ULICMS

Neue Features in UliCMS 9.8.1

Stand: 19.12.2015

Inhaltsverzeichnis

Über dieses Dokument....3
  Zielgruppe....3
Übersicht der neuen Features....3
Detailbeschreibungen....4
  Neues Standard-Design....4
  Option "Design für Mobilgeräte nicht auf Tablets verwenden" hinzugefügt....5
  CKEditor aktualisiert....5
  Parameter $order zu den Funktionen menu() und get_menu() hinzugefügt....5
  Automatische Überprüfung auf Aktualisierungen für Pakete....6
    Prüfung deaktivieren....6
Neue Funktionen und Klassen....7
  API-Funktion initconfig($key, $value) hinzugefügt....7
  API-Funktion get_files($root_dir) hinzugefügt....8
  Objekte....10
    Settings-Klasse hinzugefügt....10
    Template-Klasse hinzugefügt....11
  Neue Filter....12
Willkommensbotschaft für das Loginformular....13
Sonstige Änderungen....14
  Änderungen der Datenbankstruktur....14
  Änderungen der API....14

Über dieses Dokument

Dieses Dokument beschreibt die neuen Features in UliCMS 9.8.1.

Zielgruppe

Zielgruppe dieses Dokuments sind Systemadministratoren, Webentwickler und Endanwender von UliCMS.

Übersicht der neuen Features

Diese Version von UliCMS enthält alle Features die in UliCMS 9.8.0 enthalten waren, sowie die folgenden neuen Features.

- Neues Standard-Design „2016“
- Option "Design für Mobilgeräte nicht auf Tablets verwenden" hinzugefügt
- CKEditor auf Version 4.5.5 aktualisiert
- Parameter $order zu den Funktionen menu() und get_menu() hinzugefügt
- Die Paketverwaltung überprüft nun die Paketquelle auf Aktualisierungen
- API-Funktion get_files() hinzugefügt
- API-Funktion initconfig() hinzugefügt
- Settings-Klasse hinzugefügt
- Template-Klasse hinzugefügt
- unique_identifier_filter hinzugefügt
- Es kann nun eine Willkommensbotschaft für das Loginformular festgelegt werden

Detailbeschreibungen

Im Folgenden eine detailierte Beschreibung der neuen Features.

Neues Standard-Design

UliCMS 9.8.1 kommt mit dem neuen Standard-Design „2016“. Das alte Standard-Design „cleanblue“ wird aber weiterhin mit installiert.

Das Design „2016“ hat die folgenden Eigenschaften:

Menüs	Oben und Unten Das obere Menü ist ein Dropdown Menü mit einer Unterebene. Das unter Menü hat keine Unterebenen
Einstellungen	Alle Designeinstellungen von UliCMS werden unterstützt, darunter Farben, Schriftart und Schriftgröße
Mobiltauglich	„2016“ ist Mobiltauglich, es nutzt dafür HTML 5, CSS3 und SlickNav. Wenn die Anzeigefläche gleich oder kleiner als 830 ist, wechselt das Design in die mobile Ansicht.
Grafik	Die Header Grafik steht unter Public Domain. Sie kann jedoch leicht durch eine eigene Datei ausgetauscht werden. Einfach die header.jpg ersetzen.

Das Design „2016“ eignet sich wegen seiner Einfachheit als Basis für eigene Designs.

Das Design ist unter dem Namen theme-2016 als Paket in der Paketquelle für UliCMS 9.8.x verfügbar.

Für UliCMS 9.0.1 wird es zukünftig auch verfügbar sein.

Option "Design für Mobilgeräte nicht auf Tablets verwenden" hinzugefügt

Die Option „Design für Mobilgeräte nicht auf Tablets verwenden“ wurde unter **Einstellungen** > **Design** hinzugefügt.

Diese Option wird in der Datenbank als „no_mobile_design_on_tablet“ gespeichert.
Wenn ein Datensatz mit diesem Key in der Datenbank existiert, ist die Option aktiviert, falls nicht, dann ist die Option nicht aktiviert.

Wenn diese Option aktiviert ist, behandelt UliCMS Tablets wie z.B. das iPad nicht mehr als mobiles Gerät.

Die Funktion is_mobile() gibt bei der Nutzung eines Tablets false zurück.
Dies führt außerdem dazu, dass wenn ein „Design für Mobilgeräte“ festgelegt ist, auf einem Tablet stattdessen das normale Frontend-Design für PCs angezeigt wird.

CKEditor aktualisiert

DIe CKEditor Komponente wurde auf Version 4.5.5 aktualisiert.
CKEditor dient zur einfachen Bearbeitung von HTML-Inhalten in UliCMS.
Mehr Informationen zu den Änderungen in dieser Version finden Sie unter www.ckeditor.com
Falls es nach der Durchführung des Upgrades auf UliCMS 9.8.1 zu Problemen mit dem Editor kommt, leeren Sie bitte Ihren Browsercache.

