt.

```
$this->smarty->compile_dir = t3lib_extMgm::extPath($this->extKey).'tmpl_c';
$this->smarty->template_dir = t3lib_extMgm::extPath($this->extKey).'tmpl';
```

Als nächstes wird festgelegt, wie die Abkürzungen in den HTML-Templates erkannt werden können.

```
$this->smarty->left_delimiter = '{..}';
$this->smarty->right_delimiter = '!.}';
```

Hier ist das Format der Abkürzungen also **{..\$WERT..}**, wobei in diesem Beispiel **WERT** in PHP gesetzt werden kann. Das Setzen dieser Abkürzungsvariablen in PHP geschieht durch den Befehl **assign**.

```
$this->smarty->assign('WERT', 1234);
```

Alle Abkürzungen werden innerhalb des PHP-Skripts gesetzt, meist durch Ergebnisse aus einem *form*-Tag (mit POST oder GET).

Das *smarty*-Template wird aufgerufen durch

```
$content = $this->smarty->fetch($this->smarty->template_dir.'/Datei.tpl');
return $this->pi_wrapInBaseClass($content);
```

Dies ist schon alles.

### 3.2 jscalendar

*Jscalendar* ermöglicht die Verwendung von komfortablen Kalendern in eigenen Extensions. Es ist als Extension *netcos\_jscalendar* erhältlich. Nach der Installation ist der Quellcode positioniert in

`/typo3conf/ext/netcos_jscalendar/jscalendar/`. Dieses Verzeichnis ist relativ zur Domain-URL, so dass das Root-Verzeichnis / das Hauptverzeichnis der TYPO3-Installation ist. Man erhält dieses Root-Verzeichnis, wenn man im TYPO3-Code eine Domain anlegt.

Das Programm *jscalendar* ist in JavaScript geschrieben und erlaubt die Darstellung komfortabler Kalender bei Datumseingaben. Um es zu benutzen, muss es durch *script*-Tags im HTML-Code eingelesen werden.

Im Header einer HTML-*smarty*-Datei wird folgender Code angelegt:

```
<style type="text/css">@import
  url(/typo3conf/ext/netcos_jscalendar/jscalendar/calendar-green.css);</style>
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/calendar.js"></script>
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/calendar-setup.js"></script>
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/lang/calendar-en.js"></script>
```

Darin wird als erstes eine CSS-Datei eingebettet und dann werden 3 Source-Dateien des Kalendersystems eingelesen. Dabei ist die Konfigurationsdatei für die englische Sprache *lang/calendar-en.js* unbedingt nötig, auch wenn mit anderen Sprachen gearbeitet wird.

Wenn die deutsche Sprache aktiv ist, muss auch das Skript *lang/calendar-de.js* aufgerufen werden:

```
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/lang/calendar-de.js"></script>
```

Da wir diesen Einbettungscode in jede Template-Datei aufnehmen müssten und der Sprachtest innerhalb von HTML nicht möglich ist, haben wir dies durch eine *smarty*-Variable übernommen, die in der Haupt-Klassen-Datei jedes Plugins definiert wird. Kapitel 4 gibt die Details hierzu.

Im *body*-Bereich der Template-Datei wird der Kalender innerhalb einer Input-Form aufgerufen, z.B. durch folgenden Code:

```
<form method="post">
  <input type="text" id="data_datum" name="data_datum" />
  <button id="trigger_datum">Datum</button>
  <script type="text/javascript">
    Calendar.setup(
      {
        inputField : "data_Datum", // ID of the input field
        ifFormat   : "%d.%m.%Y", // the date format
        button     : "trigger_datum" // ID of the button
      }
    );
  </script>
  <input type="submit" name="anlegen" value="anlegen" />
</form>
```

Dies sind alle Strukturen, die bei *jscalendar* zu beachten sind.

### 3.3 Mehrsprachigkeit

TYPO3 kann mehrere Sprachen verwenden. Die Begriffe für die unterschiedlichen Sprachen

werden in den *locallang\*.xml*-Dateien definiert. Im Hauptverzeichnis der Extension gibt es die Dateien *locallang.xml* und *locallang-db.xml* für die allgemeinen Definitionen der Extension, sodann gibt es in jedem Plugin-Verzeichnis eine Datei *locallang.xml*, die für das jeweilige Plugin verwendet wird.

Innerhalb der Plugins muss die Mehrsprachigkeit durch den Befehl *pi\_loadLL* aktiviert werden, die Begriffe werden durch den Befehl *pi\_getLL* abgefragt und durch den Wert für die jeweilige Sprache dargestellt.

*locallang.xml*:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
  <meta type="array">
    <type>module</type>
    <description>Language labels for extension 'netcos_renting'</description>
  </meta>
  <data type="array">
    <languageKey index="default" type="array">
      <label index="pi1_title">Renting Administration</label>
      <label index="pi2_title">Rented</label>
    </languageKey>
    <languageKey index="de" type="array">
      <label index="pi1_title">Vermietungsverwaltung</label>
      <label index="pi2_title">vermietet</label>
    </languageKey>
  </data>
</T3locallang>
```

*Locallang-db.xml*

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
  <meta type="array">
    <type>database</type>
    <description>Language labels for database tables/fields belonging to extension
'netcos_renting'</description>
  </meta>
  <data type="array">
    <languageKey index="default" type="array">
      <label index="tx_netcosrenting_fromto">Rented</label>
      <label index="tx_netcosrenting_fromto.from">From</label>
      <label index="tx_netcosrenting_fromto.to">To</label>
      <label index="tt_content.list_type_pi1">Renting Administration</label>
      <label index="tt_content.list_type_pi2">Rented</label>
    </languageKey>
    <languageKey index="de" type="array">
      <label index="tx_netcosrenting_fromto">vermietet</label>
      <label index="tx_netcosrenting_fromto.from">Von</label>
      <label index="tx_netcosrenting_fromto.to">Bis</label>
      <label index="tt_content.list_type_pi1">Vermietungsverwaltung</label>
      <label index="tt_content.list_type_pi2">vermietet</label>
    </languageKey>
  </data>
</T3locallang>
```

*pi1/locallang.xml:*

```

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
  <meta type="array">
    <type>module</type>
    <description>Language labels for plugin &quot;tx_netcosrenting_pi1&quot;</description>
  </meta>
  <data type="array">
    <languageKey index="default" type="array">
      <label index="actual">actual</label>
      <label index="apply">apply</label>
      <label index="back">Back</label>
      <label index="cancel">cancel</label>
      <label index="change1">change</label>
      <label index="change2">change</label>
      <label index="delete">delete</label>
      <label index="existing">existing</label>
      <label index="from">from</label>
      <label index="language">en</label>
      <label index="more">more</label>
      <label index="new">new</label>
      <label index="to">to</label>
      <label index="year">year</label>
      <label index="administration_bookings_actual">Bookings since today</label>
      <label index="administration_bookings_year">Bookings of Year</label>
      <label index="administration_date_format">Date format: yyyy/mm/dd</label>
      <label index="administration_header">Renting</label>
      <label index="administration_new">New Entry</label>
      <label index="administration_none">No entries available.</label>
      <label index="administration_years">Bookings of whole Years</label>
      <label index="calendar_ifformat">%Y/%m/%d</label>
      <label index="exists_header">The entry was not generated because it exists already.</label>
      <label index="overlap_header">Overlap of Entries</label>
      <label index="pi1_comes_after">comes after</label>
      <label index="pi1_erroneous">is erroneous</label>
      <label index="pi1_has_an_error">has an error</label>
      <label index="pi1_has_error">has error</label>
      <label index="to_be_changed_header">Change Entry</label>
      <label index="to_be_deleted_header">The following entry should be deleted</label>
      <label index="was_created_header">The entry was created.</label>
      <label index="was_deleted_header">The entry was deleted.</label>
      <label index="wrong_entry_header">Wrong entry</label>
    </languageKey>
    <languageKey index="de" type="array">
      <label index="actual">aktuell</label>
      <label index="apply">anlegen</label>
      <label index="back">Zurück</label>
      <label index="cancel">Abbrechen</label>
      <label index="change1">ändern</label>
      <label index="change2">wechseln</label>
      <label index="delete">löschen</label>
      <label index="existing">bestehend</label>
    </languageKey>
  </data>
</T3locallang>

