



Neue Features in UliCMS 2019.4



Inhaltsverzeichnis

Über dieses Dokument.....	1
Zielgruppe.....	1
Was ist UliCMS?.....	1
Neue Features.....	2
Neue API-Methoden.....	3
Entfernte API-Methoden.....	6
Verbesserungen.....	6
Fehlerkorrekturen.....	6
Entfernte Features.....	6
Sicherheit.....	6
Detaillierte Beschreibung der neuen Features.....	7
Automatische Einbindung von externen Medien.....	7
Download-Quelle von Paketen nun in Infos hinterlegt.....	8
Seitenverwaltung ist nun hierarchisch.....	8
Neues Standardtheme "2020".....	8
Design-Einstellungen.....	9
Vorschaubild vom ausgewählten Theme.....	9
Vorschau der gewählten Schriftart.....	10
Design-Einstellungen können nun in SCSS Code als Variablen verwendet werden.....	10
controller_function_permissions unterstützt nun Wildcards.....	11
Sprachen können wieder gelöscht werden.....	13
Username beim Login ist nun Case-Insensitive.....	13
Durchführung der Installation.....	14
Durchführung des Upgrades.....	15
Benötigen Sie Hilfe?.....	16
Abbildungsverzeichnis.....	16

Über dieses Dokument

Dieses Dokument beschreibt die neuen Features in UliCMS 2019.4.

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

- Automatische Einbindung von externen Medien
- Download-Quelle von Paketen nun in Infos hinterlegt
- Seitenverwaltung ist nun hierarchisch
- Neues Standardtheme "2020"
- Design-Einstellungen: Vorschaubild vom ausgewählten Theme
- Design-Einstellungen: Vorschau der gewählten Schriftart
- `controller_function_permissions` unterstützt nun Wildcards
- Sprachen können wieder gelöscht werden
- Design-Einstellungen können nun in SCSS Code als Variablen verwendet werden
- Username beim Login ist nun Case-Insensitive

Neue API-Methoden

- `ContentFactory::getAllByType($type, $order = "id")`
- `ContentFactory::getAllByParent($parent_id, $order = "id")`
- `Content->getChildren($order)`
- `Content->hasChildren()`
- `Page->getHistory()`
- `Page->getParent()`
- `Page->makeFrontPage()`
- `Page->makeErrorPage403(bool $enabled = true): void`
- `Page->makeErrorPage404(bool $enabled = true): void`
- `Page->isFrontPage()`
- `Page->isErrorPage403(): bool`
- `Page->isErrorPage404(): bool`
- `Page->isErrorPage(): bool`
- `Page->getDeletedAt()`
- `Audio_Page->getAudio(): ?Audio`
- `Audio_Page->setAudio(?Audio $audio): void`
- `Video_Page->getVideo(): ?Video`
- `Video_Page->setVideo(?Video $video): void`
- `Video::getAll(string $order = "id"): array`
- `Audio::getAll(string $order = "id"): array`
- `Video->getWidth(): ?int`
- `Video->getHeight(): ?int`
- `Video->setWidth(?int $val): void`
- `Video->setHeight(?int $val): void`
- `TypeMapper::getModel($type)`
- `File::deleteIfExists($file)`
- `ActionRegistry::getAction($action)`
- `ActionRegistry::getActions()`

- User->checkPassword(\$password)
- User->toSessionData()
- User->registerSession(\$redirect)
- User::fromSessionData()
- Comment::getReadCount()
- Comment::getAllCount()
- Database::createSchema(\$name)
- Model->isPersistent()
- Model->hasChanges()
- Module->isInstalled()
- Module->getShortCode()
- ModuleHelper::getMainClass(string \$module): ?Controller
- StringHelper::isUpperCase(string \$val): bool
- StringHelper::isLowerCase(string \$val): bool
- ArrayHelper::flatten(\$input)
- PackageManager->isInstalled(\$package, \$type = PackageTypes::TYPE_MODULE)
- UliCMS\HTML\icon(\$classes, \$htmlAttributes = array())
- UliCMS\HTML|stringContainsHtml(\$string)
- UliCMS\HTML\link(\$url, \$text, \$allowHtml = false, \$target = null, \$htmlAttributes = array())
- UliCMS\HTML\buttonLink(\$url, \$text, \$type = null, \$allowHtml = false, \$target = null, \$htmlAttributes = array())
- UliCMS\HTML\Button::button(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)
- UliCMS\HTML\Button::default(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)
- UliCMS\HTML\Button::primary(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)
- UliCMS\HTML\Button::info(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)
- UliCMS\HTML\Button::warning(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)

