

2011

www.time4mambo.de

Axel Tüting

Version 1.0

TUTORIAL: EIGENE TEMPLATES IN JOOMLA!1.6

In diesem Tutorial wird erklärt, wie man ein einfaches Template erstellen kann. Es wird ein sehr einfaches drei Spalten-Template erzeugt, die Install-Datei wird erklärt und die meisten CSS-Klassen, die Joomla benutzt genannt. Es geht nicht darum, ein kompliziertes Template mit Parametern und vielerlei mehr zu erstellen, sondern das grundsätzliche Wissen zu vermitteln, mit dem man eigene Templates erstellen kann.

Inhalt

Vorwort	3
Verzeichnisstruktur	3
index.php.....	3
Übung mit einfachen Div-Design.....	5
Positionen.....	6
Besondere Positionen / Platzhalter.....	7
Einfügen der Positionen ins Template	7
templateDetails.xml	8
Template.css.....	10

Vorwort

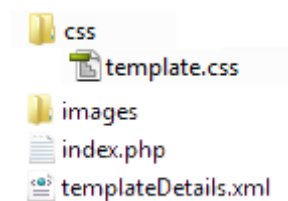
Die hier erzeugten Dateien stehen im Downloadcenter auf meiner Website zur Verfügung. Einfach dem Link auf der Startseite oder am Seitenende folgen.

Dieses Tutorial darf frei und kostenlos verbreitet werden, sofern inhaltlich nichts geändert wird.

Das Urheberrecht bleibt beim Autor.

Verzeichnisstruktur

Um ein eigenes Template zu erstellen, wird folgende Verzeichnisstruktur benötigt:



In der "index.php" steht die Grundstruktur des Templates. Also der eigentliche HTML-Code. Die "templateDetails.xml" wird zum Installieren benötigt und gibt auch u.a. die möglichen Positionen wider.

Im Verzeichnis "css" steht die CSS-Datei und im images-Verzeichnis evt. Grafiken, die für das Template benötigt werden.

index.php

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
?>
```

Dieser Teil verhindert, dass die index.php einfach so aufgerufen wird. Beispielsweise um schädlichen Code einzuschleusen. Wer keine Rechte zum editieren hat, kann die index.php nicht verändern. Dabei geht es nicht um einen Schutz gegen unberechtigten Einblick in den Quellcode. Das kann hiermit nicht verhindert werden.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang=
"<?php echo $this->language; ?>" lang="<?php echo $this->language; ?>"
dir="<?php echo $this->direction; ?>" > >
```

Die eingestellte Sprache wird automatisch eingeladen und verarbeitet. Auch ein wichtiger Eintrag für mehrsprachige Websites. Die ausgewählte Sprache wird per PHP-Code automatisch eingeladen und angezeigt. Beide Angaben sind beispielsweise für den Browser wichtig.

```
<head>
<jdoc:include type="head" />
```

Die Meta-Tags, die in den Artikeln angegeben werden, Erklärungen (Description), Autor und alles weitere, was wichtig in den Metatags sind, werden von Joomla entsprechend Ihrer Einstellungen zusammengestellt und automatisch zur Laufzeit hier ausgegeben

```
<link rel="stylesheet" href="<?php echo $this->baseurl;
?>/templates/system/css/system.css" type="text/css" /> <link
rel="stylesheet" href="<?php echo $this->baseurl;
?>/templates/system/css/general.css" type="text/css" />
```

Zwei CSS-Dateien, die von Joomla für generelle Ausgaben benutzt werden.

```
<link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php
echo $this->template ?>/css/template.css" type="text/css" />
```

In "\$this->baseurl" steht der Domainname und gfls. das Joomla-Verzeichnis, wenn es nicht im Root untergebracht ist. In "\$this->template" der Name des Template-Verzeichnisses, welches benutzt wird. Wird die URL auf diese Art eingeben, braucht man sich keine Gedanken mehr zu machen, wo denn nun die richtigen Dateien liegen. Joomla stellt automatisch das richtige Verzeichnis zusammen.

```
<link rel="shortcut icon" href="/images/favicon.ico" />
```