Parameter $order zu den Funktionen menu() und get_menu() hinzugefügt

Der optionale Parameter $order wurde zu den Funktionen menu(), sowie get_menu() hinzugefügt.

Dieser Parameter ermöglicht, anzugeben, nach welcher Datenbankspalte die Einträge im Menü sortiert werden sollen. Der Standardwert für $order ist „position“.

Mit folgendem Befehlsaufruf können beispielsweise die Menüeinträge des Menüs „produkte“ aufsteigend nach dem Seitentitel sortiert angezeigt werden.

```
menu('produkte', null, true, 'title desc')
```

Automatische Überprüfung auf Aktualisierungen für Pakete

Die Paketverwaltung prüft nun, ob in der Paketquelle Aktualisierungen für die installierten Pakete verfügbar sind. Dabei wird die Versionsnummer des Pakets je nach Status in unterschiedlichen Farben angezeigt.

Grün: Paket ist aktuell.

Rot: Paket ist veraltet und muss aktualisiert werden.

Schwarz: Aktualisierungsstatus konnte nicht ermittelt werden

Prüfung deaktivieren

Sie können die hier beschriebene Prüfung auf Aktualisierungen deaktivieren in dem Sie unter **Einstellungen** > **Expertenmodus** den Schlüssel **disable_package_update_check** mit einem beliebigen Wert anlegen.

Um es anschließend wieder zu deaktivieren löschen Sie diesen Schlüssel einfach.

Neue Funktionen und Klassen

API-Funktion initconfig($key, $value) hinzugefügt

Die API-Funktion initconfig() wurde hinzugefügt.
Diese Funktion legt die Konfigurationseinstellung $key mit dem Wert $value an.
Hier bei ist zu beachten, dass $value vorher mit db_escape() maskiert werden muss, um SQL-Injections zu vermeiden.

Wenn die Konfigurationseinstellung $key bereits existiert, so wird diese nicht überschrieben.
Das ist der einzige Unterschied zwischen initconfig() und setconfig().

Die Funktion gibt **true** zurück, wenn die Einstellung angelegt wurde.
Wenn die Einstellung $key bereits existiert wird **false** zurückgegeben.

API-Funktion get_files($root_dir) hinzugefügt

Die API Funktion get_files wurde hinzugefügt.
Die Funktion hat den Parameter $root_dir.
Die Funktion gibt eine Liste aller Dateien in $root_dir und den Unterordnern von $root_dir als Array zurück. Die Ordner sind nicht in der Rückgabe enthalten..
Die Pfade sind relativ zu $root_dir.

Ein Beispielaufruf:

```
<?php
include_once "init.php";
$files = get_files(ULICMS_ROOT."/content/modules");
var_dump($files);
```

Gibt bei einem frisch installierten System folgendes aus:

```
array(32) {
  [0]=>
  string(27) "fortune/cookies/de/computer"
  [1]=>
  string(27) "fortune/cookies/de/gedichte"
  [2]=>
  string(26) "fortune/cookies/de/gesetze"
  [3]=>
  string(30) "fortune/cookies/de/letzteworte"
  [4]=>
  string(21) "fortune/cookies/de/ms"
  [5]=>
  string(25) "fortune/cookies/de/murphy"
  [6]=>
  string(24) "fortune/cookies/de/namen"
  [7]=>
  string(23) "fortune/cookies/de/quiz"
  [8]=>
  string(25) "fortune/cookies/de/regeln"
  [9]=>
  string(38) "fortune/cookies/de/sicherheitshinweise"
  [10]=>
  string(30) "fortune/cookies/de/sprichworte"
  [11]=>
  string(31) "fortune/cookies/de/sprichwortev"
  [12]=>
  string(27) "fortune/cookies/de/sprueche"
  [13]=>
  string(30) "fortune/cookies/de/stilblueten"
  [14]=>
  string(23) "fortune/cookies/de/tips"
  [15]=>
  string(24) "fortune/cookies/de/unfug"
  [16]=>
  string(27) "fortune/cookies/de/vornamen"
  [17]=>
  string(30) "fortune/cookies/de/warmduscher"
  [18]=>
  string(24) "fortune/cookies/de/witze"
  [19]=>
  string(29) "fortune/cookies/de/wusstensie"
```

```
  [20]=>
  string(25) "fortune/cookies/de/zitate"
  [21]=>
  string(27) "fortune/cookies/en/fortunes"
  [22]=>
  string(36) "fortune/fortune_accordion_layout.php"
  [23]=>
  string(27) "fortune/fortune_lang_de.php"
  [24]=>
  string(27) "fortune/fortune_lang_en.php"
  [25]=>
  string(23) "fortune/fortune_lib.php"
  [26]=>
  string(24) "fortune/fortune_main.php"
  [27]=>
  string(36) "fortune/fortune_register_actions.php"
  [28]=>
  string(21) "fortune/metadata.json"
  [29]=>
  string(24) "fortune/show_fortune.php"
  [30]=>
  string(25) "pfbc_sample/metadata.json"
  [31]=>
  string(32) "pfbc_sample/pfbc_sample_main.php"
}
```