- UliCMS\HTML\Button::danger(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)
- UliCMS\HTML\Button::link(\$text, \$type = ButtonType::BUTTON_SUBMIT, \$htmlAttributes = [], \$allowHtml = false)
- NumberFormatHelper::timestampToHtml5Datetime(\$timestamp = null)
- is_today(\$datetime = null)
- is_tomorrow(\$datetime = null)
- is_yesterday(\$datetime = null)
- is_past(\$datetime = null)
- is_future(\$datetime = null)
- get_carbon(\$datetime = null)
- _t(string \$name, array \$placeholders = [])
- t(string \$name, array \$placeholders = [])
- singularOrPlural(int \$count, string \$singular, string \$plural)
- is_zero(\$number)
- midnight(\$datetime = null)
- midday(\$datetime = null)
- end_of_day(\$datetime = null)
- is_midnight(\$datetime = null)
- is_end_of_day(\$datetime = null)
- is_midday(\$datetime = null)
- is_zero(\$val)
- is_decimal(\$val): bool
- stringContainsShortCodes(string \$content, ?string \$module = null): bool

Entfernte API-Methoden

- addUser()
- PackageManager->replacePlaceholders(\$url)

Verbesserungen

- Ein- und Ausblenden der Filter in der Seitenliste ist nun schneller
- Auf iOS Geräten wird nun für alle scrollbaren Bereiche impulsbasiertes Scrolling verwendet

Fehlerkorrekturen

- datetime-local Eingabefelder auf iOS Safari sind immer ungültig
- Das Speichern des States der DataTables ist fehlerhaft
- Einige Stellen an denen Texte hardgecodet waren sind nun korrekt lokalisiert
- render() Methode eines Modules wird nun nur noch aufgerufen, wenn ein HTML-String dessen Shortcode enthält

Entfernte Features

- Unterstützung für veraltete unsichere Passwortverschlüsselung entfernt
- Unbenutztes Feld "Homepage" in den Metadaten eines Artikels entfernt
- Feld "Open Graph - Typ" entfernt

Sicherheit

- Fix für XSS-Sicherheitslücke CVE-2019-11398 8 (<https://nvd.nist.gov/vuln/detail/CVE-2019-11398>)

Detaillierte Beschreibung der neuen Features

Automatische Einbindung von externen Medien

URLs von externen Medien werden bei Aufruf einer Seite automatisch durch eingebettete Media Player ersetzt. Es werden über 150 Dienste unterstützt.

Unter den unterstützten Diensten befinden sich unter anderem die Folgenden:

- YouTube
- Dailymotion
- MyVideo
- Vimeo
- Ustream

Kopieren Sie einfach die URL des einzubindenden Videos und fügen Sie dieses im Editor ein. Der Editor fügt die URL ein und verlinkt diese automatisch. Speichern Sie nun die Seite und rufen Sie diese auf. Die URL wird automatisch durch einen Player ersetzt.

Download-Quelle von Paketen nun in Infos hinterlegt

Bei Modulen und Designs, die aus dem eXtend Portal bezogen wurden, wird nun im Info-Dialog die Downloadseite verlinkt.

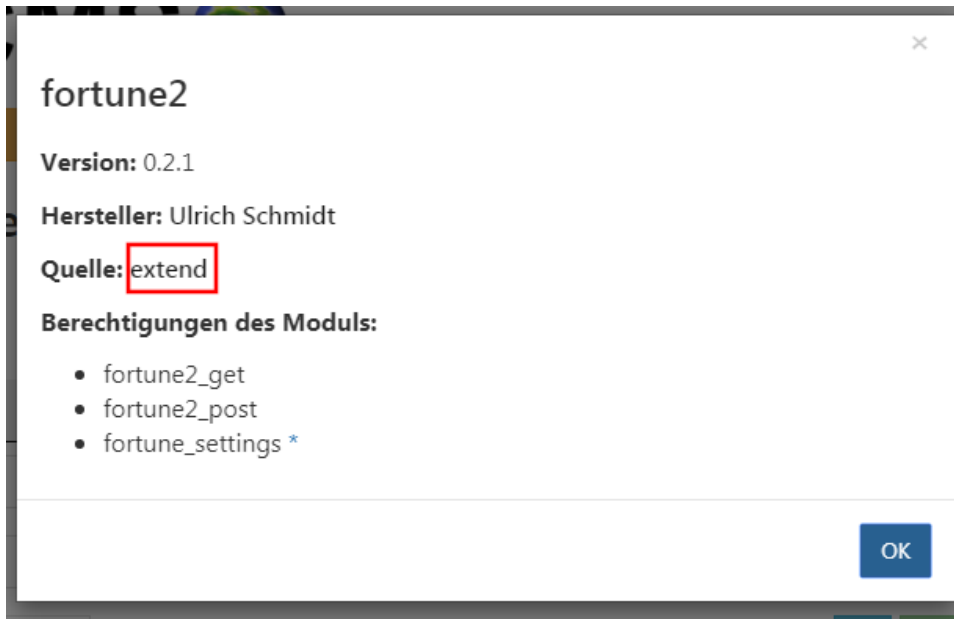


Abbildung 1: Link zur Download-Seite eines Pakets

Seitenverwaltung ist nun hierarchisch

[...]