Ein eigenes Favicon kann geladen werden. Favicons sind die kleinen Icons, die in der Adresszeile des Browsers angezeigt werden. Meist auch im einzelnen TAB eines Browsers.

```
</head>
```

Das Ende des Kopfes der index.php

```
<body>
...
</body>
</html>
```

Zwischen den "body-Tags" kommt das grundsätzliche Aussehen des Templates und in der CSS-Datei wird das Template formatiert.

Übung mit einfachen Div-Design

Zur Übung wollen wir ein einfaches DIV-Template erstellen:



Der Code sieht folgendermaßen aus:

```
<div id="oben_beispiel">oben</div>
<div>
<div id="links_beispiel">links</div>
<div id="rechts_beispiel">rechts</div>
<div id="mitte_beispiel">mitte</div>
</div>
<div id="unten_beispiel">unten</div>
```

Und in die CSS-Datei (template.css) müssen diese Angaben rein:

```
#oben_beispiel {
border-width:thin;border-style:solid;border-color:#000000;
background-color:#FFFF00;
}

#links_beispiel {
width:200px;float:left;border-width:thin;border-style:solid;border-
color:#000000;
background-color:#80FF80;
}

#rechts_beispiel {
width:200px;float:right;border-width:thin;border-style:solid;border-
color:#000000;
background-color:#00FF80;
}

#mitte_beispiel {
margin: 0 15.5em;border-width:thin;border-style:solid;border-color:#000000;
background-color:#80FFFF;
}

#unten_beispiel {
border-width:thin;border-style:solid;border-color:#000000;
background-color:#FF8040;
}
```

Positionen

Um nun Inhalte in die einzelnen Bereiche zu bekommen, müssen sogenannte "Positionsangaben" gemacht werden. Positionsangaben sind nichts weiter als Platzhalter, an Stelle dieser später Inhalte ausgegeben werden.

Der Hautinhalt, der Content (Artikel, Komponenten) soll im Bereich "mitte" ausgegeben werden. "links" soll das Menü hin, "oben" ein Header, "rechts" ein paar Module, wie beispielsweise das aktuelle Wetter und "unten" soll das Copyright stehen.

Die Inhalte erzeugen wir später im Backend (Adminbereich) von Joomla. In dem wir Artikel schreiben und Menüs anlegen. Das Copyright und den Header erstellen wir, in dem wir eigene Module definieren und das Wetter ist ein externes Modul, welches wir im Joomla installieren und dann an der Position "rechts" an Stelle des Platzhalters ausgeben.

Ein Platzhalter hat immer das gleiche aussehen:

```
<jdoc:include type="modules" name="left" style="none" />
```

In **type** steht, um was es sich handelt. Bis auf wenige Ausnahmen (die wir gleich noch betrachten werden), sind das meistens "modules" - Module, die vom Platzhalter eingelesen werden sollen.

In **name** steht die eigentliche Position. Diese Position wird beim Modul ausgewählt und damit festgelegt, wo das Modul angezeigt wird. Werden mehrere Module für eine Position ausgewählt, werden die einfach untereinander angezeigt.

style gibt an, wie das Modul verarbeitet werden soll. Dazu stehen folgende mögliche Stile standardmäßig zur Verfügung:

- none
kein spezieller Stil soll ausgegeben werden. Diese Angabe kann auch weggelassen werden. Also einfach nur:

```
<jdoc:include type="modules" name="left">
```
- xhtml
Der Inhalt wird in DIV-Tags ausgegeben
- table
Der Inhalt wird in TABLE-Tags ausgegeben (td)
- horz
Der Inhalt wird horizontal ausgegeben
- outline
Der Inhalt wird hervorgehoben (mit einer Hintergrundfarbe belegt)
- rounded
Der Inhalt wird mit einem Rahmen und abgerundeten Ecken versehen. Voraussetzung ist allerdings, dass selbige auch in der CSS-Datei definiert sind. Sonst wird diese Angabe einfach ignoriert.

