



Neue Features in UliCMS 2017.4



Inhaltsverzeichnis

Über dieses Dokument.....	1
Zielgruppe.....	1
Was ist UliCMS?.....	1
Zusammenfassung der Änderungen.....	1
Neue Features.....	1
Neue API-Funktionen.....	2
Neue Hooks.....	3
Sonstige Änderungen.....	3
Entfernte Features.....	4
Detailbeschreibung der neuen Features.....	4
Gruppen können nun Sprachen zugeordnet bekommen.....	4
Neuer Inhaltstyp "Snippet".....	4
Zulässige Tags können nun beschränkt werden.....	5
Hinweis wenn Mobile_Detect nicht installiert ist.....	5
Benutzer können nach Gruppe gefiltert werden.....	5
Es kann nun eine Standardsprache je Benutzer ausgewählt werden.....	6
main_class Direktive wird nun ausgewertet.....	7
settings können nun über die metadata.json eines Moduls initialisiert werden.....	7
Model Klasse für 'Language' hinzugefügt.....	7
Model Klasse für 'Category' hinzugefügt.....	7
Bei fehlenden Sprachdateien erfolgt ein Fallback auf Englisch.....	7
Attribut \$newWindow zu MenuEntry hinzugefügt.....	7
"Gruppe" in der Benutzerliste verlinkt auf "Gruppe bearbeiten".....	8
metadata.json: Attribut "objects" umbenannt in "models".....	8
metadata.json: Alias für "actions" eingeführt "views".....	8
REQUEST_METHOD spezifische Controller-Funktionen.....	8
Link zu der Settings-Seite eines Moduls können nun angepasst werden.....	8
Klasse "Model" erweitert.....	9
Neuer Inhaltstyp "Link zu Sprache".....	10
Berechtigungsabfrage für Controller-Funktionsaufrufe eingebaut.....	11
Encoding auf utf8mb4 umgestellt.....	11
Durchführung der Installation.....	12
Durchführung eines Upgrades.....	12
Benötigen Sie Hilfe?.....	14
Abbildungsverzeichnis.....	14

Über dieses Dokument

Dieses Dokument beschreibt die neuen Features in UliCMS 2017.4.

Zielgruppe

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

Was ist UliCMS?

UliCMS ist eine flexible Open Source Web Content Management Lösung aus Deutschland, die unter einer BSD-Lizenz mit Advertising-Klausel lizenziert ist.

UliCMS ermöglicht die Erstellung und Verwaltung von kleinen und mittelgroßen Websites.

Zusammenfassung der Änderungen

Neue Features

- Gruppen können nun Sprachen zugeordnet bekommen
- Neuer Inhaltstyp "Snippet"
- Zulässige Tags können nun beschränkt werden
- Es wird nun ein Hinweis angezeigt, wenn man ein "Design für Mobilgeräte" festlegt und Mobile_Detect nicht installiert ist.
- Benutzer können nach Gruppe gefiltert werden
- Es kann nun eine Standardsprache je Benutzer ausgewählt werden.
- `main_class` Direktive wird nun ausgewertet. Methoden (render, admin, Hooks, Filter) in der `main_class` werden automatisch aufgerufen.
- settings können nun über die `metadata.json` eines Moduls initialisiert werden
- Model Klasse für 'Language' hinzugefügt
- Model Klasse für 'Category' hinzugefügt
- Bei fehlenden Sprachdateien erfolgt ein Fallback auf Englisch
- Attribut `$newWindow` zu `MenuEntry` hinzugefügt
- "Gruppe" in der Benutzerliste verlinkt auf "Gruppe bearbeiten"
- `metadata.json`: Attribut "objects" umbenannt in "models"
- `metadata.json`: Alias für "actions" eingeführt "views"
- `REQUEST_METHOD` spezifische Controller-Funktionen