Neues Standardtheme "2020"

[...]

Design-Einstellungen

Vorschaubild vom ausgewählten Theme

Es wird nun eine Vorschau des ausgewählten Themes angezeigt.

Wenn ein Theme ausgewählt wird, wird ein Screenshot des Themes angezeigt.

Der Screenshot muss sich im Hauptordner eines Themes unter dem Dateinamen screenshot.jpg, screenshot.png oder screenshot.gif befinden, damit es angezeigt wird.



Abbildung 2: Vorschau eines Themes

Vorschau der gewählten Schriftart

Es wird nun eine Vorschau der ausgewählten Schriftart angezeigt.

Schriftart	Comic Sans MS ▼
Schriftgröße	16px ▼
Franz jagt im komplett verwahrlosten Taxi quer durch Bayern	

Abbildung 3: Vorschau der gewählten Schriftart

Design-Einstellungen können nun in SCSS Code als Variablen verwendet werden

Die Design-Einstellungen sind nun als SCSS Variablen verfügbar.

Diese werden in der Datei **ULICMS_ROOT/content/generated/design_variables.scss**, welche automatisch generiert wird, definiert.

Hier ein Beispiel für den Inhalt der Datei.

```
/*
    This file is autogenerated
    DON'T EDIT THIS FILE BECAUSE ALL CHANGES ARE GETTING OVERWRITTEN
    If you want to customize these values
    change it at the Design Settings User Interface.
*/

$header-background-color: #35A1E8;
$body-text-color: #153154;
$body-background-color: #FCFCFC;
$default-font: Arial, 'Helvetica Neue', Helvetica, sans-serif;
$font-size: 16px;
```

controller_function_permissions unterstützt nun Wildcards

controller_function_permissions unterstützt nun Wildcard Permissions.

Dies bedeutet, dass alle Methoden eines Controllers eine bestimmte Berechtigung voraussetzen können, ohne jede Methode einzeln in der metadata.json des Moduls eintragen zu müssen.

Geben Sie hierfür als Identifier für die Controller Action NameDesControllers::* an.

Es kann zusätzlich für einzelne Methoden eine abweichende Berechtigung definiert werden.

Sofern für die aufgerufene Methode keine Berechtigung explizit definiert ist, wird dann die Berechtigung im Eintrag NameDesControllers::* vorausgesetzt.

Hier ein Beispiel.

Ihr Controller sieht wie folgt aus

```
class MyController extends Controller{
    public function fooBar(){
        // do something...
    }

    public function sayHello(){
        // do something...
    }

    public function calculatePi(){
        // do something...
    }
}
```

Die `controller_function_permissions` Sektion in der Metadaten Datei des Moduls sieht wie folgt aus.

```
{  
  ...  
  "controller_function_permissions" : {  
    "MyController::*" : "my_permission",  
    "MyController::calculatePi" : "calculate_pi"  
  }  
}
```



Dies ergibt folgende Berechtigungstabelle:

Methode	Benötigte Berechtigung
MyController::fooBar()	my_permission
MyController::sayHello()	my_permission
MyController::calculatePi()	Calculate_pi

Sprachen können wieder gelöscht werden

Es ist nun wieder möglich Sprachen zu löschen.

2 Datensätze gefunden.

Kürzel ▲	Volle Bezeichnung ▼	Standard	
de	Deutsch	✓	
en	English	Als Standard festlegen	

1 bis 2 von 2 Einträgen

Zurück

1

Nächste

Abbildung 4: Löschen von Sprachen ist wieder möglich

Username beim Login ist nun Case-Insensitive

Bei der Anmeldung im UliCMS Backend wird beim Benutzernamen nun nicht mehr zwischen Groß- und Kleinschreibung unterschieden.

Somit ist sowohl die Anmeldung als „max.muster“ als auch als „Max.Muster“ möglich.

Bei Passwörtern wird weiterhin zwischen der Schreibweise unterschieden.

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 des Upgrades

[...]

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

Abbildungsverzeichnis

Abbildung 1: Link zur Download-Seite eines Pakets.....	8
Abbildung 2: Vorschau eines Themes.....	9
Abbildung 3: Vorschau der gewählten Schriftart.....	10
Abbildung 4: Löschen von Sprachen ist wieder möglich.....	13