Besondere Positionen / Platzhalter

Ein paar Positionsangaben unterscheiden sich von den anderen:

```
<jdoc:include type="component" />
<jdoc:include type="message" />
```

component kennzeichnet den Hauptinhalt. Also Artikel und Komponenten, die dann an der Stelle ausgegeben werden. In früheren Joomla- und Mamboversionen wurde diese Positionangabe mit "Main" gekennzeichnet.

message ist verantwortlich für etwaige Fehlerausgaben oder wenn man eine eMail in den Kontakten abschickt - dann wird an der Stelle der erfolgreiche Versand gemeldet.

Weit verbreitet sind diese Angaben, die aber nicht zwingend so vorgegeben sind. Man kann also seine Banner auch an Positionen mit anderen Namen ausgeben:

```
1.<jdoc:include type="modules" name="breadcrumbs" style="none" />
2.<jdoc:include type="modules" name="banner" style="none" />
```

banner läßt schon vermuten, dass man damit das Bannermodul, bzw. deren Inhalte ausgeben kann.

Die **breadcrumbs** ist die jetzige Bezeichnung für den "Pathway". Also die Pfadangabe, wo man sich gerade befindet im Webauftritt.

Einfügen der Positionen ins Template

Die Platzhalter / Positionsangaben werden einfach zwischen den DIV-Tags geschrieben:

```
<div id="oben_beispiel">
<jdoc:include type="modules" name="top" style="none" />
</div>
<div>
<div id="links_beispiel">
<jdoc:include type="modules" name="left" style="none" />
</div>
<div id="rechts_beispiel">
<jdoc:include type="modules" name="right" style="none" />
</div>
<div id="mitte_beispiel">
<jdoc:include type="component" />
</div>
</div>
<div id="unten_beispiel">
<jdoc:include type="modules" name="bottom" style="none" />
```

```
</div>
```

Das war's schon. Jetzt noch die Datei als `index.php` im Templateordner speichern.

templateDetails.xml

Die **templateDetails.xml** Datei dient zum einen der Installation, enthält Informationen über das Template und auch die möglichen Positionen des Templates.

Wenn man die Datei speichert, muss man auf die richtige Schreibweise - mit einem großen "D" in der Mitte - achten!

Betrachten wir die Datei im Einzelnen:

```
<?xml version="1.0" encoding="utf-8"?>
```

"utf-8" kann auch gegen einen anderen Zeichensatz ausgetauscht werden. Wobei **utf-8** den europäischen Zeichensatz darstellen kann.

```
<!DOCTYPE install PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN"  
"http://www.joomla.org/xml/dtd/1.6/template-install.dtd">
```

Das DOCTYPE gibt an, um welche Sprache es sich handelt. Also das es ein XML-File ist und wie es im detail zu interpretieren ist.

```
<extension version="1.6" type="template" client="site">  
<name>TemplateName</name>
```

Die Version bezieht sich auf Joomla und der Typ kennzeichnet, das es ein Template ist. *client* gibt an, für welchen Bereich das Template gemacht ist. `client="site"` gibt an, dass es ein Frontend-Template ist. Mit `client="admin"` wird ein Template für das Backend gekennzeichnet.

