Публикация на тему

Установка CKEditor на Laravel

Расширенный текстовый редактор необходим для написания расширенного контента для ваших страниц или статей. На веб-странице нам нужны разные элементы, такие как изображения, абзацы, ссылки и т. Д. Все эти разные элементы можно добавить с помощью CKEditor.

Анотация

-

-

Автор

Михалькевич Александр Викторович

Публикация

Наименование Установка CKEditor на Laravel

Автор А.В.Михалькевич

Специальность Расширенный текстовый редактор необходим для написания расширенного контента для ваших страниц или статей. На веб-странице нам нужны разные элементы, такие как изображения, абзацы, ссылки и т. Д. Все эти разные элементы можно добавить с помощью CKEditor.,

Анотация

-

Anotation in English

-

Ключевые слова CKEditor, Laravel, PHP, редкатор кода, загрузка изображений, загрузчик изображений

Количество символов 6794

Содержание

Введение
1 Почему CKEditor?
2 Установка CKEditor
3 Подготовка html-элемента
4 Подключение CKEditor в blade-шаблонах laravel

5 Опция загрузки изображения
6 Создание символической ссылки
7 Экшн контроллера загрузчика изображений для серверов с поддержкой линкования
8 Альтернативные экшны контроллера CKEditorController
Заключение
Список использованных источников
Приложения

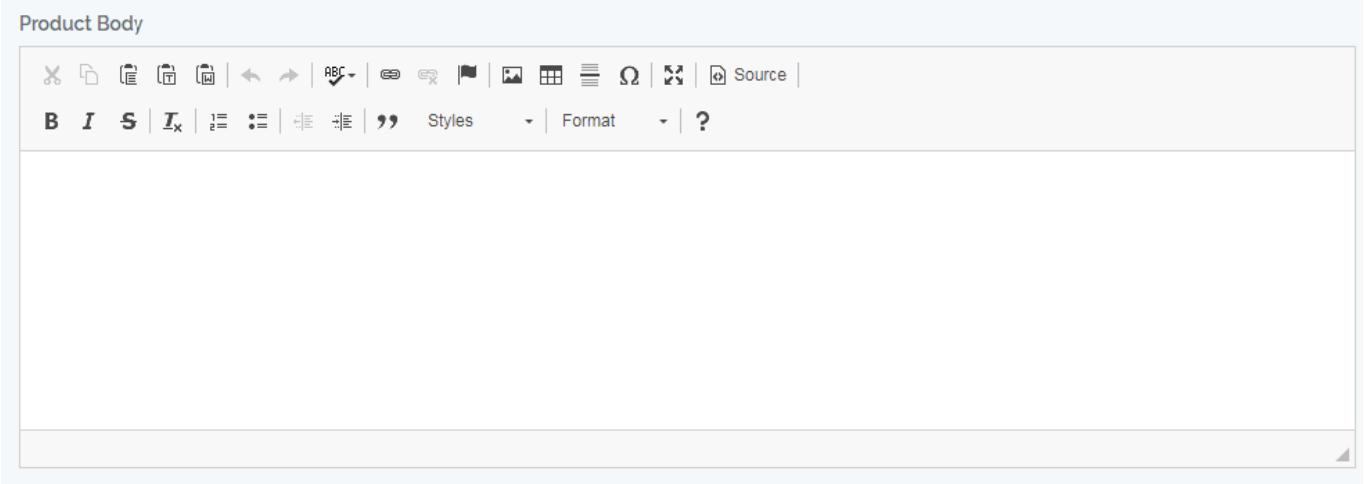
Введение

Устанавливаем CKEditor с возможностью загрузки изображений

1 Почему CKEditor?

CKEditor - это JavaScript библиотека, которая превращает обычный html-элемент, будь то textarea, div, p или любой другой в великолепный редактор кода с возможностью загрузки изображений. Конечно, существует множество других плагинов, предлагающих подобные решения, однако на фоне их CKEditor выделяется своей простотой, актуальностью решений, списком поддерживоемых фрэймворков, сообществом поддержки, и т.д.

