

Technisches Handbuch phpComasy

Version

26. Februar 2013

Autor

Emanuel Zuber (emanuel.zuber@indual.ch)

1 Inhaltsverzeichnis

1 INHALTSVERZEICHNIS	2
1 TEMPLATES	3
1.1 Grundaufbau	3
1.1.1 Grund-Struktur für Templates.....	3
1.1.2 Optionale Struktur für Templates	3
1.2 Template Tags.....	4
1.2.1 Reguläre Template Tags.....	4
1.2.2 Modulspezifische Template Tags.....	7
1.3 Beispiel Template mit HTML5 Doctype.....	8
2 SEITENVORLAGEN	8
3 MODULE	9
3.1 Platzierung von Modulen	9
3.2 Grundaufbau von Modulen	9
3.3 Installation und Verwendung von Modulen.....	9
4 WEITERFÜHRENDE LINKS	10
5 GLOSSAR	10
6 ÜBER DIESES DOKUMENT.....	10

1 Templates

1.1 Grundaufbau

Der Aufbau eines Templates muss gewissen Vorgaben entsprechen. Es muss eine klar definierte Struktur eingehalten werden.

- Im Allgemeinen sollten keine Sonderzeichen, Umlaute oder Leerzeichen in Dateinamen jeder Art verwendet werden.
- Es können beliebig viele Templates-Ordner erstellt werden. In einem Template-Ordner befinden sich der Aufbau der Webseite (HTML/CSS/JS), sowie auch spezielle Konfigurationen von Modulen wie auch komplette Module die nur für eine Vorlage entwickelt wurde.
- Alle Templates müssen im Hauptordner «templates» gespeichert werden.

1.1.1 Grund-Struktur für Templates

./index.htm	Vorlage in HTML
./img/{Bild-Dateien}	Beliebig viele Bild-Dateien unter dem Ordner img/
./css/{CSS-Dateien.css}	Beliebig viele CSS-Dateien unter dem Ordner css/

1.1.2 Optionale Struktur für Templates

./mobile.htm	Vorlage für mobile optimierte Ansicht
./email.htm	E-Mail HTML-Vorlage
./script.php	PHP-Skript für individuelle PHP Skripte
./js/{Javascript-Dateien.js}	Beliebig viele Javascript-Dateien unter dem Ordner js/
./language/{Sprachfile.php}	Definition von Sprachabhängigen Inhalten als Konstanten
./modules/	Unterordner mit Modulnamen, Konfigurationen oder ganze Module
./page_templates/{Seitenvorlagen}	Beliebig viele Seitenvorlagen, siehe Abschnitt «Seitenvorlagen»

1.2 Template Tags

Template Tags sind immer gross geschrieben und besitzen jeweils «__» am Anfang und am Ende. Bei einigen Tags ist es möglich, Parameter zu übergeben. Parameter werden in eckigen Klammern angegeben: «[]» und müssen auch so in den Templates gesetzt werden.

Optionale Parameter werden in geschweiften Klammern angegeben: «{ }». Diese können gesetzt werden, müssen aber nicht. Geschweifte Klammern dienen nur in diesem Dokument als Hinweise, diese werden nie so in den Template Tags verwendet.

1.2.1 Reguläre Template Tags

Template Tag	Beschreibung
__SITE_TITLE__	Name der aktuellen Seite.
__SITE_NAME__	Name der Webseite (gemäss Einstellungen).
__SITE_DESCRIPTION__	Beschreibung der Webseite (gemäss Einstellungen).
__STANDARD_HEAD__	Erstellt einen Standard <head></head>-Abschnitt mit allen entsprechenden Informationen: title-Tag, Allgemeine Header-Informationen, META-Informationen und CSS-Definitionen.
__HEAD_CONTENT__	Erstellt einen Abschnitt mit dem title-Tag, Allgemeinen Header-Informationen, META-Informationen und CSS-Definition. Kann innerhalb des <head>-Abschnittes platziert werden. Dieser Tag ist zwingend, falls nicht __STANDARD_HEAD__ verwendet wird.
__META_DATA__	Erstellt einen Abschnitt mit alle META-Informationen. Kann innerhalb des <head>-Abschnittes platziert werden.
__STYLES__	Erstellt einen Abschnitt mit allen CSS-Definitionen. Kann innerhalb des <head>-Abschnittes platziert werden. Dieser Tag ist zwingend, falls nicht __STANDARD_HEAD__ oder __HEAD_CONTENT__ verwendet wird.
__JAVASCRIPTS__	Erstellt einen Abschnitt mit alle Javascript-Deklarationen. Kann innerhalb des <head>-Abschnittes platziert werden. Dieser Tag ist zwingend.
__BODY_HANDLER__	Wird vom System im <body> Tag verwendet und muss wie folgt integriert werden Dieser Tag ist zwingend.

