

ULICMS

Neue Features in UliCMS 2018.1



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
Neue Filter.....	3
Neue Model-Klassen.....	3
Sonstige Änderungen.....	3
Bugfixes.....	4
Entfernte Features.....	4
Detailbeschreibung der neuen Features.....	5
MVC-Strukturierung des Backends.....	5
Inhaltstypen mit eigenen Feldern.....	6
Parameter \$htmlAttributes zur Funktion ModuleHelper::buildMethodCallForm() hinzugefügt....	8
Der Confirm-Dialog "Wirklich löschen?" kann über eine CSS-Klasse an einem Formular ausgelöst werden.....	8
Neues Metadaten-Attribut „shy“.....	8
Es können nun Standardwerte für benutzerdefinierte Werte gesetzt werden.....	9
User->resetPassword() implementiert.....	9
Neue Metadaten Direktive „action_permissions“.....	9
Parameter \$order zu allen Funktionen im User-Manager hinzugefügt.....	10
CodeMirror kann per CSS-Klasse auf eine Textarea angewendet werden.....	10
Screenshots bei SimpleInstall v2 Paketen.....	10
Durchführung der Installation.....	11
Durchführung eines Upgrades.....	11
Benötigen Sie Hilfe?.....	13
Abbildungsverzeichnis.....	13

Über dieses Dokument

Dieses Dokument beschreibt die neuen Features in UliCMS 2018.1.

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

- Teile des Backends wurden in Core-Module verschoben und in ein MVC-Pattern umstrukturiert
- Module können nun Inhaltstypen mit eigenen Formularfeldern erstellen
- Neuer Parameter `$htmlAttributes` (Array) zu der Funktion `ModuleHelper::buildMethodCallForm()` hinzugefügt
- Der Confirm-Dialog "Wirklich löschen?" kann über eine CSS-Klasse an einem Formular ausgelöst werden.
- Durch das booleanische "shy" Attribut in einer Metadatenfile können Module und Themes im Paketmanager versteckt werden.
- Es können nun Standardwerte für benutzerdefinierte Werte gesetzt werden
- `User->resetPassword()` implementiert
- Neue Direktive `action_permissions` in den Metadaten eines Moduls wird nun ausgewertet
- Parameter `$order` zu allen Funktionen im User-Manager hinzugefügt
- `CodeMirror` kann per CSS-Klasse auf eine Textarea angewendet werden.
- Beim Upload eines SimpleInstall v2 Pakets wird nun ein Screenshot angezeigt, sofern vorhanden.

Neue API-Funktionen

- `esc()`
- `_esc()`
- `ArrayHelper::take($count, $data)`
- `ModuleHelper::buildMethodCallUrl()`
- `ModuleHelper::deleteButton()`
- `ModuleHelper::buildQueryString()`
- `ModuleHelper::buildMethodCallUploadForm()`
- `ModuleHelper::buildRessourcePath()`
- `ModuleHelper::buildMethodCallButton()`
- `ModuleHelper::endForm()`
- `CustomData::setDefault()`
- `CustomData::getDefaultAsJSON()`
- `Settings::mappingStringToArray()`
- `Banners::getByCategory()`
- `str_contains()`
- `register_translation()`
- `add_translation()`
- `json_readable_encode()`
- `getLanguageByDomain()`

Neue Hooks

- before_functions
- after_functions
- before_backend_header
- after_backend_header
- before_login_form
- after_login_form
- before_reset_password_form
- after_after_password_form
- before_backend_run_methods
- after_backend_run_methods

Neue Filter

- session_data
- body_classes

Neue Model-Klassen

- Audio
- Video

Sonstige Änderungen

- Wenn möglich wird bei PHP Fehlern der HTTP Status "500 Internal Server Error" gesendet.
- Funktion is_ajax_request() verschoben nach Request::isAjaxRequest()
- Die Logik, welche Felder abhängig vom ausgewählten Inhaltstyp ein- und ausblendet wurde neu implementiert
- Benutzer bei denen "Ist Admin" gesetzt ist, dürfen nun alle Seiten bearbeiten, unabhängig von den gesetzten Berechtigungen