- Link zu der Settings-Seite eines Moduls können nun angepasst werden
- Klasse "Model" erweitert
- Neuer Inhaltstyp "Link zu Sprache"
- Berechtigungsabfrage für Controller-Funktionsaufrufe eingebaut
- Encoding auf utf8mb4 umgestellt, UliCMS benötigt mindestens MySQL bzw. MariaDB 5.5

Neue API-Funktionen

- Request::getMethod()
- Request::isGet()
- Request::isPost()
- Request::isHead()
- Request::hasVar()
- Request::isSSL()
- Request::getIp()
- Request::getProtocol()
- Request::isHeaderSent(\$header)
- ModuleHelper::buildModuleResourcePath(\$module, \$path)
- ModuleHelper::buildMethodCall(\$sClass, \$sMethod)
- CustomData::getCustomDataOrSetting(\$name)
- ModuleHelper::underscoreToCamel(\$str)
- get_secure_translation()
- secure_translation()
- secure_translate()
- ArrayHelper::insertBefore(\$input, \$index, \$element)
- ArrayHelper::insertAfter(\$input, \$index, \$element)
- faster_in_array()
- Page→getEmbeddedModules()
- Template::doctype()
- Template::getDoctype()
- Template::ogHTMLPrefix()
- Template::getOgHTMLPrefix()
- iddefine(\$key, \$value)
- JsonResult(\$data, \$status = 200)
- RawJsonResult(\$data, \$status = 200)

- `HTMLResult($data, $status = 200)`
- `TextResult($data, $status = 200)`
- `getStringLengthInBytes($data)`
- `HttpStatusCode` Konstanten

Neue Hooks

- `admin_footer`
- `admin_copyright_footer_left`
- `admin_copyright_footer_right`

Sonstige Änderungen

- Nach dem Erstellen einer Seite wird nun auf die Bearbeiten-Ansicht statt auf die Listenansicht weitergeleitet
- Die Funktionen `Template::executeModuleTemplate()` und `Template::executeDefaultOrOwnTemplate()` akzeptieren nun auch Templatennamen mit Dateieindung
- Funktion `getCurrentLanguage()` - Der Parameter `$current` ist nun per Default `false`
- `getCurrentURL()` fügt nun nicht mehr den Port an die Domain bei `https://`
- Auf Fehlerseiten wird der "Bearbeiten" Button nicht mehr angezeigt
- Bugfix: Ein Fehler bei dem ein Menüpunkt im Backend fälschlicherweise nicht angezeigt wurde behoben.
- Bugfix: Das Feld "Modul" wird nun beim Klonen einer Seite korrekt ausgefüllt.
- Bugfix: `$user->update()` überschreibt die ID nicht mehr
- Parameter `$query` von `Database::query()` und `Database::pQuery()` in `$sql` umbenannt
- `uninstall()` Methode in `main_class` wird nun aufgerufen
- Der "Anzeigen" Button beim Bearbeiten vom Inhaltstyp "Knotenpunkt" wird nicht mehr angezeigt
- Bugfix: "Letzte Änderungen" wenn `lastchangeby` nicht gesetzt ist, wird nun der Ersteller der Seite als Durchführender der letzten Änderung angezeigt
- Bugfix: Listen zeigen nun nur noch aktive Datensätze
- Bugfix: Die globale Funktion `getAllPages()` gab nicht alle Seiten zurück wenn `$exclude_hash_links = false` ist.

Entfernte Features

- Funktion "Menüeinträge verschieben" wurde in ein optionales Modul ausgelagert.
https://extend.ulicms.de/move_menu_items.html

Detailbeschreibung der neuen Features

Gruppen können nun Sprachen zugeordnet bekommen

Es ist nun möglich einer Benutzergruppe Sprachen zu zuweisen.

Es wurden Checkboxes für jede Sprache in der Gruppenverwaltung hinzugefügt.