```

```
<label index="from">Von</label>
<label index="language">de</label>
<label index="more">Weiter</label>
<label index="new">neu</label>
<label index="to">Bis</label>
<label index="year">Jahr</label>
<label index="administration_bookings_actual">Buchungen ab heute</label>
<label index="administration_bookings_year">Buchungen Jahr</label>
<label index="administration_date_format">Datumsformat: tt.mm.jjjj</label>
<label index="administration_header">Vermietung</label>
<label index="administration_new">Neuer Eintrag</label>
<label index="administration_none">Keine Einträge vorhanden.</label>
<label index="administration_years">Buchungen von ganzen Jahren</label>
<label index="calendar_ifformat">%d.%m.%Y</label>
<label index="exists_header">Der Datensatz wurde nicht erstellt, da er bereits
existiert.</label>
<label index="overlap_header">Einträge überschneiden sich</label>
<label index="pi1_comes_after">kommt nach</label>
<label index="pi1_erroneous">ist fehlerhaft</label>
<label index="pi1_has_an_error">hat einen Fehler</label>
<label index="pi1_has_error">hat den Fehler</label>
<label index="to_be_changed_header">Eintrag ändern</label>
<label index="to_be_deleted_header">Folgender Eintrag soll gelöscht werden</label>
<label index="was_created_header">Der Datensatz wurde erstellt.</label>
<label index="was_deleted_header">Der Datensatz wurde gelöscht.</label>
<label index="wrong_entry_header">Falscher Eintrag</label>
</languageKey>
</data>
</T3locallang>
```

*pi2/locallang.xml:*

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
  <meta type="array">
    <type>module</type>
    <description>Language labels for plugin &quot;tx_netcosrenting_pi2&quot;</description>
  </meta>
  <data type="array">
    <languageKey index="default" type="array">
      <label index="listFieldHeader_from">From</label>
      <label index="listFieldHeader_to">To</label>
      <label index="list_mode_1">Mode 1</label>
      <label index="list_mode_2">Mode 2</label>
      <label index="list_mode_3">Mode 3</label>
      <label index="back">Back</label>
      <label index="pi_list_browserresults_prev">&lt; Previous</label>
      <label index="pi_list_browserresults_page">Page</label>
      <label index="pi_list_browserresults_next">Next &gt;</label>
      <label index="pi_list_browserresults_displays">Displaying results
        ###SPAN_BEGIN###%s to %s&lt;/span&gt; out of
        ###SPAN_BEGIN###%s&lt;/span&gt;</label>
      <label index="pi_list_searchBox_search">Search</label>

      <label index="booked">booked</label>
      <label index="free">free</label>
      <label index="year">year</label>

      <label index="previous_month">Previous Month</label>
      <label index="actual_month">Actual Month</label>
      <label index="next_month">Next Month</label>

      <label index="january">January</label>
      <label index="february">February</label>
      <label index="march">March</label>
      <label index="april">April</label>
      <label index="may">May</label>
      <label index="june">June</label>
      <label index="july">July</label>
      <label index="august">August</label>
      <label index="september">September</label>
      <label index="october">October</label>
      <label index="november">November</label>
      <label index="december">December</label>

      <label index="calendar_week">calendar week</label>

      <label index="monday">Monday</label>
      <label index="tuesday">Tuesday</label>
      <label index="wednesday">Wednesday</label>
      <label index="thursday">Thursday</label>
      <label index="friday">Friday</label>
    </languageKey>
  </data>
</T3locallang>
```

<label index="saturday">Saturday</label>

<label index="sunday">Sunday</label>

<label index="booked\_change">change month</label>

<label index="booked\_future">Choose future month</label>

<label index="booked\_header">Renting</label>

<label index="error\_back">back to the actual month</label>

<label index="error\_header">No Access to Month</label>

<label index="error\_text">The oldest possible month is the previous month.

No limits for the future.</label>

</languageKey>

<languageKey index="de" type="array">

<label index="booked">gebucht</label>

<label index="free">frei</label>

<label index="year">Jahr</label>

<label index="previous\_month">Voriger Monat</label>

<label index="actual\_month">Aktueller Monat</label>

<label index="next\_month">Nächster Monat</label>

<label index="january">Januar</label>

<label index="february">Februar</label>

<label index="march">März</label>

<label index="april">April</label>

<label index="may">Mai</label>

<label index="june">Juni</label>

<label index="july">Juli</label>

<label index="august">August</label>

<label index="september">September</label>

<label index="october">Oktober</label>

<label index="november">November</label>

<label index="december">Dezember</label>

<label index="calendar\_week">Kalenderwoche</label>

<label index="monday">Montag</label>

<label index="tuesday">Dienstag</label>

<label index="wednesday">Mittwoch</label>

<label index="thursday">Donnerstag</label>

<label index="friday">Freitag</label>

<label index="saturday">Samstag</label>

<label index="sunday">Sonntag</label>

<label index="booked\_change">Monat wechseln</label>

<label index="booked\_future">Zukünftigen Monat wählen:</label>

<label index="booked\_header">Vermietung</label>

<label index="error\_back">Zurück zum aktuellen Monat</label>

<label index="error\_header">Kein Zugriff auf den Monat</label>

```
<label index="error_text">In die Vergangenheit kann höchstens auf den  
  Vormonat zugegriffen werden. Für die Zukunft bestehen keine  
  Einschränkungen. </label>  
</languageKey>  
</data>  
</T3locallang>
```

## 4. PHP-Quellcode anpassen

Die Hauptänderungen geschehen in jedem Plugin-Verzeichnis *typo3conf/ext/netcos\_renting/pi?* in der Datei *class.tx\_netcosrenting\_pi?.php*. Dabei steht ? für 1 oder 2, da wir zwei Plugins eingerichtet haben. Diese Dateien bilden die Erweiterungsklassen für **tslib\_pibase**. Die Methode **main** darin ruft den ganzen Darstellungsprozess auf.

### 4.1 Datenbankzugriff

Normalerweise greift TYPO3 auf die Datenbank *MySQL* zu. Doch es wurde eine allgemeine Schnittstelle kreiert, die auf beliebige Datenbanken zugreifen kann. Dazu wurden neue Funktionen geschaffen, die die bekannten *MySQL*-Befehle von PHP ersetzen.

Wir verwenden diese Schnittstelle, obwohl wir z.Z. nur *auf MySQL* zugreifen. Dazu muss die TYPO3-Extension *dbal* installiert werden. Dokumentation zu den zugehörigen Datenbankfunktionen befindet sich in der [TYPO3 Core Documentation, Kapitel 1.4](#).

Wir verwenden folgende Variable:

```
$table_name = 'tx_netcosrenting_fromto';
```

In den Plugins werden folgende Datenbankzugriffe genutzt:

```
$where = 'uid='.$$uid;
$GLOBALS['TYPO3_DB']->exec_DELETEquery($table_name, $where);
```

Dies löscht aus der Tabelle *\$table\_name* den Eintrag mit der uid *\$uid*.