Bugfixes

Die Gruppe "Administratoren" wird bei der Installation wieder erstellt

- Auf einigen Systemen wurde beim Klick auf das Icon "Cache leeren" die Ladeanimation nicht angezeigt
- Der Menüpunkt "Update" wurde nicht angezeigt, wenn der Benutzer zwar zu update_system berechtigt ist, aber "Ist Admin" nicht gesetzt ist.
- Bei einigen Mailservern funktionierte der Versand von "Passwort zurücksetzen" E-Mails nicht.
- Ein Problem, bei dem die Datenbanktabelle password_reset bei der Installation und bei einem Update nicht erstellt wurde, wurde behoben.
- Model-Klasse "Banner" speichert nun korrekt
- Der Fallback-Link zu HTML5-Medien für nicht unterstützte Browser wird nun korrekt erstellt
- "Anzeigen" Link in der Seitenliste war fehlerhaft, wenn Domain2Language Mapping aktiv ist und UliCMS in einem Unterordner installiert ist.

Entfernte Features

- Funktion ulicms_redirect() entfernt.
Nutzen Sie stattdessen Request::redirect()

Detailbeschreibung der neuen Features

MVC-Strukturierung des Backends

Große Teile des Backends wurden in Core-Module verschoben und in ein MVC-Pattern umstrukturiert. Bitte beachten Sie, dass dies ein Work-in-Progress ist.

Die Core-Module lassen sich unter „Pakete“ über die Option „Core-Module anzeigen“ einblenden.



[\[Paket installieren\]](#)

Core-Module anzeigen

Installierte Module

Hier finden Sie eine Auflistung der installierten Module.

1. **bootstrap 3.3.7**
Kein Embed Modul

[\[Löschen\]](#)

2. **core_content 2018.1**
Kein Embed Modul

3. **core_help 2018.1**
Kein Embed Modul

4. **core_history 2018.1**
Kein Embed Modul

Abbildung 1: Anzeige der Core-Module in der Liste der Module

Inhaltstypen mit eigenen Feldern

Module können nun Inhaltstypen mit eigenen Formularfeldern erstellen.

Der Name eines Felds muss eindeutig sein. Wenn mehrere Inhaltstypen mit einem gleich benannten Feld existieren, führt dies zu Fehlern beim Speichern.

Das Modul `custom_types_example` demonstriert die Definition eines Inhaltstyps mit eigenen Feldern. Dieses Modul ist im Installationspaket enthalten und kann bei GitHub und per UliCMS eXtend bezogen werden. Sie können es deinstallieren, wenn sie es nicht benötigen.

Feldtypen

- TextField
- MultilineTextField
- EmailField
- MonthField
- DatetimeField
- NumberField
- ColorField
- HtmlField
- SelectField
- MultiSelectField

