

ULICMS

Neue Features in UliCMS 9.8.3



Stand: 10.04.2016

Inhaltsverzeichnis

Über dieses Dokument.....	3
Zielgruppe.....	3
Übersicht der neuen Features.....	3
Detailbeschreibung der neuen Features.....	3
„Banner“ Klasse hinzugefügt.....	3
„Banners“ Klasse hinzugefügt.....	4
Funktionen zur Klasse „Path“ hinzugefügt.....	4
Klasse „Cache“ hinzugefügt.....	4
Shell-Script „sinfo.php“ hinzugefügt.....	5
Shell-Script sremove.php hinzugefügt.....	6
Shell-Script „version.php“ hinzugefügt.....	6
Funktion containsModule() zur Klasse Page hinzugefügt.....	6
Funktionen zur Klasse ContentFactory hinzugefügt.....	7
Getter Funktionen.....	7
FilterBy Funktionen.....	7
Klasse Request hinzugefügt.....	7
Funktionen zur Template Klasse hinzugefügt.....	7
Neue Hooks.....	8
Aktualisierte Komponenten.....	8
Sonstige Änderungen.....	9
Änderungen der Datenbankstruktur.....	9
Änderungen der API.....	9
Externe Links im Menü.....	9

Über dieses Dokument

Dieses Dokument beschreibt die neuen Features in UliCMS 9.8.2.

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 9.8.2 enthalten waren, sowie die folgenden neuen Features.

- Klassen Banner und Banners hinzugefügt
- Shell-Script sinfo.php hinzugefügt
- Shell-Script sremove.php hinzugefügt
- Shell-Script version.php hinzugefügt
- Funktion containsModule() zur Klasse Page hinzugefügt
- Funktion removeDir() zur Klasse Path hinzugefügt
- Funktionen zur Klasse ContentFactory hinzugefügt
- Klasse Request hinzugefügt
- Klasse Cache hinzugefügt
- Funktionen zur Template Klasse hinzugefügt
- Aktualisierte Komponenten
- Neue Hooks hinzugefügt

Detailbeschreibung der neuen Features

„Banner“ Klasse hinzugefügt

Die Klasse Banner wurde hinzugefügt.

Diese ermöglicht, Banner programmatisch zu erstellen, zu bearbeiten und zu löschen.

Die Banner Klasse hat die folgenden Funktionen:

\$banner->loadByID(\$id)

Banner mit der ID laden.

\$banner->loadRandom()

Zufälligen Banner laden

\$banner->setType(\$type)

Setzt den Typen des Banners.

\$type kann „gif“ oder „html“ sein.

Default ist „gif“.

\$banner->getType()

Gibt den Typen des Banners zurück

\$banner->save()

Wenn id = null ist, wird ein Insert durchgeführt, ansonsten ein Update

\$banner->create()

Führt ein SQL Insert durch

\$banner->update()

Führt ein SQL Update durch.

\$banner->delete()

Löscht den Datensatz, sofern dieser eine ID hat.

„Banners“ Klasse hinzugefügt

Desweiteren wurde die Klasse Banners (Mehrzahl) hinzugefügt.

Diese Funktionen geben jeweils Arrays mit Banner Objekten zurück.

Banners::getAll(\$order = "id")

Alle Banner als Array zurückgeben, geordnet nach der Spalte \$order.

Banners::getByLanguage(\$language, \$order = "language")

Alle Banner mit der Sprache \$language als Array zurückgeben, geordnet nach der Spalte \$order.

Banners::getByType(\$type, \$order = "language")

Alle Banner mit dem Typ \$type als Array zurückgeben, geordnet nach der Spalte \$order.

Funktionen zur Klasse „Path“ hinzugefügt

Die folgende Funktionen wurde zur Klasse „Path“ hinzugefügt:

Path::removeDir(\$dir, \$deleteMe)

Die Funktion leert den Ordner \$dir inklusive aller darin enthaltenen Dateien und Unterordner.

\$deleteMe ist eine Boolean Variable. Wenn \$deleteMe true ist, wird der Ordner \$dir anschließend gelöscht.

Klasse „Cache“ hinzugefügt

Die Klasse „Cache“ wurde hinzugefügt.

Diese enthält die folgenden Funktionen:

Cache::clear()

Alle Caches des Systems leeren.

Cache::clearAPC()

Den APC Cache leeren, sofern APC installiert und aktiv ist.

Cache::store(\$data, \$requestURI = null)

String \$data für die URL \$requestURI im Cache speichern. Wenn \$requestURI == null ist, wird die aktuelle URL genommen. Auf Basis von der \$requestURI wird der cacheFilePath generiert.

Cache::get(\$requestURI = null)

Gibt den Inhalt des Caches für die URL \$requestURI zurück, sofern vorhanden. Ansonsten null
Wenn \$requestURI == null ist, wird die aktuelle URL genommen. Auf Basis von der \$requestURI wird der cacheFilePath generiert.

Cache::buildCacheFilePath(\$request_uri)

Generiert anhand von \$request_uri den Namen zu einer Cache Datei.

Dabei wird \$request_uri, die aktuelle Sprache und der Rückgabewert von is_mobile() zusammengefügt, gehasht und als Pfad zurückgegeben

Shell-Script „sinfo.php“ hinzugefügt

Das Shell-Script sinfo.php wurde hinzugefügt.

Dieses gibt eine Liste aller installierten Module und/oder Themes inklusive Versionsnummer zurück.