Für "TemplateName" tragen Sie den Namen Ihres Templates ein. Unter diesen Namen ist das Template später auch auswählbar.

```
<creationDate>05/2009</creationDate>  
<author>Axel Tueting</author>  
<copyright>time4mambo - Axel Tueting</copyright>  
<authorEmail>tueting@time4mambo.de</authorEmail>  
<authorUrl>http://www.time4mambo.de</authorUrl>
```

Ich denke, die Angaben sprechen für sich und bedürfen keiner weiteren Erklärung. Tun Sie mir aber bitte den Gefallen und kopieren Sie nicht meinen Namen mit, wenn Sie Ihre eigenen Templates erstellen 😊

Mit den Umlauten gibt es manchmal Probleme. Weshalb es Sinn macht, statt der Umlaute **ue**, **ae**, etc. zu schreiben

```
<version>0.1</version>
```

Versionskontrolle kann nie schaden. Manchmal verändert man am Template etwas - hiermit hat man den Überblick.

```
<description>This template ist free under the GNU/GPL-lizense</description>
```

Eine kurze Erklärung - zum Copyright oder wofür das Template ist oder ...

```
<files>
<filename>index.php</filename>
<filename>templateDetails.xml</filename>
</files>
```

Schreiben Sie hier die Files hinein, die für das Template benötigt werden und geben Sie den relativen Pfad an, wo die Files hingeschrieben werden sollen, wenn das Template installiert wird. Dazu gehört natürlich auch diese Datei, die Sie gerade schreiben.

```
<images>
<filename>images/bild.jpg</filename>
</images>
```

Sie brauchen nicht extra einen "images"-Block, auch nicht einen "css-Block", wie im Anschluss. Sie können diese Files auch direkt in den "files-Block" schreiben. So ist es aber etwas übersichtlicher. Besonders wenn Sie viele Grafiken oder auch mehrere CSS-Dateien haben. Denken Sie auch wieder an den korrekten relativen Pfad!

```
<css>
<filename>css/template.css</filename>
</css>
```

Wie gesagt: Sie müssen nicht extra einen css-Block anlegen, schaden kann es aber auch nicht...

```
<positions>
<position>left</position>
<position>right</position>
<position>top</position>
<position>bottom</position>
</positions>
```

Hier finden Sie die Positionen wieder, die wir in die "index.php" eingetragen haben. Wichtig zu wissen ist, dass hier die Positionen stehen, die Sie später auch in dem Dropdown-Button im Modul wieder finden. Anders ausgedrückt: vergessen Sie hier eine Position einzutragen, kann diese nachher auch nicht mit dem Dropdown-Button ausgewählt werden (wohl aber per Hand direkt in das Feld geschrieben werden).

Viele Templates, die Sie im Internet finden, haben hier mehr Positionen stehen, als tatsächlich im Template enthalten sind. Lassen Sie sich also nicht davon täuschen, dass Sie so viele Positionen auswählen können - schauen Sie besser nach, was tatsächlich da ist.

```
</extension>
```

Am Ende die XML-Datei schließen und speichern.