Produkteigenschaften

Preis *

7,99

Ohne Mehrwertsteuer

Spezielle Mehrwertsteuer

7

Teaser Text

Super tolles Produkt

Kundendienst

service@company.de

Verfügbar bis

Dezember 2019

Datum

09.11.2017 13:55:01

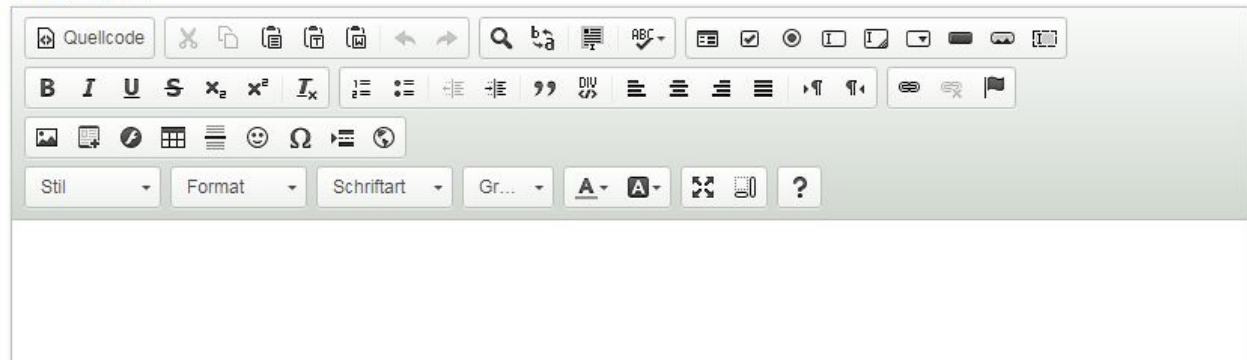
Lagermenge

26

Farbe

#69B9FF

HTML Editor



The HTML Editor interface includes a toolbar with the following elements:

- Buttons for Quellcode, Cut, Copy, Paste, Undo, and Redo.
- Buttons for Find, Replace, Spell Check, and ABC (Text to Speech).
- Buttons for Bold, Italic, Underline, Strikethrough, Subscript, and Superscript.
- Buttons for Bulleted List, Numbered List, Decrease Indent, Increase Indent, Left Align, Right Align, Center Align, and Justify.
- Buttons for Link, Unlink, and Flag.
- Buttons for Image, Table, Table of Contents, Smiley, Omega, and Help.
- Dropdown menus for Stil, Format, and Schriftart.
- Buttons for Gr... (Grid), Font Color, Background Color, and a Help icon.

Abbildung 2: Demo-Modul *custom_types_example*

Parameter \$htmlAttributes zur Funktion ModuleHelper::buildMethodCallForm() hinzugefügt

Der optionale Parameter \$htmlAttributes wurde zur Funktion ModuleHelper::buildMethodCallForm() hinzugefügt. Dies ist ein assoziatives Array für optionale Attribute die zum <form> HTML-Tag hinzugefügt wird.

Die vollständige Signatur der Funktion

```
ModuleHelper::buildMethodCallForm($sClass, $sMethod, $otherVars = array(),  
$requestMethod = "post", $htmlAttributes = array())
```

Der Confirm-Dialog "Wirklich löschen?" kann über eine CSS-Klasse an einem Formular ausgelöst werden

Um dieses Feature zu nutzen fügen Sie die CSS-Klasse „delete-form“ an einem <form> Tag hinzu und stellen Sie sicher, dass die Übersetzung ASK_FOR_DELETE in Javascript geladen wird.

Code-Beispiel Übersetzung für „Wirklich löschen?“ Dialog laden

```
$translation = new JSTranslation ();  
$translation->addKey ( "ask_for_delete" );  
$translation->renderJS ();
```

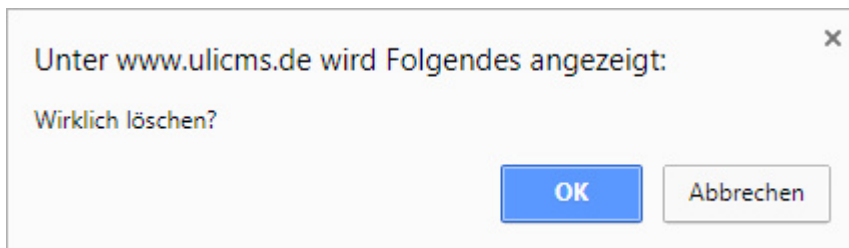


Abbildung 3: Sicherheitsnachfrage, ob ein Datensatz wirklich gelöscht werden soll

Neues Metadaten-Attribut „shy“

Durch das booleanische "shy" Attribut in einer Metadatenfile können Module und Themes im Paketmanager versteckt werden.

```
{  
  „shy“ : true  
}
```

Es können nun Standardwerte für benutzerdefinierte Werte gesetzt werden

Es können nun Standardwerte für benutzerdefinierte JSON Werte gesetzt werden.