Wenn keine Checkboxes gesetzt sind, findet keine Einschränkung statt. Wenn eine oder mehrere Checkboxes gesetzt sind, können alle Benutzer in einer Gruppe nur Inhalte in den gewählten Sprachen bearbeiten.

Sprachen

☐ Deutsch

☒ English

*Abbildung 1:
Sprachenauswahl
in der Gruppe*

Neuer Inhaltstyp "Snippet"

Der Inhaltstyp „Snippet“ bzw. „Schnippel“ wurde hinzugefügt.

Ein „Snippet“ ist ein Text, der über einen Shortcode in andere Inhalte eingefügt werden kann. Snippets sind nicht direkt verlinkbar.

Schnipsel Code

[include=84]

Mit diesem Shortcode können Sie diesen Schnipsel in eine andere Seite einbinden.

Abbildung 2: Schnipsel Code

Zulässige Tags können nun beschränkt werden

Die erlaubten HTML-Tags können nun auf Gruppenebene festgelegt werden. Die Filterung erfolgt beim Speichern einer Seite mit der Funktion `strip_tags` (<http://php.net/manual/de/function.strip-tags.php>). Wenn das Textfeld leer ist, erfolgt keine Filterung.

Zulässige Tags

`<a> <abbr> <address> <area> <article> <aside> <audio> <bdi> <bdo> <blo`

Leer = alle Tags erlaubt

Abbildung 3: Textfeld zur Filterung von HTML-Tags

Der Standardwert ist:

`<a><abbr><address><area><article><aside><audio><bdi><bdo><blockquote>

<butt
on><canvas><caption><cite><code><4col><colgroup><command><data><datalist><dd><d
etails><dfn><div><dl><dt><embed><fieldset><figcaption><figure><footer><form><
header><hgroup><hr><i><iframe><input><ins><kbd><keygen><label><legend><map
><mark><math><menu><meter><nav><object><optgroup><option><output><p><param><p
re><progress><q><rp><rt><ruby><s><samp><section><select><small><source>
<sub><summary><sup><svg><table><tbody><td><textarea><tfoot><th><thead><time><tr><trac
k><u><var><video><wbr>`

Hinweis wenn Mobile_Detect nicht installiert ist.

Es wird nun ein Hinweis angezeigt, wenn man ein "Design für Mobilgeräte" festlegt und Mobile_Detect nicht installiert ist.

impro17

Design für Mobilgeräte

Um dieses Feature zu nutzen müssen Sie Mobile_Detect installieren.

Abbildung 4: Hinweis auf fehlendes Mobile_Detect Modul

Benutzer können nach Gruppe gefiltert werden

In der Benutzerverwaltung wurde eine Dropdownliste zum Filtern der Benutzerliste nach Gruppen hinzugefügt.

Gruppe

Administrator

Ein Datensatz gefunden.




ID	Benutzername	Nachname	Vorname	E-Mail	Gruppe	Bearbeiten	Löschen
1	 admin	Muster	Max	admin@deruli.de	Administrator		

Abbildung 5: Gruppenfilter

Außerdem wurde in der Gruppenverwaltung ein Icon, welches auf die gefilterte Benutzerliste verlinkt, hinzugefügt.




ID	Name	Standard			
1	Administrator	Nein			

Abbildung 6: Icon mit Link zur Gruppe

Es kann nun eine Standardsprache je Benutzer ausgewählt werden

Es kann nun eine Standardsprache je Benutzer festgelegt werden. Zum Loginformular wurde dir vorausgewählte Option „[Standard]“ hinzugefügt. Sofern nicht anders ausgewählt, wird die eingestellte Sprache des Benutzers geladen. Wenn der Benutzer keine Sprache gesetzt hat, und beim Login keine Sprache auswählt, wird die in der Einstellung system_language konfigurierte Sprache verwendet.