```
$query = 'SELECT uid FROM '.$table_name .
' WHERE from_date='.$$f->string().' AND to_date='.$$t->string();
$res = $GLOBALS['TYPO3_DB']->sql_query($query);
if ($GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
    ...
} else {
    $fields_values = array('from_date' => $f->string(), 'to_date' => $t->string());
    $GLOBALS['TYPO3_DB']->exec_INSERTquery($table_name, $fields_values);
    ...
}
```

Dies ist ein SELECT-Request, der durch *sql\_query()* aufgerufen wird. Das Ergebnis **\$res** wird durch *sql\_fetch\_assoc()* in assoziative Arrays aufgespalten. Falls kein passender Datensatz existiert, wird ein neuer Datensatz aus den Werten erzeugt und in der Datentabelle abgespeichert durch *exec\_INSERTquery()*.

### 4.2 Erstes Plugin (Vermietungsverwaltung)

In dieser Sektion stellen wir die Datei *typo3conf/ext/netcos\_renting/pi1/class.tx\_netcosrenting\_pi1.php* vor. Da sie sehr lang ist, haben wir sie in viele Teile aufgeteilt, die alle zusammengesetzt werden müssen. Sie können natürlich die gesamte Datei aus dem Quellcode der Extension herausziehen.

Die Zeilenumbrüche müssen jeweils entfernt werden.

Start und Lizenz:

```

<?php
/*****
* Copyright notice
*
* (c) 2007,2008 Bernd Warken <be.warken@netcos.de>
* All rights reserved
*
* This script is part of the TYPO3 project. The TYPO3 project is
* free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* The GNU General Public License can be found at
* http://www.gnu.org/copyleft/gpl.html.
*
* This script is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* This copyright notice MUST APPEAR in all copies of the script!
*****/

```

Einbettung von PHP-Dateien:

```

require_once(PATH_tlib.'class.tlib_pibase.php');
require_once(t3lib_extMgm::extPath('netcos_renting').'class_termin.php');
require_once(t3lib_extMgm::extPath('smarty').'class.tx_smarty.php');
require_once(t3lib_extMgm::extPath('smarty').'Smarty/libs/Smarty.class.php');

```

Dokumentation von *kickstarter*:

```

/**
* Plugin 'own Booking Administration' for the 'eigene_buchungen' extension.
*
* @author Bernd Warken <be.warken@netcos.de>
* @package TYPO3
* @subpackage tx_eigenebuchungen
*/

```

Klassendefinition und Variablen von *kickstarter*:

```

class tx_netcosrenting_pi1 extends tlib_pibase {
var $prefixId = 'tx_netcosrenting_pi1'; // Same as class name
var $scriptRelPath = 'pi1/class.tx_netcosrenting_pi1.php';
// Path to this script relative to the extension dir.
var $extKey = 'netcos_renting'; // The extension key.

```

Stellt sicher, dass die Daten von HTML-Forms übertragen werden (POST und GET):

```
var $pi_checkCHash = true;
```

Dokumentation von *kickstarter* für function main:

```
/**  
 * Main method of your PlugIn  
 *  
 * @param string $content: The content of the PlugIn  
 * @param array $conf: The PlugIn Configuration  
 * @return The content that should be displayed on the website  
 */  
function main($content,$conf) {
```

Ab jetzt befinden wir uns innerhalb der Funktion *main*.

Speichere die Datensicherung des vorigen Durchgangs in \$POST:

```
global $_POST, $LANG;  
$POST = t3lib_div::_POST();  
$_POST = array();
```

Aktiviere die Mehrsprachigkeit:

```
$this->pi_loadLL();
```

Allgemeine Konfiguration für *smarty*:

```
$this->smarty = new Smarty;  
$this->smarty->compile_dir = t3lib_extMgm::extPath($this->extKey).'tmpl_c';  
$this->smarty->template_dir = t3lib_extMgm::extPath($this->extKey).'tmpl';  
$this->smarty->left_delimiter = '{..';  
$this->smarty->right_delimiter = '..}';
```

Das legt fest, dass man in den HTML-Template-Dateien von *smarty* Variablen durch *{..\$var..}* benutzen kann. Der Wert für diese Variablen wird in PHP mit *\$this->smarty->assign* eingestellt.

Die verwendeten *smarty*-HTML-Template-Dateien:

```
$this->templates['administration'] = 'tmpl/pi1/administration.tpl';  
$this->templates['wrong_entry'] = 'tmpl/pi1/wrong_entry.tpl';  
$this->templates['overlap'] = 'tmpl/pi1/overlap.tpl';  
$this->templates['to_be_changed'] = 'tmpl/pi1/to_be_changed.tpl';  
$this->templates['to_be_deleted'] = 'tmpl/pi1/to_be_deleted.tpl';  
$this->templates['exists'] = 'tmpl/pi1/exists.tpl';  
$this->templates['was_created'] = 'tmpl/pi1/was_created.tpl';  
$this->templates['was_deleted'] = 'tmpl/pi1/was_deleted.tpl';
```

*Smarty*-Abkürzung:

```
$this->smarty->assign('DIR_JSCALENDAR', '/typo3conf/ext/'.$this->extKey.'/jscalendar-1.0/');
```

Name der MySQL-Tabelle des Plugins:

```
$table_name = 'tx_netcosrenting_fromto';
```

Heutiges Datum:

```
$heutetermin = new Termin((string)date('d').!(string)date('n').!(string)date('Y'));
```

Ersetze im heutigen Termin den Tag durch 01:

```
$erster_des_Monats = $heutetermin->erster_des_Monats();
```

*Smarty*-Variablen:

```
$this->smarty->assign('ACTUAL', $this->pi_getLL('actual'));
$this->smarty->assign('APPLY', $this->pi_getLL('apply'));
$this->smarty->assign('CANCEL', $this->pi_getLL('cancel'));
$this->smarty->assign('CHANGE1', $this->pi_getLL('change1'));
$this->smarty->assign('CHANGE2', $this->pi_getLL('change2'));
$this->smarty->assign('DELETE', $this->pi_getLL('delete'));
$this->smarty->assign('EXISTING', $this->pi_getLL('existing'));
$this->smarty->assign('FROM', $this->pi_getLL('from'));
$this->smarty->assign('MORE', $this->pi_getLL('more'));
$this->smarty->assign('NEW', $this->pi_getLL('new'));
$this->smarty->assign('TO', $this->pi_getLL('to'));
$this->smarty->assign('YEAR', $this->pi_getLL('year'));
$this->smarty->assign('ADMINISTRATION_DATE_FORMAT',
    $this->pi_getLL('administration_date_format'));
$this->smarty->assign('ADMINISTRATION_HEADER',
    $this->pi_getLL('administration_header'));
$this->smarty->assign('ADMINISTRATION_NEW', $this->pi_getLL('administration_new'));
$this->smarty->assign('ADMINISTRATION_YEARS',
    $this->pi_getLL('administration_years'));
$this->smarty->assign('CALENDAR_IFFORMAT', $this->pi_getLL('calendar_iformat'));
$this->smarty->assign('EXISTS_HEADER', $this->pi_getLL('exists_header'));
$this->smarty->assign('OVERLAP_HEADER', $this->pi_getLL('overlap_header'));
$this->smarty->assign('TO_BE_CHANGED_HEADER',
    $this->pi_getLL('to_be_changed_header'));
$this->smarty->assign('TO_BE_DELETED_HEADER',
    $this->pi_getLL('to_be_deleted_header'));
$this->smarty->assign('WAS_CREATED_HEADER', $this->pi_getLL('was_created_header'));
$this->smarty->assign('WAS_DELETED_HEADER', $this->pi_getLL('was_deleted_header'));
$this->smarty->assign('WRONG_ENTRY_HEADER',
    $this->pi_getLL('wrong_entry_header'));
```

Durch *pi\_getLL* wird auf die Mehrsprachigkeit zurückgegriffen. Die Begriffe werden für dieses erste Plugin in *pi/locallang.xml* definiert.

Konfiguration für den Kalender:

```

$dir_jscalendar = '/typo3conf/ext/netcos_jscalendar/jscalendar/';
$import_jscalendar = '
<style type="text/css">@import url('.$dir_jscalendar.'calendar-green.css);</style>
<script type="text/javascript" src="'.$dir_jscalendar.'calendar.js"></script>
<script type="text/javascript" src="'.$dir_jscalendar.'calendar-setup.js"></script>
<script type="text/javascript" src="'.$dir_jscalendar.'lang/calendar-en.js"></script>';
;
$lang = $this->pi_getLL('language');
if ($lang == 'en') {
    $date_lang = 'date_en';
} else {
    $date_lang = 'Datum';
    $import_jscalendar .= '<script type="text/javascript" src="'.
        $dir_jscalendar .'lang/calendar-' . $lang . '.js"></script>';
}
$this->smarty->assign('IMPORT_JSCALENDAR', $import_jscalendar);

```

Dabei wird die deutsche Datei nur dann eingebettet, wenn Deutsch die aktuelle Sprache ist.

Die folgenden Abschnitte der Form **if (\$POST['...']) {...}** fangen POST-Parameter aus früheren Läufen ab. Es werden eventuell *smarty*-Abkürzungen durch **assign** gesetzt. Der Befehl **fetch** ruft eine *smarty*-Template-Datei auf. Deren Ergebnis **\$content** schließt diesen Lauf durch

```
return $this->pi_wrapInBaseClass($content);
```

ab.

```

if ($POST['to_be_deleted']) {
    $this->smarty->assign('UID', $POST['uid']);
    $this->smarty->assign('VALUE_FROM', $POST['data_from']);
    $this->smarty->assign('VALUE_TO', $POST['data_to']);
    $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
        $this->templates['to_be_deleted']);
    return $this->pi_wrapInBaseClass($content);
}

if ($POST['to_be_changed']) {
    $this->smarty->assign('UID', $POST['uid']);
    $this->smarty->assign('VALUE_FROM', $POST['data_from']);
    $this->smarty->assign('VALUE_TO', $POST['data_to']);
    $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
        $this->templates['to_be_changed']);
    return $this->pi_wrapInBaseClass($content);
}

```

Hier wird der Datensatz mit einer gegebenen UID aus der Tabelle gelöscht:

```

if ($POST['delete']) {
  $uid = $POST['uid'];
  if ($uid) {
    $where = 'uid='.$uid;
    $GLOBALS['TYPO3_DB']->exec_DELETEquery($table_name, $where);
    $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
      $this->templates['was_deleted']);
    return $this->pi_wrapInBaseClass($content);
  }
}

```

Hier wird ebenfalls der Datensatz mit einer gegebenen UID aus der Tabelle gelöscht und die Variable *\$run\_apply* konfiguriert für den nächsten Abschnitt:

```

$run_apply = FALSE;
if ($POST['change']) {
  $uid = $POST['uid'];
  if ($uid) {
    $where = 'uid='.$uid;
    $GLOBALS['TYPO3_DB']->exec_DELETEquery($table_name, $where);
  }
  $run_apply = TRUE;
}

```

Hier wird ein neuer Datensatz angelegt:

```

if ($POST['apply'] || $run_apply) {
    $richtig = TRUE;
    $errmsg = "";
    $df = $POST['data_from'];
    if ($df) {
        $f = new Termin($df);
        if ($f->is_true()) {

        } else {
            $richtig = FALSE;
            $errmsg .= "".$this->pi_getLL("from")."".' '.$this->pi_getLL("pi1_erroneous").'.';
            $f->get_error();
        }
    } else {
        $richtig = FALSE;
        $errmsg .= "".$this->pi_getLL("from")."".' '.$this->pi_getLL("pi1_erroneous").'.';
    }
    $dt = $POST['data_to'];
    if ($dt) {
        $t = new Termin($dt);
        if ($t->is_true()) {

        } else {
            $richtig = FALSE;
            $errmsg .= "".$this->pi_getLL("to")."".' '.$this->pi_getLL("pi1_erroneous").'.';
            $t->get_error();
        }
    } else {
        $richtig = FALSE;
        $errmsg .= "".$this->pi_getLL("to")."".' '.$this->pi_getLL("pi1_erroneous").'.';
    }
}
if ($richtig) {
    $query = 'SELECT uid FROM '.$table_name .
        ' WHERE from_date='.$f->string().' AND to_date='.$t->string();
    $res = $GLOBALS['TYPO3_DB']->sql_query($query);
    if ($GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
        $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
            $this->templates['exists']);
        return $this->pi_wrapInBaseClass($content);
    } else {
        $fields_values = array('from_date' => $f->string(), 'to_date' => $t->string());
        $GLOBALS['TYPO3_DB']->exec_INSERTquery($table_name, $fields_values);
        $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
            $this->templates['was_created']);
        return $this->pi_wrapInBaseClass($content);
    }
} else {
    $this->smarty->assign('ERRMSG', $errmsg);
    $this->smarty->assign('UID', $uid);
    $this->smarty->assign('VALUE_FROM', $df);
}

```

```
$this->smarty->assign('VALUE_TO', $dt);
$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['wrong_entry']);
return $this->pi_wrapInBaseClass($content);
}
}
```

Teste, ob Jahreswechsel in der Darstellung ansteht:

```
$jahr = 0;
if ($POST['wechseln']) {
    $jahr = $POST['jahr'];
    if ($jahr < 100) $jahr += 2000;
    if ($jahr >= 3000 || $jahr < 2000) $jahr = 0;
}
```

Suche aus der Datenbank alle gebuchten Termine:

```

$query = 'SELECT uid,from_date,to_date FROM '.$table_name;
$res = $GLOBALS['TYPO3_DB']->sql_query($query);
$fromto = array();
$from = array();
while ($row = $GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
    $u = $row["uid"];
    $f = new Termin($row["from_date"]);
    $t = new Termin($row["to_date"]);
    $act = FALSE;
    $errmsg = "";

    if ($f->is_false() || $t->is_false()) {
        $value_from = "";
        $value_to = "";
        $act = TRUE;

        if ($f->is_false()) {
            $msg = $f->get_error();
            if ($msg) {
                $errmsg .= ' "' . $this->pi_getLL("from") . "' ' . $this->pi_getLL("pi1_has_error") . "': '$msg';
            } else {
                $errmsg .= ' "' . $this->pi_getLL("from") . "' ' . $this->pi_getLL("pi1_has_an_error") . "':!";
            }
        } else {
            $value_from = $f->$date_lang();
        }

        if ($t->is_false()) {
            $msg = $t->get_error();
            if ($msg) {
                $errmsg .= ' "' . $this->pi_getLL("to") . "' ' . $this->pi_getLL("pi1_has_error") . "': '$msg';
            } else {
                $errmsg .= ' "' . $this->pi_getLL("to") . "' ' . $this->pi_getLL("pi1_has_an_error") . "':!";
            }
        } else {
            $value_to = $t->$date_lang();
        }
    } else {
        if ($f->int() > $t->int()) {
            $value_from = $f->$date_lang();
            $value_to = $t->$date_lang();
            $errmsg = '" . $this->pi_getLL("from") . "' ' . $this->pi_getLL("pi1_comes_after") . "': '" .
                $this->pi_getLL("to") . "':!";
            $act = TRUE;
        }
    }
}
if ($act) {
    $this->smarty->assign('ERRMSG', $errmsg);
    $this->smarty->assign('UID', $u);
    $this->smarty->assign('VALUE_FROM', $value_from);
    $this->smarty->assign('VALUE_TO', $value_to);
}

```

```

$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['wrong_entry']);
return $this->pi_wrapInBaseClass($content);
}
$from[$u] = $f;
$fromto[$u] = array("from_date" => $f, "to_date" => $t);
}
asort($from);
$sortiert = array_keys($from);

```

Stelle Fehler fest, wenn sich 2 Datensätze überschneiden:

```

$t0 = 0;
$f0 = 0;
$u0 = 0;
foreach ($sortiert as $uid) {
    $f = $from[$uid];
    $f1 = $f->int();
    $t = $fromto[$uid]['to_date'];
    $t1 = $t->int();

    if ($t0 > $f1) {
        $this->smarty->assign('UID1', $u0);
        $this->smarty->assign('UID2', $uid);
        $this->smarty->assign('VALUE_FROM1', $from[$u0]->$date_lang());
        $this->smarty->assign('VALUE_TO1', $fromto[$u0]['to_date']->$date_lang());
        $this->smarty->assign('VALUE_FROM2', $f->$date_lang());
        $this->smarty->assign('VALUE_TO2', $t->$date_lang());
        $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
            $this->templates['overlap']);
        return $this->pi_wrapInBaseClass($content);
    }

    $f0 = $f1;
    $t0 = $t1;
    $u0 = $uid;
}

```

Filtere die passenden Buchungstermine aus:

```

$aktuell_sortiert = array();
if ($jahr) {
    $this->smarty->assign('BOOKINGS_TITLE',
        $this->pi_getLL('administration_bookings_year').'.$jahr);
    $erster = new Termin('1.1.'.$jahr);
    $letzter = new Termin('31.12.'.$jahr);
    foreach ($sortiert as $uid) {
        $f = $from[$uid];
        $t = $fromto[$uid]['to_date'];
        if ( ($f >= $erster && $f <= $letzter) ||
            ($t >= $erster && $t <= $letzter) ||
            ($f < $erster && $t > $letzter)
        ) array_push($aktuell_sortiert, $uid);
    }
} else {
    $this->smarty->assign('BOOKINGS_TITLE',
        $this->pi_getLL('administration_bookings_actual'));
    foreach ($sortiert as $uid) {
        if ($fromto[$uid]['to_date'] > $heutetermin) array_push($aktuell_sortiert, $uid);
    }
}
}

```

Erstelle Monatsdiagramm:

```

if (count($aktuell_sortiert) == 0) {
    $aktuelle_buchungen = $this->pi_getLL('administration_none');
} else {
    $aktuelle_buchungen = '<table width="350px"><tr align="center">
        <th>UID</th><th>'. $this->pi_getLL("from").'</th><th>'.
        $this->pi_getLL("to").'</th><th></th><th></th></tr>';
    foreach ($aktuell_sortiert as $uid) {
        $aktuelle_buchungen .= '<tr>
            <form method="post">
            <td style="background-color: yellow;">
                <input type="hidden" name="uid" value="'. $uid.' />'. $uid.'
            </td>
            <td style="background-color: LightGreen;">
                <input type="hidden" name="data_from" value="'. $from[$uid]->$date_lang().' />'.
                $from[$uid]->$date_lang().'</td>
            <td style="background-color: yellow;">
                <input type="hidden" name="data_to" value="'. $fromto[$uid]["to_date"]->$date_lang().
                ' />'. $fromto[$uid]["to_date"]->$date_lang().'</td>
            <td><input type="submit" name="to_be_deleted" value="'. $this->pi_getLL("delete").'" />
            </td>
            <td><input type="submit" name="to_be_changed" value="'. $this->pi_getLL("change1").
            '" /></td>
            <td></td>
            </form></tr>';
    }
    $aktuelle_buchungen .= '</table>';
}
$this->smarty->assign('AKTUELLES_JAHR', $heutetermin->Jahr());
$this->smarty->assign('AKTUELLE_BUCHUNGEN', $aktuelle_buchungen);
$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['administration']);
return $this->pi_wrapInBaseClass($content);

```

Ende der Datei:

```

    }
}

if (defined('TYPO3_MODE') && $TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
    ['ext/netcos_renting/pi1/class.tx_netcosrenting_pi1.php']) {
    include_once($TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
        ['ext/netcos_renting/pi1/class.tx_netcosrenting_pi1.php']);
}
?>

```

### 4.3 Zweites Plugin (vermietet)

In dieser Sektion stellen wir die Datei `typo3conf/ext/netcos_renting/pi2/class.tx_netcosrenting_pi2.php` vor. Da sie sehr lang ist, haben wir

sie in viele Teile aufgeteilt, die alle zusammengesetzt werden müssen. Sie können sie jedoch auch aus dem Quellcode der Extension herausziehen.

Die Zeilenumbrüche müssen jeweils entfernt werden.

Start und Lizenz:

```
<?php
/*****
* Copyright notice
*
* (c) 2007,2008 Bernd Warken <be.warken@netcos.de>
* All rights reserved
*
* This script is part of the TYPO3 project. The TYPO3 project is
* free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* The GNU General Public License can be found at
* http://www.gnu.org/copyleft/gpl.html.
*
* This script is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* This copyright notice MUST APPEAR in all copies of the script!
*****/
```

Einbettung von PHP-Dateien:

```
require_once(PATH_tslib.'class.tslib_pibase.php');
require_once('class_termin.php');
require_once(t3lib_extMgm::extPath('smarty').'class.tx_smarty.php');
require_once(t3lib_extMgm::extPath('smarty').'Smarty/libs/Smarty.class.php');
```

*Kickstarter*-Dokumentation, Klassendefinition, verwendete Variablen:

```

/**
 * Plugin 'vermietet' for the 'netcos_renting' extension.
 *
 * @author Bernd Warken <be.warken@netcos.de>
 * @package TYPO3
 * @subpackage tx_netcosrenting
 */
class tx_netcosrenting_pi2 extends tslib_pibase {
    var $prefixId = 'tx_netcosrenting_pi2'; // Same as class name
    var $scriptRelPath = 'pi2/class.tx_netcosrenting_pi2.php';
    // Path to this script relative to the extension dir.
    var $extKey = 'netcos_renting'; // The extension key.