__ADMINISTRATION_PANEL__	<p>Dies ist das Administrationspanel von phpComasy. Wird üblicherweise nach dem </head>-Tag verwendet.</p> <p>Dieser Tag ist zwingend.</p>
__PHPSCRIPT__	<p>Mit diesem Tag wird die Datei script.php im Template-Ordner eingelesen und ausgeführt.</p>
__PHPSCRIPT[parameter]__	<p>Mit diesem Tag wird die Datei script.php im Template-Ordner eingelesen und ausgeführt. Es ist möglich einen freien Parameter zu übermitteln der in der Datei script.php als \$param zur Verfügung steht.</p>
__TITLE__	<p>Titel der aktuellen Seite als <h1></h1>.</p>
__TITLE_WITHOUT_HTML__	<p>Titel der aktuellen Seite ohne HTML-Tags.</p>
__PAGE__	<p>Hier wird der Inhalt einer Seitenvorlage angezeigt. Ist keine Seitenvorlage angegeben wird automatisch die Seitenvorlage «system.tpl» geladen.</p>
__CONTENT__	<p>Hier wird der Inhalt der Seite angezeigt. Falls ein Einleitungstext existiert, wird dieser im Element <div class="introduction"> angezeigt.</p>
__INTRODUCTION__	<p>Es wird der Einleitungstext angezeigt.</p>
__MESSAGE__	<p>Status-Meldungen wie erfolgreicher Login, falsch ausgefülltes Kontaktformular etc.</p> <p>Dieser Tag ist zwingend.</p>
__LANGUAGE_SWITCHER__	<p>Zeigt eine HTML-Struktur mit einem Sprachwechsler.</p>
__LOGIN_PANEL__	<p>Zeigt eine HTML-Struktur für eine Login-Box an.</p>
__SEARCH_BOX__	<p>Zeigt eine HTML-Struktur für ein Suchfeld an.</p>
__BREADCRUMB__	<p>Zeigt eine HTML-Struktur mit aktuellen Navigationspfad an.</p>
__NAVIGATION__	<p>Zeigt die Navigation als ul-li-HTML Struktur an.</p>
__NAVIGATION_DESC__	<p>Zeigt die Navigation umgekehrt sortiert als ul-li-HTML Struktur an.</p>
__ROOT_NAVIGATION__	<p>Zeigt nur die Hauptnavigation als ul-li-HTML Struktur an.</p>
__NAVIGATION[level]__	<p>Zeigt die Navigation ab einen bestimmten Level. Der Wert</p>

	«level» muss numerisch sein, beispielsweise ab Level «1» wäre die 1. Unternavigation.
__SUB_NAVIGATION[category_id]__	Zeigt die Navigationsstruktur ab einer bestimmte Kategorie. Die Kategorie_id «category_id» muss numerisch sein.
__META_NAVIGATION[category_id]__	Zeigt die Navigationsstruktur ab einer bestimmte Kategorie und blendet diese Kategorie in anderen Navigations-Ansichten aus. Die Kategorie_id «category_id» muss numerisch sein.
__PICTURE[...]__	Zeigt die Bilder aus dem Ressourcen-Pool der aktuellen Seite an. Die Nutzung dieses Tags ist in der Datei «page_templates/standard.tpl» beschrieben.
__SLIDER[...]__	Zeigt die Bilder aus dem Ressourcen-Pool als Slider der aktuellen Seite an. Die Nutzung dieses Tags ist in der Datei «page_templates/standard.tpl» beschrieben.
__MUSIC__	Zeigt die Musik-Dateien aus dem Ressourcen-Pool der aktuellen Seite an. Die Nutzung dieses Tags ist in der Datei «page_templates/standard.tpl» beschrieben.
__VIDEOS__	Zeigt die Video-Dateien aus dem Ressourcen-Pool der aktuellen Seite an. Die Nutzung dieses Tags ist in der Datei «page_templates/standard.tpl» beschrieben.
__DOCUMENTS__	Zeigt die Dokumente aus dem Ressourcen-Pool der aktuellen Seite an. Die Nutzung dieses Tags ist in der Datei «page_templates/standard.tpl» beschrieben.
__NO_MOBILE_LINK__	Wird bei einem mobilen Template verwendet, wenn man auf die klassische Desktop Ansicht wechseln möchte. Gibt ein <a>-HTML Element aus.
__FONTSIZE_SWITCHER__	Mit diesem Tag ist es möglich eine Wechsler für die Schriftgröße zu integrieren. Wird benötigt um barrierefreie Webseiten zu erstellen.