Es hat die folgende Syntax:

```
$ ulicms/shell/sinfo.php [all|modules|themes]
```

Das Dollar-Zeichen steht für den Eingabeprompt und darf nicht mit abgetippt werden.

sinfo.php hat einen optional Parameter der „all“, „modules“, oder „themes“ sein kann, wobei „all“ der Standardwert ist.

Der folgende Befehlsaufruf

```
$ ulicms/shell/sinfo.php all
```

gibt beispielsweise folgendes aus:

Modules:

```
- fortune 0.0.5
- pfbc_sample 9.0.2
- slicknav 1.0.2
```

```
3 modules total
```

Themes:

```
- 2016 1.0.3
```

```
1 themes total
```

Shell-Script sremove.php hinzugefügt

Das Shell-Script **shell/sremove.php** wurde hinzugefügt.
Dieses ermöglicht Module und Templates zu deinstallieren.

Es hat die folgende Befehls-Syntax:

```
$ ulicms/shell/sremove.php [module|theme] package1 package2 package3
```

Beispiele:

```
$ ulicms/shell/sremove.php module kontaktformular fortune jquery_tablesorter
```

```
$ ulicms/shell/sremove.php template cleanblue
```

Das Dollerzeichen steht für den Prompt der Kommanodzeile und darf nicht mit eingegeben werden.

Bitte beachten Sie, dass Shellscripts mit einem anderen Account als der Webserver laufen.
Wenn die Dateien für den Shelluser nicht beschreibbar sind, schlägt **sremove** fehl.

Shell-Script „version.php“ hinzugefügt

Das Shell-Script shell/version.php wurde hinzugefügt. Dieses hat keine Parameter und gibt die Version der UliCMS Installation zurück.

Der Befehlsaufruf

```
$ shell/version.php
```

Gibt eine Zeichenkette wie

```
UliCMS Release 9.8.3
```

zurück.

Funktion containsModule() zur Klasse Page hinzugefügt

Es wurde die Funktion containsModule zur Klasse Page hinzugefügt.

Die Funktion hat einen Parameter, den optionalen Parameter \$module.

Der Name eines Moduls kann als Parameter übergeben werden, um zu prüfen, ob die Seite ein bestimmtes Modul enthält.

Diese ist wie im folgenden Beispiel zu nutzen.

```
$page = ContentFactory::loadBySystemnameAndLanguage („willkommen“, „de“);

if($page->containsModule()){
    echo „Seite enthält ein Modul.“;
}

if($page->containsModule („blog“)){
    echo „Seite enthält das Modul blog.“;
}
```

Funktionen zur Klasse **ContentFactory** hinzugefügt

Getter Funktionen

Die folgenden Getter wurden zur Klasse **ContentFactory** hinzugefügt.

Diese Funktionen geben ein Array aus Content-Element zurück (derzeit nur Page).

- **ContentFactory::getAll(\$order = „id“)**
- **ContentFactory::getAllByLanguage(\$language, \$order = „id“)**
- **ContentFactory::getAllByMenu(\$menu, \$order = „id“)**
- **ContentFactory::getAllByMenuAndLanguage(\$menu, \$language, \$order = „id“)**

FilterBy Funktionen

Desweiteren wurden die folgenden Funktionen hinzugefügt.

Diese bekommen als ersten Parameter ein Resultat einer getAllX() Funktion zurück und geben ein gefiltertes Array zurück.

- **ContentFactory::filterByEnabled(\$elements, \$enabled = 1)**
- **ContentFactory::filterByCategory(\$elements, \$category = 1)**
- **ContentFactory::filterByAutor(\$elements, \$autor = 1)**
- **ContentFactory::filterByLastChangeBy(\$elements, \$lastchangeby = 1)**

Klasse **Request** hinzugefügt

Die Klasse „Request“ wurde hinzugefügt. Die Klasse enthält Fragen zur Verarbeitung des HTTP-Requests. Die Klasse enthält derzeit die folgenden Funktionen:

Request::getStatusCodeByNumber(\$nr)

Gibt einen vollständigen HTTP Status anhand der Statusnummer als String zurück.

Request::redirect(\$url = "http://www.ulicms.de", \$status = 302)

Leitet mit dem HTTP Status \$status auf die URL \$url weiter.

In zukünftigen UliCMS Versionen werden weitere Funktionen zur Klasse hinzugefügt.

Funktionen zur **Template Klasse** hinzugefügt

Die folgenden Funktionen wurden zur Klasse **Template** hinzugefügt.

Template::logo()

Gibt das Logo aus.

Template::year()

Gibt das aktuelle Jahr aus.

Template::getMotto()

Gibt das Motto der Webseite zurück.

Template::motto()

Gibt das Motto der Webseite aus.

Neue Hooks

Die folgenden neuen Hooks wurden hinzugefügt:

- before_set_language_by_domain
- after_set_language_by_domain
- before_include_custom_lang_file
- after_include_custom_lang_file
- before_custom_lang
- after_custom_lang
- before_set_locale_by_language
- after_set_locale_by_language
- before_ajax_handler
- after_ajax_handler

Aktualisierte Komponenten

Die folgenden integrierten Komponenten wurden aktualisiert.

- Mobile_Detect (Version 2.0.20)
- CKEditor (Version 4.5.7)

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

An der API wurden des weiteren die folgenden Änderungen durchgeführt.

Die folgende Funktion ist nun statisch:

ContentFactory::loadBySystemnameAndLanguage(\$name, \$language)

Externe Links im Menü

Links auf externe Seiten werden im Menü nun direkt verlinkt, statt wie bisher über eine HTTP-Weiterleitung zu gehen.