    var $pi_checkCHash = true;

```

Methode *main()*:

```

/**
 * Main method of your PlugIn
 *
 * @param string $content: The content of the PlugIn
 * @param array $conf: The PlugIn Configuration
 * @return The content that should be displayed on the website
 */
function main($content,$conf) {
    $this->conf=$conf;
    $this->pi_setPiVarDefaults();
    $this->pi_loadLL();

```

Speichere die Datensicherung des vorigen Durchgangs in \$POST:

```

global $_POST;
$POST = t3lib_div::_POST();
$_POST = array();

```

Aktiviere die Mehrsprachigkeit:

```

$this->pi_loadLL();

```

*Smarty*-Konfiguration:

```

$this->smarty = new Smarty;
$this->smarty->compile_dir = t3lib_extMgm::extPath($this->extKey).'tmpl_c';
$this->smarty->template_dir = t3lib_extMgm::extPath($this->extKey).'tmpl';
$this->smarty->left_delimiter = '{..';
$this->smarty->right_delimiter = '..}';

$this->templates['booked'] = 'tmpl/pi2/booked.tpl';
$this->templates['error'] = 'tmpl/pi2/error.tpl';

```

Name der MySQL-Tabelle:

```
$stable_name = 'tx_netcosrenting_fromto';
```

Smarty-Variablen für die HTML-Template-Dateien:

```
$this->smarty->assign('PREVIOUS_MONTH', $this->pi_getLL('previous_month'));
$this->smarty->assign('ACTUAL_MONTH', $this->pi_getLL('actual_month'));
$this->smarty->assign('NEXT_MONTH', $this->pi_getLL('next_month'));

$this->smarty->assign('MONDAY', $this->pi_getLL('monday'));
$this->smarty->assign('TUESDAY', $this->pi_getLL('tuesday'));
$this->smarty->assign('WEDNESDAY', $this->pi_getLL('wednesday'));
$this->smarty->assign('THURSDAY', $this->pi_getLL('thursday'));
$this->smarty->assign('FRIDAY', $this->pi_getLL('friday'));
$this->smarty->assign('SATURDAY', $this->pi_getLL('saturday'));
$this->smarty->assign('SUNDAY', $this->pi_getLL('sunday'));

$this->smarty->assign('CALENDAR_WEEK', $this->pi_getLL('calendar_week'));

$this->smarty->assign('BOOKED_CHANGE', $this->pi_getLL('booked_change'));
$this->smarty->assign('BOOKED_FUTURE', $this->pi_getLL('booked_future'));
$this->smarty->assign('BOOKED_HEADER', $this->pi_getLL('booked_header'));

$this->smarty->assign('ERROR_BACK', $this->pi_getLL('error_back'));
$this->smarty->assign('ERROR_HEADER', $this->pi_getLL('error_header'));
$this->smarty->assign('ERROR_TEXT', $this->pi_getLL('error_text'));
```

Variablen zu aktuellem Datum:

```
$aktuelles_jahr = date('Y');
$aktueller_monat = date('n');
$heutetermin = new Termin((string)date('d').'.'. (string)date('n').'.'. (string)date('Y'));
$monate = array(0, $this->pi_getLL("january"), $this->pi_getLL("february"),
    $this->pi_getLL("march"), $this->pi_getLL("april"), $this->pi_getLL("mai"),
    $this->pi_getLL("june"), $this->pi_getLL("july"), $this->pi_getLL("august"),
    $this->pi_getLL("september"), $this->pi_getLL("october"),
    $this->pi_getLL("november"), $this->pi_getLL("december"));
```

Variablen aus POST-Werten vom vorigen Lauf:

```
$monat = $POST['Monat'];
$jahr = $POST['Jahr'];
```

Stelle auf passenden Monat um:

```

Saktuell_text = "";
if ( (int)$monat > 0 && (int)$jahr > 0 ) {
    if ($jahr == $saktuelles_jahr && $monat == $saktueller_monat)
        Saktuell_text = $this->pi_getLL('actual_month').': ';
    else {
        $vormonat = $saktueller_monat - 1;
        $vormonat_jahr = $saktuelles_jahr;
        if ($vormonat == 0) {
            $vormonat = 12;
            $vormonat_jahr -= 1;
        }
        if ($jahr < $vormonat_jahr ||
            ($jahr == $vormonat_jahr && $monat < $vormonat)) {
            # Verbot für Monate vor dem Vormonat
            $this->smarty->assign('ZEIT', $monate[$monat].'$jahr');
            $this->smarty->assign('AKTUELLES_JAHR', $saktuelles_jahr);
            $this->smarty->assign('AKTUELLER_MONAT', $saktueller_monat);
            $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
                $this->templates['error']);
            $res = $this->pi_wrapInBaseClass($content);
            return $res;
        }
        if ($jahr == $vormonat_jahr && $monat == $vormonat)
            Saktuell_text = $this->pi_getLL('previous_month').': ';
        else {
            $folgemonat = $saktueller_monat + 1;
            $folgemonat_jahr = $saktuelles_jahr;
            if ($folgemonat == 13) {
                $folgemonat = 1;
                $folgemonat_jahr += 1;
            }
            if ($jahr == $folgemonat_jahr && $monat == $folgemonat)
                Saktuell_text = $this->pi_getLL('next_month').': ';
        }
    }
} else {
    switch($monat) {
    case $this->pi_getLL('previous_month'):
        if ($saktueller_monat == 1) {
            $monat = 12;
            $jahr = $saktuelles_jahr - 1;
        } else {
            $monat = $saktueller_monat - 1;
            $jahr = $saktuelles_jahr;
        }
        Saktuell_text = $this->pi_getLL('previous_month').': ';
        break;
    case $this->pi_getLL('next_month'):
        if ($saktueller_monat == 12) {
            $monat = 1;
            $jahr = $saktuelles_jahr + 1;
        }
    }
}

```

```
} else {  
    $monat = $aktueller_monat + 1;  
    $jahr = $aktuelles_jahr;  
}  
$aktuell_text = $this->pi_getLL('next_month').': '  
break;  
default:  
    $jahr = $aktuelles_jahr;  
    $monat = $aktueller_monat;  
    $aktuell_text = $this->pi_getLL('actual_month').': '  
}  
}
```

Erstelle Display zum ausgesuchten Monat:

```

$termin_erster = new Termin('1.'.$monat.'.$jahr);
$stei = $termin_erster->int();
$max_tage = (string)($termin_erster->max_Tage_des_Monats());
$termin_letzter = new Termin($max_tage.'.$monat.'.$jahr);
$tli = $termin_letzter->int();
$stage_gebucht = array();
for ($i = 0; $i <= $max_tage; $i++) $stage_gebucht[$i] = 0;
$query = 'SELECT from_date,to_date ' .
'FROM '.$table_name.' ' .
'WHERE (from_date>='.$stei.' AND from_date<='.$tli.') OR ' .
'(to_date>='.$stei.' AND to_date<='.$tli.') OR ' .
'(from_date<='.$stei.' AND to_date>='.$tli.)';
$res = $GLOBALS['TYPO3_DB']->sql_query($query);
while ($row = $GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
    $v = new Termin($row['from_date']);
    $b = new Termin($row['to_date']);

    if ($v->is_true() && $b->is_true()) {
        $vi = $v->int();
        $vti = (int)$v->Tag();
        $bi = $b->int();
        $bti = (int)$b->Tag();
        if ($vi <= $stei && $bi >= $tli) {
            for ($i=1; $i <= $max_tage; $i++) $stage_gebucht[$i] = 1;
            break;
        } elseif ($vi <= $stei && $bi <= $tli) {
            for ($i=1; $i <= $bti; $i++)
                $stage_gebucht[$i] = 1;
        } elseif ($vi >= $stei && $bi <= $tli) {
            for ($i=$vti; $i <= $bti; $i++)
                $stage_gebucht[$i] = 1;
        } elseif ($vi >= $stei && $bi > $tli) {
            for ($i=$vti; $i <= $max_tage; $i++)
                $stage_gebucht[$i] = 1;
        }
    }
}