Template.css

Anbei die meisten CSS-Klassen und –Ids, die Joomla benutzt. Ich erhebe keinen Anspruch auf Vollständigkeit. Ich denke aber, die allermeisten Klassen sind damit abgedeckt.

```
/**
 * -----
 * Eine kommentierte CSS-Datei für Joomla!1.6
 * Es handelt sich dabei um alle möglichen Klassen oder IDs,
 * die innerhalb von Joomla vorkommen koennen.
 * -----
 * Zusammengestellt von:
 * Axel Tueting
 * http://www.time4mambo.de
 * tueting@time4mambo.de
 * -----
 * Juni 2011
 *
 *
 */

/**
 *
 * Beitrag
 *
 */

.item-page { }

/* Mit eigener 'Seitenklasse' */
/* Mit Bindestrich */

.item-page[-meineSeitenklasse] { }
```

```
/* Als Klasse (also mit einem Leerzeichen am Beginn der 'Seitenklasse') */
```

```
.item-page.[meineSeitenklasse] { }
```

```
/* Beitragsueberschrift */
```

```
h2 { }
```

```
/* Leiste in der rechts das Druck-Icon, etc. steht */
```

```
ul.actions { }
```

```
li.print-icon { }
```

```
li.email-icon { }
```

```
/* Beitragsbewertung */
```

```
.content_rating { }
```

```
content_vote { }
```

```
.content_vote input { }
```

```
.content_vote input.button { }
```

```
/* Beitragsinfo */
```

```
dl.article-info { }
```

```
dd.category-name { }
```

```
dd.create { }
```

```
dd.modified { }
```

```
dd.published { }
```

```
dd.createdby { }
```

```
dd.hits { }
```

```
/* Beitragsblog */
```

```
.blog {}  
  
.blog h2 {}  
  
.subheading-category {}  
  
.category-desc {}  
  
.items-leading {}  
  
.leading-0 {}  
  
.item column-1 {}  
  
.items-row {}  
  
.cols-2 {}  
  
.row_0 {}  
  
.row-seperator {}  
  
  
/**  
*  
* Beitragsliste  
*  
*/  
  
.category-list {}  
  
.display-limit {}  
  
select#limit {}  
  
.cat-items {}  
  
form#adminForm {}  
  
table.category {}  
  
th#tableOrdering {}  
  
th#tableOrdering2 {}  
  
th#tableOrdering3 {}  
  
th#tableOrdering4 {}
```

```
td.list-title {}

tr.cat-list-row0 {}

tr.cat-list-row1 {}

td.createdby {}

td.list-hits {}

.cat-children {}

.cat-children h3 {}

.cat-items {}

li.first {}

li.first span.item-title {}

li.first span.item-title a {}

li.last {}

li.last span.item-title {}

li.last span.item-title a {}

/**
 *
 * Kategorienliste
 *
 */

.categories-list {}

li.first {}

.categories-list li.first {}

.categories-list li.last {}

.categories-list span.item-title {}

.categories-list dl.article-count {}
```

```
/* Text: Beitragsanzahl */
```

```
dl.article-count dt {}
```

```
/* Anzahl der Beitrage */
```

```
dl.article-count dd {}
```

```
/**
```

```
*
```

```
*Menue
```

```
*
```

```
*/
```

```
ul.menu {}
```

```
/* Alle Menueverweise */
```

```
ul.menu li a:link {}
```

```
ul.menu li a:hover {}
```

```
ul.menu li a:visited {}
```

```
ul.menu li.active {}
```

```
ul.menu li.current {}
```

```
ul.menu li.current.active {}
```

```
/*
* Einzelner Menueverweise
* -----
* in [MenuID] muss die MenuID vom Backend rein.
* Also in der Menueuebersicht ganz rechts die ID auslesen und dann hier eingeben.
* So koennen einzelne Menuepunkte direkt angesprochen werden.
* -----
* Beispiel:
* li#item-101 {...}
*
*/

li#item-[MenuID] {}

li#item-101.active {}

li#item-101.current {}

li#item-101.current.active {}

/* Wenn ein Icon im Menueverweis angezeigt wird */

li#item-101 a img {}

li#item-101 a span.image-title {}

/* Hauptverweis bei Untermenues */

li#item-101.parent {}

/* Individuelles Menue-Modul mit dem 'Menu Tag ID' vom Backend */

ul#[meineMenuTagID] {}

ul#[meineMenuTagID].menu {}

/* Eigener 'Link CSS Style' im Backend gesetzt */

a.[MeinLinkCSSStyle] {}
```

```
/**
 *
 * pathway - breadcrumbs
 *
 */

.breadcrumbs {}

.breadcrumbs img {}

a.pathway {}
```

```
/**
 *
 * Module
 *
 */
```

```
.moduletable {}

moduletable h3 {}
```

```
/* Individuelles Modul mit der 'Modulklassensuffix' */
```

```
/* Mit Bindestrich: */
```

```
.moduletable-[MeineSuffix] {}
```

```
/* Als Klasse (also mit einem Leerzeichen zu Beginn in der Modulklassensuffix) */
```

```
.moduletable.[MeineSuffix] {}
```