Objekte

Settings-Klasse hinzugefügt

Die Klasse **Settings** wurde hinzugefügt, diese soll zukünftig alle globalen *config Funktionen ersetzen. Sie enthälten die folgenden statischen Funktionen:

Settings::register($key, $value)
Settings::init($key, $value)

Equivalent zu initconfig()

Settings::get($key)

Equivalent zu setconfig()

Settings::set($key, $value)

Equivalent zu **setconfig()**
Achtung!
Diese Funktion unterscheidet sich im Verhalten von setconfig().
$value wird automatisch maskiert, um SQL Injections zu vermeiden.
Ein vorheriges maskieren von $value mit db_escape() führt zu doppelter Maskierung.

Settings::output($key)

Einstellung $key ausgeben.
Achtung!
Die Funktion ist anfällig gegen XSS.
Stattdessen bitte Settings::outputEscaped($key) verwenden.

Settings::outputEscaped($key)

Wie Settings::output($key) mit dem Unterschied, dass HTML-Tags und Anführungszeichen zuvor maskiert werden.

Settings::getLang($key, $language)

Equivalent zu get_lang_config($key, $language).

Settings::delete($key)

Equivalent zu deleteconfig().

Template-Klasse hinzugefügt

Die **Template** Klasse wurde hinzugefügt. Sie soll zukünftig die globalen Funktionen in der Datei templating.php ersetzen.

Die Klassen aus der Datei templating.php sollen nach und nach in diese Klasse verschoben werden.

Die Template-Klasse hat bisher die folgenden Funktionen.

Template::executeModuleTemplate($module, $template)

Bindet ein Template eines Moduls ein. Das Modul muss einen Unterordner „templates“ haben in dem sich eine gleichnamige PHP-Datei befindet. Es ist jedoch möglich, das Template durch ein eigenes zu ersetzen. In dem man im Ordner seines Themes einen Unterordner, der so wie das Modul heißt erstellt. Sofern ein Template mit dem Namen $template vorhanden ist, wird dieses statt dem Standard-Template des Moduls eingebunden.

Original Template Pfad:

```
ULICMS_ROOT/content/modules/$module/templates/$template.php
```

Eigener Template Pfad:

```
ULICMS_ROOT/content/templates/$theme/$module/$template.php
```

Template::footer()

Diese Funktion kann vor dem schließenden </body> Tag in einem Theme aufgerufen werden.
Sie ruft die Hook frontend_footer auf, die dazu dient, JavaScripts einzubinden, die vor dem Seitenende aufgerufen werden können.

Das hilft dabei, die Ladezeiten zu verbessern, da so das Laden der Javascript-Dateien nicht den Seitenaufbau blockiert.

Die Funktion gibt den HTML-Code als String zurück, er wird jedoch nicht automatisch ausgegeben.

Neue Filter

Die folgenden filter wurden hinzugefügt:

unique_identifier

Hash-Wert, der vom Cache genutzt wird, um eine Seite einmalig zu identifizieren.

Willkommensbotschaft für das Loginformular

Es kann nun eine Willkommensbotschaft für das Loginformular festgelegt werden, durch das Setzen der Konfigurationseinstellung **login_welcome_tex**t in dem Expertenmodus der Einstellungen.

Das definieren eines Textes ist auch für mehrere Sprachen möglich, in dem man Einstellungen mit dem Sprachcode am Ende anlegt.
Beispielsweise:
login_welcome_text_de
login_welcome_text_en
login_welcome_text_fr

Sonstige Änderungen

Änderungen der Datenbankstruktur

In dieser Version wurden keine Änderungen an der Datenbank vorgenommen.

Das update-Skript im Upgrade-Paket dient lediglich dazu, die Liste der installed_patches zurückzusetzen.

Änderungen der API

Es wurden die weiter oben in dieser Dokumentation genannten API-Funktionen hinzugefügt. Diese Version von UliCMS enthält keine Änderungen, die die Kompatiblität brechen könnten. Alle Module und Themes die unter UliCMS 9.8.0 funktionieren, werden auch unter UliCMS 9.8.1 funktional sein.