1.2.2 Modulspezifische Template Tags

__MODULE[modulname{,free_parameter}]__	Integration eines Modules. Der Modulname «modulname» ist zwingend erforderlich. Einige Module unterstützen einen freien Parameter «free_parameter».
__PORTLET_ZONE[zonenname {,drag_direction,width,height}]__	<p>Integration einer Portlet Zone. Es muss ein Name «zonenname» für die Portlet Zone definiert werden. Optional können weitere Parameter definiert werden.</p> <p>«drag_direction» gibt an ob die Module innerhalb der Zone Horizontal oder Vertikal verschoben werden können.</p> <p>Die Angaben «width» und «height» beziehen sich auf Bilder, die Masse können so in Pixel vordefiniert werden.</p> <p>In der Portlet Zone ist es möglich mehrere Template-Module zu platzieren und diese per Drag'n'drop zu verschieben.</p>

1.3 Beispiel Template mit HTML5 Doctype

Vereinfachtes Beispiel wie Template Tags in die HTML Struktur integriert werden können.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"><title>__SITE_TITLE__</title>
  __META_DATA__
  <link rel="shortcut icon" type="image/x-icon" href="favicon.ico" />
  <link rel="icon" type="image/x-icon" href="favicon.ico" />
  __HEAD_CONTENT__
  __STYLES__
  __JAVASCRIPTS__
</head>
<body id="phpcomasy" __BODY_HANDLER__ >
  __ADMINISTRATION_PANEL__

  <div id="siteframe">
    <div id="rootnavigation">
      __ROOT_NAVIGATION__
    </div>
    <div id="contentframe">
      <div id="sidebar">
        __NAVIGATION[1]__
        __MEMBER_PANEL__
        __PORTLET_ZONE[left]__
      </div>
      <div id="content">
        __TITLE__
        __PAGE__
      </div>
    </div>
  </div>
</body>
</html>
```

2 Seitenvorlagen

Seitenvorlagen helfen dem Kunden alle Seiten nach gewissen Richtlinien zu erstellen. So werden einheitliche Ansichten erstellt und es ist genau definiert welcher Text wo erscheint, welches Bild wo mit welchen Massen angezeigt wird, etc.

Die Standard-Seitenvorlagen befinden sich im Haupt-Ordner «page_templates». Über eine Einstellung kann gesteuert werden ob diese Standard-Seitenvorlagen angezeigt werden sollen oder nicht.

Es besteht die Möglichkeit, individuelle Seitenvorlagen anzulegen. Für diesen Zweck muss innerhalb des Template-Ordners ein Ordner «page_templates» angelegt werden. Eine Seitenvorlage existiert immer aus zwei Dateien: Einer Definition (vorlagenname.tpl) sowie einem Icon (vorlagenname.gif). Das Icon muss dabei die Masse 75x50px einhalten.

In den Seitenvorlagen kann genau gleich wie in der index.html mit HTML gearbeitet werden. Alle Template Tags sind auch hier erlaubt.

3 Module

3.1 Platzierung von Modulen

Mit Modulen lässt sich der Standardumfang von phpComasy erweitern. Die mitgelieferten Module von phpComasy befinden sich in dem Haupt-Ordner «modules».

Es gibt zwei verschiedene Modultypen: Inhaltsmodule, welche im Inhalt wie reguläre Seiten angezeigt werden und Portlets die innerhalb von Portlet Zonen oder fix als Module ins Template integriert werden können.

Es besteht die Möglichkeit, individuelle Module anzulegen. Für diesen Zweck muss innerhalb des Template-Ordners ein Ordner «modules » angelegt werden und das entsprechende Modul dort erstellt werden.

Es gibt folgende Möglichkeiten den «modules» Ordner zu nutzen:

- Es wird ein vorhandenes Modul genutzt, aber die Standard-Eigenschaften dieses Modules will man anders haben.
In diesem Fall erstellt man einen Ordner mit dem Modulnamen und kopiert darin die config.php Datei des Modules. Nun kann diese frei angepasst werden ohne dass das Originalmodul überschrieben wird.
z.B. modules/listing/config.php
- Es wird ein vorhandenes Modul umgeschrieben.
In diesem Fall verschiebt man das Modul von Haupt-Ordner «modules» in den Template-Ordner «modules» und passt das Modul nach freien Wünschen an.
- Es wird ein komplett neues Modul entwickelt.
Ist das Modul etwas generelles, was auch bei anderen Kunden verwendet werden kann, so macht es Sinn dieses im Haupt-Ordner «modules» anzulegen. Ist das Modul eine kundenspezifische Erweiterung die nur für ein spezifisches Template gilt so sollte das Modul im Template-Ordner «modules» angelegt werden.

3.2 Grundaufbau von Modulen

Siehe Beispielm modul «hello_world».

3.3 Installation und Verwendung von Modulen

Siehe phpComasy Handbuch.

4 Weiterführende Links

indual GmbH	www.indual.ch
WYSIWYG Editor	www.ckeditor.net
Mozilla Firefox	www.mozilla.com
FileZilla	www.filezilla.de

5 Glossar

WYSIWYG	Steht für «What you see is what you get» – Ein Editor, welcher direkt den eingegebenen Text so formatiert, wie er zum Schluss auf der Seite angezeigt wird.
Mozilla Firefox	Mozilla Firefox ist ein moderner, kostenloser Webbrowser.
FTP	Protokoll zum Übertragen von Dateien.
FileZilla	Eine kostenlose und gute Software zur Nutzung von FTP.

6 Über dieses Dokument

Dieses Dokument gehört der Firma indual GmbH. Diese Dokumentation oder Teile davon dürfen nur mit Einverständnis der indual GmbH veröffentlicht werden.