Финальный CKEditor выглдяти так:



2 Установка CKEditor

Существует 2 способа установки CKEditor в пакете Laravel - CDN или загрузка пакета Вы можете получить оба ресурса на их странице загрузки - https://ckeditor.com/ckeditor-4/download.

3 Подготовка html-элемента

Для элемента textarea добавляем уникальный идентификатор. Это может быть id или class. Еще этот элемент должен содержать атрибут name.

```
textarea name="body" id="body"
```

4 Подключение CKEditor в blade-шаблонах laravel

Подключение скриптов осуществляется в blade-файлах.

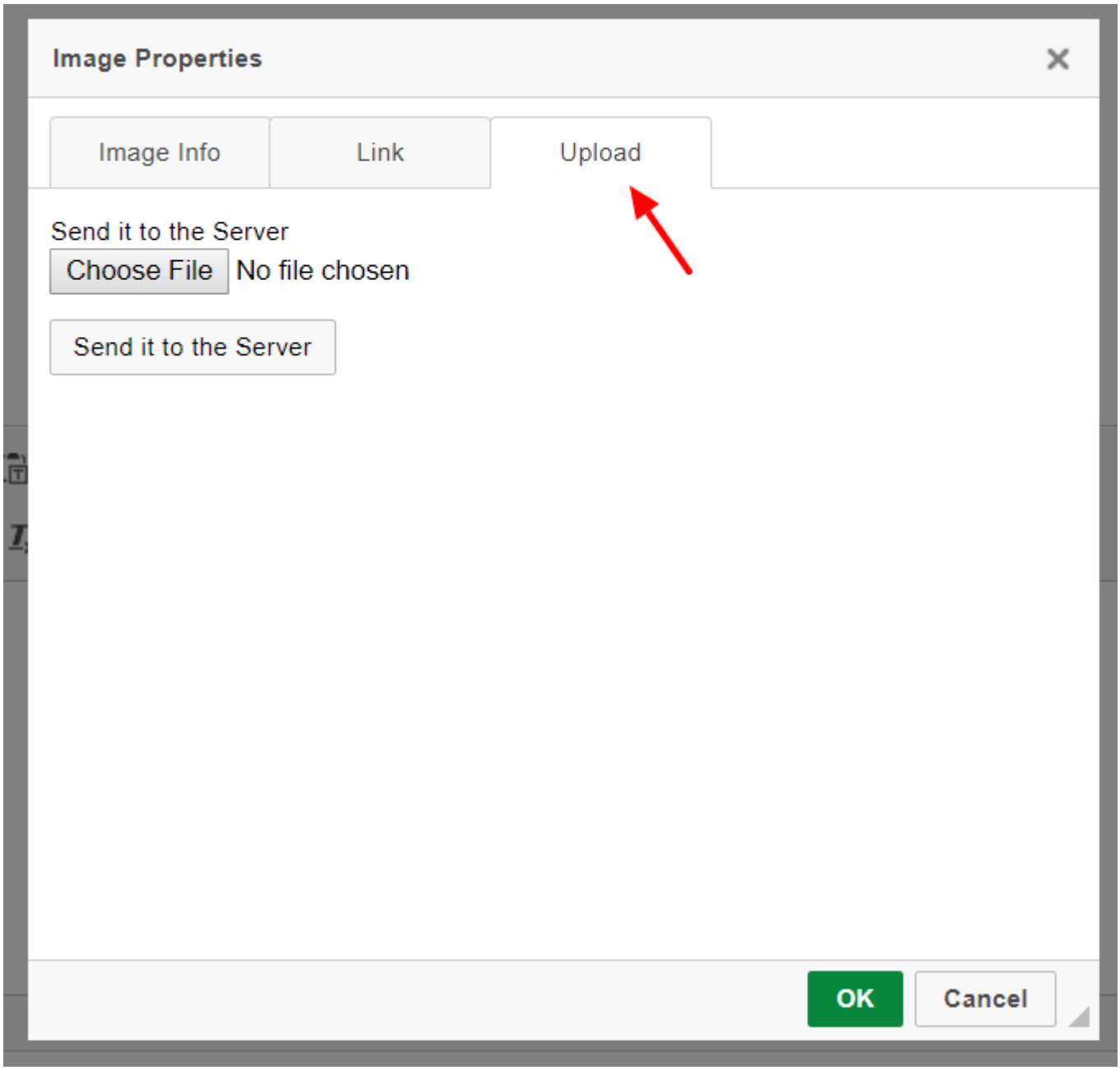
```
<script src="{{ asset('ckeditor/ckeditor.js') }}"></script>
```