Standardsprache

Deutsch

Abbildung 7: Auswahl für Standardsprache

Bitte authentifizieren Sie sich:

Benutzername	deruli
Passwort
Passwort anzeigen	<input type="checkbox"/>
Sprache	[Standard]
<input type="button" value="Login"/>	

[\[Passwort zurücksetzen\]](#)

Abbildung 8: Sprachauswahl beim Login

main_class Direktive wird nun ausgewertet

Sofern definiert, wird die main_class Direktive in der Datei „metadata.json“ nun ausgewertet. Dabei wird ein Controller als main_class angegeben, der alle Filter, Hooks und sonstige Modulfunktionen enthält.

Funktionsnamen werden dabei in Camel-Case statt mit Unterstrichen angegeben, aus before_content wird z.B. beforeContent(). Da die Funktionen in einer Klasse enthalten sind, müssen diese nicht mehr mit dem Namen des Moduls geprefixt werden. Schauen Sie sich als Beispiel das Modul fortune2 unter der folgenden URL an <https://github.com/derUli/fortune2>

settings können nun über die metadata.json eines Moduls initialisiert werden

Es ist nun möglich Einstellungen mit Standardwerten in der metadata.json eines Moduls bekannt zu machen. Beim Leeren des Caches wird die „settings“ Direktive ausgewertet, und die darin enthaltenen Optionen per Settings::register() angelegt.

```
{
    „settings“ : {
        „option1“ : „Hallo“,
        „option2“ : „Welt!“
    }
}
```

Model Klasse für 'Language' hinzugefügt

Es wurde eine Klasse Model für die Datenbanktabelle „languages“ hinzugefügt.

Model Klasse für 'Category' hinzugefügt

Es wurde eine Klasse Model für die Datenbanktabelle „categories“ hinzugefügt.

Bei fehlenden Sprachdateien erfolgt ein Fallback auf Englisch

Wenn UliCMS, ein Modul oder ein Design keine Sprachdatei für die ausgewählte Sprache enthält, wird stattdessen die Englische Sprachdatei geladen.

Attribut \$newWindow zu MenuEntry hinzugefügt

Das Boolean Attribut \$newWindow wurde zum Konstrukt der Klasse \$newWindow hinzugefügt. Wenn \$newWindow = true ist, wird zum Link im Menü das Target auf „_blank“ gesetzt. Dies bewirkt, dass der Link in einem neuen Fenster geöffnet wird.

"Gruppe" in der Benutzerliste verlinkt auf "Gruppe bearbeiten"

Die Spalte Gruppe in der Benutzerliste verlinkt nun auf die „Bearbeiten“ Ansicht der Gruppe.

metadata.json: Attribut "objects" umbenannt in "models"

Das Attribut „objects“ wurde in „models“ umbenannt. „objects“ funktioniert aus Kompatibilitätsgründen jedoch weiterhin.

metadata.json: Alias für "actions" eingeführt "views"

Das Attribut „actions“ wurde in „views“ umbenannt. „actions“ funktioniert aus Kompatibilitätsgründen jedoch weiterhin.

REQUEST_METHOD spezifische Controller-Funktionen

Es können Funktionen erstellt werden, die nur bei bestimmten Request-Typen ausgeführt werden. Beispielsweise die folgende URL ?sClass=MyClass&sMethod=myFunction.

Ruft, wenn es ein HTTP Post Request ist, sofern vorhanden die Funktion myFunctionPost() und wenn es ein HTTP Get Request ist, die Funktion myFunctionGet() auf. Wenn keine Funktion für den Request Typ existiert, wird die Funktion myFunction() aufgerufen.

Link zu der Settings-Seite eines Moduls können nun angepasst werden

Der Text des Links zu der Settings-Page eines Moduls kann nun angepasst werden. Erstellen Sie dazu in der main_class die Funktion getSettingsLinkText() geben Sie den gewünschten Linktext als String zurück.

4. oneclick_upgrade 1.2
Kein Embed Modul

[Einstellungen] [Löschen]

Abbildung 9: Anpassbarer Linktext

Klasse "Model" erweitert

Die Klasse „Model“ wurde erweitert. Die Klasse enthält nun die folgenden Funktionen.

