



WEB разработка

# Laravel Работа с файлове

# Съдържание

Качване на файлове  
[ресурси](#)

Валидация при качване на файлове  
[ресурси](#)

# Качване на файлове

# Задача

Добавете функционалност за добавяне на домашни към платформата.

Всеки курсист трябва да може да добавя домашно към всяка лекция. При добавяне на домашно към лекцията се изтрива предишното домашно/ако има качено/ и остава само новокаченото домашно.

\*В списъка с лекции, курсиста вижда името на каченото домашно, ако липсва домашно се вижда текст - “не е добавено домашно” и бутон за добавяне на домашно.

В своя списък домашни, курсиста вижда качените домашни и лекцията, към която принадлежи.

# Checklist 1

## *Checklist 1*

1. БД - таблица - lectures
  - a. id, name, course\_id
2. БД - таблица - homeworks
  - a. Id, filename, user\_id, lecture\_id
3. Lecture Model - hasMany homeworks
4. Homework Model -
  - a. belongsTo user
  - b. belongsTo lecture
5. User model hasMany homeworks
6. HomeworksController - resource

# Checklist 2

## *Checklist 2*

1. Homework routes/middleware specific routes/
2. Lectures index view - link to upload homework - leads to homework create
3. Homeworks views folder
  - a. Index - list of auth user's homeworks + edit + delete button
  - b. Edit

# Особености във формата

```
{!! Form::open(['route' => 'add_new_user', 'files' => true]) !!}
```

```
...
```

```
{!! Form::close() !!}
```

```
//enctype="multipart/form-data"
```

# Поле за качване на файл

```
{!! Form::file('my_photo', array('class'=>'file', 'id'=>'my_photo')) !!}
```

# Записване на файла в ...

Записваме файла на дисковото пространство в следните стъпки:

1. \$filename = \$request->**file('my\_photo')**->**getClientOriginalName()**;  
**file('my\_photo')** - достъп до файла, качен с формата  
**getClientOriginalName()** - името на качения файл
2. \$request->**file('my\_photo')**->**move('temp', \$filename)**;  
**move('temp'...)** - записваме файла в указаната директория, ако не съществува, директорията се създава

# Записване на файла в ...

```
3. $path = public_path()'//temp'.'/'.$filename;  
   //пълния път до файла  
   //в базата данни записваме само $filename
```

Не забравяйте да добавите в съответния модел, разрешение за записване на името на файла в базата данни

```
protected $fillable = ['bio', 'photo_path', 'user_id'];
```

# Записване на файла в ...

В метода за записване/промяна на данни в съответния контролер, добавяме ред за запис на името на качения файл, /взимаме го от оригиналното име или генерираме свое име/

```
$user_profile = Profile::create([
    'photo_path'      => $filename,
    'bio'             => $request['bio'],
    'user_id'         => $user->id
]);
```

# Изтриване на файл ...

Изчистваме директорията, предназначена за качване на файл

Не забравяйте да поставите в началото - **use File;**

```
File::cleanDirectory('temp\\');
```

# Изтриване на файл ...

//информация за новия файл, постъпваща със заявката

```
$file = $request->file('images');
$filename = $file->getClientOriginalName();
$file->move(public_path().'/assets/user_img', $filename);
```

//информация за файла, когото новия файл замества и е логично да изтрием  
//тази информация, също може да постъпва със заявката

```
$file_to_delete = public_path().'/files_folder_name/'.$filename;
File::delete($file_to_delete);
```

Други функции на File - vendor/league/flysystem/src/File

**Достъп до файлове, качени с приложението**

# Отпечатване на снимка

//в profiles е записано името на файла

//генерираме пълния път до снимката във вюто

```
photo_path }}">
```

# Валидация при качване на файлове

# Валидация при ...

```
public function rules()  
{  
    return [  
        'my_photo'=>'required|mimes:jpg,png'  
    ];  
}
```

# Валидация при ...

```
public function messages()  
{  
    return [  
        'my_photo.required' => 'Изберете файл!',  
        'my_photo.mimes' => 'Изберете файл jpg или png!',  
    ];  
}
```