После чего пишем следующий скрипт:

```
CKEDITOR.replace( 'body', {
    filebrowserUploadUrl: "{{route('upload', ['_token' => csrf_token() ])}}",
    filebrowserUploadMethod: 'form'
});
```

5 Опция загрузки изображения

Перезагрузите страницу. Теперь, если вы нажмете на значок изображения CKEditor, вы увидите опцию загрузки изображения, как показано ниже.



6 Создание символической ссылки

Для хранения изображения на сервере будем использовать функциональные возможности Laravel. Итак, сначала создаем символическую ссылку на папку «storage». Запустите команду ниже, чтобы создать символическую ссылку:

```
php artisan storage:link
```

7 Экшн контроллера загрузчика изображений для серверов с поддержкой линкования

Сперва создадим маршут в файле routes/web.php

```
Routes::post('upload','CKEditorController@upload')->name('upload');
```

Далее создадим контроллер CKEditorController:

```
php artisan make:controller CKEditorController
```

Переходим в него, и добавляем экшн upload:

```
public function upload(Request $request)
    {
        if($request->hasFile('upload')) {
            if(!Auth::guest()){
                $puth_url = $_SERVER['DOCUMENT_ROOT'].'/storage/uploads/' .
Auth::user()->id;
                if (!file_exists($puth_url)) {
                    mkdir($puth_url, 0777, true);
                }
            }else{

            }
            //get filename with extension
            $filenamewithextension =
$request->file('upload')->getClientOriginalName();

            //get filename without extension
            $filename = pathinfo($filenamewithextension, PATHINFO_FILENAME);

            //get file extension
            $extension =
$request->file('upload')->getClientOriginalExtension();

            //filename to store
            $filenametostore = $filename.'_'.time().'.'.$extension;

            //Upload File
$request->file('upload')->storeAs('public/uploads/'.Auth::user()->id,
```

```
$filenametostore);

            $CKEditorFuncNum = $request->input('CKEditorFuncNum');
            $url =
asset('storage/uploads/'.Auth::user()->id.'/'.$filenametostore);
            $msg = 'Image successfully uploaded';
            $re =
"<script>window.parent.CKEDITOR.tools.callFunction($CKEditorFuncNum, '$url',
'$msg')</script>";

            // Render HTML output
            @header('Content-type: text/html; charset=utf-8');
            echo $re;
        }
    }
```

8 Альтернативные экшны контроллера CKEditorController

Если у вас нет возможности создать символьную ссылку, используйте этот экшн:

```
public function upload_without_storage(Request $request){
        if($request->hasFile('upload')) {
            if (!Auth::guest()) {
                $puth_url = $_SERVER['DOCUMENT_ROOT'] .
'/public/user_uploads/' . Auth::user()->id;
                if (!file_exists($puth_url)) {
                    mkdir($puth_url, 0777, true);
                }
                $path = $request->upload->path();
                $filenamewithextension =
$request->file('upload')->getClientOriginalName();

                //get filename without extension
                $filename = pathinfo($filenamewithextension,
PATHINFO_FILENAME);

                //get file extension
                $extension =
$request->file('upload')->getClientOriginalExtension();

                //filename to store
                $filenametostore = $filename.'_'.time().'.'.$extension;

                move_uploaded_file($path, "$puth_url/$filenametostore");

                $CKEditorFuncNum = $request->input('CKEditorFuncNum');
                $url =
asset('/user_uploads/'.Auth::user()->id.'/'.$filenametostore);
```

```
                $msg = 'Image successfully uploaded';
                $re =
"<script>window.parent.CKEDITOR.tools.callFunction($CKEditorFuncNum, '$url',
'$msg')</script>";

                // Render HTML output
                @header('Content-type: text/html; charset=utf-8');
                echo $re;
            }

        }

    }
```

Заключение

Список использованных источников

Приложения