$ekw = $termin_erster->get_Kalenderwoche();
$ewz = $termin_erster->get_Wochentag_Zahl();
$lkw0 = $termin_letzter->get_Kalenderwoche();
if ($lkw0 == 1) $lkw = 53;
else $lkw = $lkw0;
$letzter_tag = $termin_letzter->Tag();

$a = array();
for ($j = 1; $j < $ewz; $j++) {
    $a[$ekw][$j] = 0;
}
$stag = 1;

```

```

for ($j = $ewz; $j <= 7; $j++) {
    $a[$ekw][$j] = $tag;
    $tag++;
}
for ($i = $ekw + 1; $i <= $lkw - 1; $i++) {
    for ($j = 1; $j <= 7; $j++) {
        $a[$i][$j] = $tag;
        $tag++;
    }
}
for ($j = 1; $tag <= $letzter_tag; $j++) {
    $a[$lkw][$j] = $tag;
    $tag++;
}
for ($j; $j <= 7; $j++) {
    $a[$lkw][$j] = 0;
}

$a2 = array();
for ($i = $ekw; $i <= $lkw; $i++) {
    if ($i == 53) $i0 = $lkw0;
    else $i0 = $i;
    $a2[$i][0] = '<tr><td align="center" style="background-color: #FFFFCC;">'. $i0. '</td>';
    for ($j = 1; $j <= 7; $j++) {
        if ($a[$i][$j] == 0) $a2[$i][$j] = '<td></td>';
        else {
            if ($stage_gebucht[$a[$i][$j]])
                $a2[$i][$j] = '<td align="center" style="background-color: LightGreen;">'. $a[$i][$j].
                    '<br />'. $this->pi_getLL("booked"). '</td>';
            else
                $a2[$i][$j] = '<td align="center" style="background-color: yellow;">'. $a[$i][$j]. '<br />'.
                    $this->pi_getLL("free"). '</td>';
        }
    }
    $a2[$i][7] .= '</tr>';
}

$kalender_monat = "";
for ($i = $ekw; $i <= $lkw; $i++) {
    for ($j = 0; $j <= 7; $j++) {
        $kalender_monat .= $a2[$i][$j];
    }
}

$monatsradio = "";
for ($i = 1; $i <= 12; $i++) {
    $checked = "";
    if ($i == $aktueller_monat) $checked = ' checked="checked"';
}

```

```

$monatsradio .= '
    <input name="Monat" type="radio" value="'. $i. '" .
    $checked.'>'. $monate[$i]. '</input>';
}

$this->smarty->assign('AKTUELL_TEXT', $aktuell_text);
$this->smarty->assign('MONAT', $monate[$monat]);
$this->smarty->assign('JAHR', $jahr);
$this->smarty->assign('BUCHUNGEN', $tag_str);
$this->smarty->assign('KALENDER_MONAT', $kalender_monat);
$this->smarty->assign('MONATSRADIO', $monatsradio);
$this->smarty->assign('AKTUELLES_JAHR', $aktuelles_jahr);

$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['booked']);
$res = $this->pi_wrapInBaseClass($content);
return $res;

```

Dateiende:

```

    }
}
if (defined('TYPO3_MODE') && $TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
['ext/netcos_renting/pi2/class.tx_netcosrenting_pi2.php']) {
    include_once($TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
['ext/netcos_renting/pi2/class.tx_netcosrenting_pi2.php']);
}
?>

```

## 5. HTML-Templates

Diese Templates entstehen durch *smarty*. Es sind HTML-Dateien, die zusätzlich Abkürzungsvariablen von *smarty* besitzen. Nach unserer Definition haben diese Variablen die Form `{..$WERT..}`. Alle diese Dateien sind im Template-Verzeichnis abgespeichert, das wir als *typo3conf/ext/netcos\_renting/tmpl* definiert haben. Wir definieren in diesem Verzeichnis zwei Unterverzeichnisse *pi1* und *pi2*, die die Templates für die verschiedenen Plugins speichern.

### 5.1 Erstes Plugin (Vermietungsverwaltung)

Das Unterverzeichnis für das erste Plugin *tmpl/pi1* enthält 8 *smarty*-Template-Dateien.

*Administration.tmpl*:

```

<html><head>
  {..$IMPORT_JSCALENDAR..}
</head><body>
  <h1>{..$ADMINISTRATION_HEADER..}</h1><br /><br />
  <h2>{..$ADMINISTRATION_NEW..}</h2><br />
  <p>{..$ADMINISTRATION_DATE_FORMAT..}</p><br />
  <form method="post">
    <input type="text" id="data_from" name="data_from" />
    <button id="trigger_from">{..$FROM..}</button>
    <script type="text/javascript">
      Calendar.setup(
        {
          inputField : "data_from",      // ID of the input field
          ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
          button     : "trigger_from"    // ID of the button
        }
      );
    </script>

    <input type="text" id="data_to" name="data_to" />
    <button id="trigger_to">{..$TO..}</button>
    <script type="text/javascript">
      Calendar.setup(
        {
          inputField : "data_to",      // ID of the input field
          ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
          button     : "trigger_to"    // ID of the button
        }
      );
    </script>
    <input type="submit" name="apply" value="{..$APPLY..}" />
  </form>
  <br />
  <h2>{..$ADMINISTRATION_YEARS..}</h2><br />
  <form method="post">
    <span>{..$YEAR..}</span>
    <input type="text" size="4" maxlength="4" name="jahr"
      value="{..$AKTUELLES_JAHR..}" />
    <input type="submit" name="wechseln" value="{..$CHANGE2..}" />
    <input type="submit" name="aktuell" value="{..$ACTUAL..}" />
  </form>
  <br />
  <h2>{..$BOOKINGS_TITLE..}</h2><br />
  {..$AKTUELLE_BUCHUNGEN..}
</body></html>

```

*exists.tpl:*

```
<html><head></head><body>
  <h2>{..$EXISTS_HEADER..}</h2><br />
  <form method="post">
    <td><input type="submit" name="more" value="{..$MORE..}" /></td>
  </form>
</body></html>
```

*overlap.tpl:*