__construct(\$id = null)

loadByID(\$id)

save()

fillVars(\$query = null)

insert()

update()

delete()

setID(\$id)

getID()

Neuer Inhaltstyp "Link zu Sprache"

Der neue Inhaltstyp „Link zu Sprache“ wurde hinzugefügt. Mit diesem kann man die Startseite einer anderen Sprache im Menü verlinken. Sofern Domain2Language Mapping verwendet wird, wird die der verlinkten Sprache zugewiesene Domain verlinkt. Ansonsten index.php?language=[Sprachcode]



Typ

- ☐ Seite
- ☐ Artikel
- ☐ Schnipsel
- ☐ Liste
- ☐ Link
- ☒ **Link zu Sprache**
- ☐ Knotenpunkt
- ☐ Bild
- ☐ Modul
- ☐ Video
- ☐ Audio

Abbildung 10: Checkbox für Inhaltstyp "Link zu Sprache"



Link zu Sprache

Link zu Sprache

Deutsch ▼

Abbildung 11: Auswahl Link zu Sprache

Berechtigungsabfrage für Controller-Funktionsaufrufe eingebaut

Es wurde eine Berechtigungsabfrage für den Aufruf von Controller-Funktionen durch den Anwender implementiert.

Die Berechtigungen können wie folgt in der Datei metadata.json eines Moduls gesetzt werden.

```
{
  „controller_function_permissions“ : {
    „MyClass::myFunction“ : „my_permission“,
    „MyClass::otherFunction“ : „other_permission“,
    „AnotherClass::doMagic“ : „magic_permission“
  }
}
```

Encoding auf utf8mb4 umgestellt

Die Zeichenkodierung bei Neuinstallationen ist nun utf8mb4. Dies ermöglicht die Verwendung von Emojis. Utf8mb4 wird seit MySQL 5.5 unterstützt. Damit benötigt UliCMS nun mindestens MySQL 5.5. Bei einem Update wird die Zeichenkodierung in der Datenbank nicht automatisch konvertiert.

Zum Update von bestehenden Datenbanken verwenden Sie das folgende UliCMS Modul:

https://extend.ulicms.de/convert_to_utf8mb4.html

Durchführung der Installation

1. Entpacken Sie die heruntergeladene zip-Datei. Verbinden Sie sich per FTP mit dem Webserver und laden Sie den Inhalt des Ordners „ulicms“ in den Hauptordner oder einen Unterordner hoch.
2. Wenn Sie UliCMS in dem Hauptordner hochgeladen haben, rufen Sie [www.ihredomain.de]/installer im Browser auf.
3. Wenn Sie UliCMS in einem Unterordner hochgeladen haben, rufen Sie [www.ihredomain.de]/[ihr-unterordner]/installer im Browser auf. Die eckigen Klammern sind Platzhalter.
4. Folgen Sie den Anweisungen des Installationsassistenten.
5. Löschen Sie nach erfolgreichem Abschluss der Installation den Ordner „installer“ vom Server.

Durchführung eines Upgrades

Ein Upgrade kann nur von UliCMS 2017.1, UliCMS 2017.2 oder UliCMS 2017.3 auf UliCMS 2017.3 erfolgen. Ein Upgrade von UliCMS 9.8.x und älteren Versionen erfordert dass Sie zuerst auf UliCMS 2017.1 aktualisieren und anschließend auf UliCMS 2017.3.

1. Entpacken Sie das heruntergeladene Upgrade-Paket.
2. Verbinden Sie sich per FTP mit dem Webserver, und laden Sie den Inhalt des Ordners „ulicms“ in den Ordner, in dem sich Ihre UliCMS Installation befindet hoch. Wenn Ihr FTP-Client Sie fragt, ob bereits vorhandene Dateien ersetzt werden sollen, bestätigen Sie dieses für alle Dateien und Ordner.
3. Überprüfen Sie ob die Dateiberechtigungen weiterhin korrekt gesetzt sind. Die Datei „update.php“, sowie der Ordner „content“ müssen die Dateirechte auf 0755 stehen haben.