Codebeispiel

```
CustomData::setDefault(„my_var“, „Default Value“)
```

User->resetPassword() implementiert

Die Funktion User→resetPassword() wurde implementiert. Die Funktion war zwar schon vorher vorhanden, jedoch war erzeugte sie nur eine NotImplementedException.

Codebeispiel

```
$user = new User();  
$user->loadByUsername("maxmuster");  
$user->resetPassword();
```

Neue Metadaten Direktive „action_permissions“

Die neue Direktive action_permissions in den Metadaten eines Moduls wird nun ausgewertet.

Es kann nun die notwendige Berechtigung für eine Action in der Metadatenfile eines Moduls festgelegt werden.

Codebeispiel

```
{  
    "actions" : {  
        "my_action" : "templates/my_action.php"  
    },  
    "action_permissions" : {  
        "my_action" : "my_permission"  
    }  
}
```

Parameter \$order zu allen Funktionen im User-Manager hinzugefügt

Der optionale Parameter \$order wurde zu den folgenden Funktionen im UserManager hinzugefügt. Mit diesem kann man die Sortierung der zurückgegebenen Benutzer festlegen.

```
UserManager->getUsersByGroupId($gid, $order = "id")
UserManager->getAllUsers($order = "id")
UserManager->getLockedUsers($locked = true, $order = "id")
```

CodeMirror kann per CSS-Klasse auf eine Textarea angewenden werden

Der CodeMirror HTML-Editor kann nun doch anfügen der CSS-Klasse „codemirror“ an eine Textarea angefügt werden. Der Typ des zu bearbeitenden Quellcodes kann über das Attribut data-mimetype festgelegt werden. Der Standard Mimetype ist „text/html“.

Codebeispiel

```
<textarea name="sql_code" id="sql_code" class="codemirror" data-mimetype="text/x-sql"><?php echo htmlspecialchars($_SESSION["sql_code"])?></textarea>
```

Screenshots bei SimpleInstall v2 Paketen

Beim Upload eines SimpleInstall v2 Pakets wird nun ein Screenshot angezeigt, sofern vorhanden.

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.

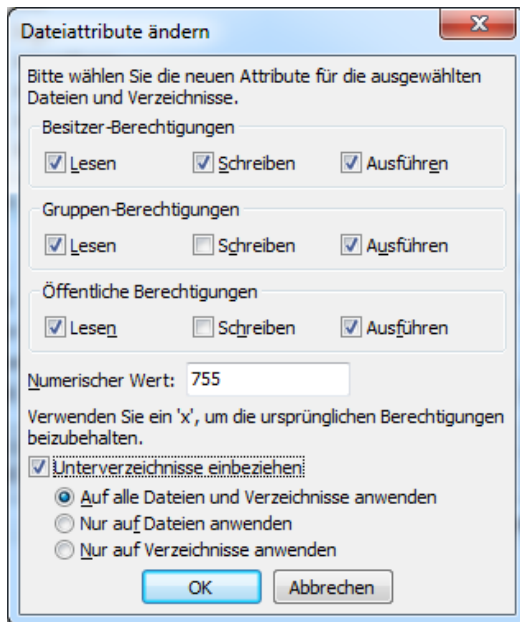


Abbildung 4: Abbildung, setzen der Datei-Berechtigungen auf dem FTP-Server

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: Anzeige der Core-Module in der Liste der Module.....	5
Abbildung 2: Demo-Modul custom_types_example.....	7
Abbildung 3: Sicherheitsnachfrage, ob ein Datensatz wirklich gelöscht werden soll.....	8
Abbildung 4: Abbildung, setzen der Datei-Berechtigungen auf dem FTP-Server.....	12