```

<html><head>
  {..$IMPORT_JSCALENDAR..}
</head><body>
  <h2>{..$OVERLAP_HEADER..}</h2><br />
  <table width="700px">
    <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
    <th></th><th></th></tr>
    <tr><form method="post">
      <td><input type="hidden" name="uid" value="{..$UID1..}" />{..$UID1..}</td>
      <td><input type="text" id="data_from1"
        name="data_from" value="{..$VALUE_FROM1..}" />
      <button id="trigger_from1">{..$FROM..}</button>
      <script type="text/javascript">
        Calendar.setup(
          {
            inputField : "data_from1",      // ID of the input field
            ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
            button     : "trigger_from1"    // ID of the button
          }
        );
      </script></td>
      <td><input type="text" id="data_to1" name="data_to" value="{..$VALUE_TO1..}" />
      <button id="trigger_to1">{..$TO..}</button>
      <script type="text/javascript">
        Calendar.setup(
          {
            inputField : "data_to1",      // ID of the input field
            ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
            button     : "trigger_to1"    // ID of the button
          }
        );
      </script></td>
      <td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
      <td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
      <td>{..$NEW..}</td>
    </form></tr><tr><form method="post">
      <td><input type="hidden" name="uid" value="{..$UID2..}" />{..$UID2..}</td>
      <td><input type="text" id="data_from2" name="data_from"
        value="{..$VALUE_FROM2..}" />
      <button id="trigger_from2">{..$FROM..}</button>
      <script type="text/javascript">
        Calendar.setup(
          {
            inputField : "data_from2",      // ID of the input field
            ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
            button     : "trigger_from2"    // ID of the button
          }
        );
      </script></td>

```

```

<td><input type="text" id="data_to2" name="data_to" value="{..$VALUE_TO2..}" />
<button id="trigger_to2">{..$TO..}</button>
<script type="text/javascript">
  Calendar.setup(
  {
    inputField : "data_to2",      // ID of the input field
    ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
    button     : "trigger_to2"   // ID of the button
  }
  );
</script></td>
<td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
<td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
<td>{..$EXISTING..}</td>
</form></tr></table>
</body></html>

```

*to\_be\_changed.tmpl:*

```

<html><head>
  {..$IMPORT_JSCALENDAR..}
</head><body>
<h2>{..$TO_BE_CHANGED_HEADER..}</h2><br />
<table width="700px">
  <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
  <th></th><th></th></tr>
  <form method="post">
    <tr><td><input type="hidden" name="uid" value="{..$UID..}" />{..$UID..}</td>
    <td><input type="text" id="data_from" name="data_from" value="{..$VALUE_FROM..}" />
    <button id="trigger_from">{..$FROM..}</button>
    <script type="text/javascript">
      Calendar.setup(
        {
          inputField : "data_from",      // ID of the input field
          ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
          button    : "trigger_from"    // ID of the button
        }
      );
    </script></td>
    <td><input type="text" id="data_to" name="data_to" value="{..$VALUE_TO..}" />
    <button id="trigger_to">{..$TO..}</button>
    <script type="text/javascript">
      Calendar.setup(
        {
          inputField : "data_to",      // ID of the input field
          ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
          button    : "trigger_to"    // ID of the button
        }
      );
    </script></td>
    <td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
    <td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
  </tr></table>
</body></html>

```

*to\_be\_deleted.tmpl:*

```
<html><head></head><body>
<h2>{..$TO_BE_DELETED_HEADER..}</h2><br />
<table width="700px">
  <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
  <th></th><th></th></tr>
  <tr align="center"><form method="post">
  <td><input type="hidden" name="uid" value="{..$UID..}">{..$UID..}</td>
  <td><input type="hidden" name="data_from"
  value="{..$VALUE_FROM..}" />{..$VALUE_FROM..}</td>
  <td><input type="hidden" name="data_to"
  value="{..$VALUE_TO..}" />{..$VALUE_TO..}</td>
  <td><input type="submit" name="delete" value="{..$DELETE..}" /></td>
  <td><input type="submit" name="more" value="{..$CANCEL..}" /></td>
  </form></tr></table>
</body></html>
```

*was\_created.tmpl:*

```
<html><head></head><body>
<h2>{..$WAS_CREATED_HEADER..}</h2><br />
<form method="post">
  <td><input type="submit" name="more" value="{..$MORE..}" /></td>
</form>
</body></html>
```

*was\_deleted.tmpl:*

```
<html><head></head><body>
<h2>{..$WAS_DELETED_HEADER..}</h2><br />
<form method="post">
  <td><input type="submit" name="more" value="{..$MORE..}" /></td>
</form>
</body></html>
```

*wrong\_entry.tmpl:*

```

<html><head>
  {..$IMPORT_JSCALENDAR..}
</head><body>
  <h2>{..$WRONG_ENTRY_HEADER..}</h2><br />
  <p>{..$ERRMSG..}</p><br />
  <table width="700px">
    <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
      <th></th><th></th></tr>
    <form method="post">
      <tr><td><input type="hidden" name="uid" value="{..$UID..}">{..$UID..}</td>
        <td><input type="text" id="data_from" name="data_from" value="{..$VALUE_FROM..}" />
          <button id="trigger_from">{..$FROM..}</button>
          <script type="text/javascript">
            Calendar.setup(
              {
                inputField : "data_from",      // ID of the input field
                ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
                button     : "trigger_from"    // ID of the button
              }
            );
          </script></td>

        <td><input type="text" id="data_to" name="data_to" value="{..$VALUE_TO..}" />
          <button id="trigger_to">{..$TO..}</button>
          <script type="text/javascript">
            Calendar.setup(
              {
                inputField : "data_to",      // ID of the input field
                ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
                button     : "trigger_to"    // ID of the button
              }
            );
          </script></td>
        <td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
        <td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
      </tr></table>
</body></html>

```

## 5.2 Zweites Plugin (gebucht)

Das Unterverzeichnis für das zweite Plugin *tmpl/pi2* enthält 2 *smarty*-Template-Dateien.

*booked.tmpl*:

```

<html><head>
</head><body>
<h1>{..$BOOKED_HEADER..}</h1>
<br />
<h2>{..$AKTUELL_TEXT..} {..$MONAT..} {..$JAHR..}</h2>
<br />
<table width="700px" style="border: 10px rgb(135,148,116) solid;">
  <tr align="center" style="background-color: #CCFFCC;">
    <th>{..$CALENDAR_WEEK..}</th>
    <th>{..$MONDAY..}</th><th>{..$TUESDAY..}</th><th>{..$WEDNESDAY..}</th>
    <th>{..$THURSDAY..}</th><th>{..$FRIDAY..}</th><th>{..$SATURDAY..}</th>
    <th>{..$SUNDAY..}</th></tr>
  {..$KALENDER_MONAT..}
</table><br />
<form method="post">
  <input type="submit" name="Monat" value="{..$PREVIOUS_MONTH..}" />
  <input type="submit" name="Monat" value="{..$ACTUAL_MONTH..}" />
  <input type="submit" name="Monat" value="{..$NEXT_MONTH..}" />
</form><br />
<form method="post">
<h3>{..$BOOKED_FUTURE..}</h3>
  {..$MONATSRADIO..}
  <br />
  <span>{..$YEAR..}</span>
  <input name="Jahr" size="4" maxlength="4" type="text"
  value="{..$AKTUELLES_JAHR..}" />
  <br />
  <input type="submit" name="wechseln" value="{..$BOOKED_CHANGE..}" />
</form>
</body></html>

```

*error.tmpl:*

```

<html><head>
</head><body>
<h2>{..$ERROR_HEADER..} {..$ZEIT..}</h2>
<br />
  {..$ERROR_TEXT..}
<br />
<br />
<form method="post">
  <input type="hidden" name="Monat" value="{..$AKTUELLER_MONAT..}" />
  <input type="hidden" name="Jahr" value="{..$AKTUELLES_JAHR..}" />
  <input type="submit" name="wechseln" value="{..$ERROR_BACK..}" />
</form>
</body></html>

```