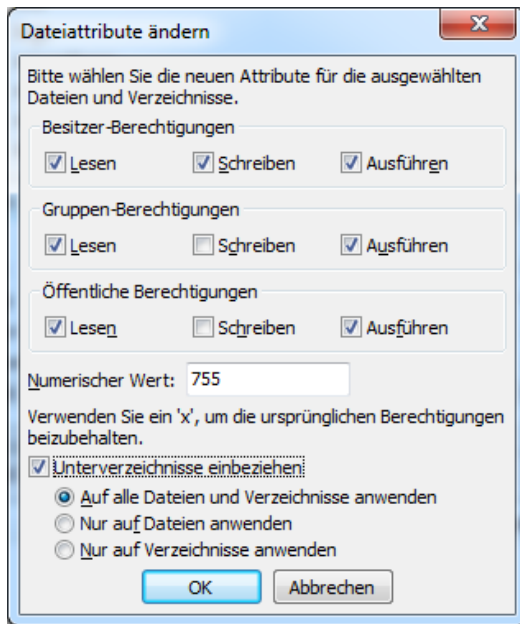


Abbildung 12: Abbildung, setzen der Datei-Berechtigungen auf dem FTP-Server

4. Wenn Sie UliCMS in dem Hauptordner installiert haben, rufen Sie [www.ihredomain.de]/update.php im Browser auf.
5. Wenn Sie UliCMS in einen Unterordner hochgeladen haben, rufen Sie [www.ihredomain.de]/[ihr-unterordner]/update.php im Browser auf. Die eckigen Klammern sind Platzhalter. Das update-Skript aktualisiert die Datenbank.
6. Loggen Sie sich nun in das CMS ein, prüfen Sie, ob eine Paketquelle für diese Version von UliCMS ausgewählt ist.
7. Falls Patches für UliCMS verfügbar sind, installieren Sie diese.
8. Leeren Sie den Cache des CMS.
9. Rufen Sie „Pakete“ auf, und merken Sie sich alle Pakete, deren Versionsnummer rot markiert ist. Installieren Sie die neuen Versionen dieser Pakete.
10. Falls Probleme mit Modulen und Themes auftreten, die nicht aus der Paketquelle stammen, prüfen Sie, ob von diesen Paketen beim Hersteller eine neue Version verfügbar ist, und installieren Sie diese gegebenenfalls.

Benötigen Sie Hilfe?

Falls Sie Fragen haben, oder auf Probleme bei der Benutzung von UliCMS stoßen können Sie den Entwickler entweder auf dem UliCMS unter <http://www.ulicms.de/kontakt.html> oder auf Facebook <https://www.facebook.com/UliCMS9> kontaktieren.

Abbildungsverzeichnis

Abbildung 1: Sprachenauswahl in der Gruppe.....	4
Abbildung 2: Schnipsel Code.....	4
Abbildung 3: Textfeld zur Filterung von HTML-Tags.....	5
Abbildung 4: Hinweis auf fehlendes Mobile_Detect Modul.....	5
Abbildung 5: Gruppenfilter.....	6
Abbildung 6: Icon mit Link zur Gruppe.....	6
Abbildung 7: Auswahl für Standardsprache.....	6
Abbildung 8: Sprachauswahl beim Login.....	6
Abbildung 9: Anpassbarer Linktext.....	8
Abbildung 10: Checkbox für Inhaltstyp "Link zu Sprache".....	10
Abbildung 11: Auswahl Link zu Sprache.....	10
Abbildung 12: Abbildung, setzen der Datei-Berechtigungen auf dem FTP-Server.....	13