

ŽINIATINKLIO PROGRAMUOTOJO

PHP MODULIS

8 SAVAITĖS
4 PASKAITOS PER SAVAITĘ
PASKAITOS TRUKMĖ: 2 VAL. 15 MIN.
27/32 paskaita

Refleksija

- Artisan
- Migracijos
- Eloquent Models
- Ryšiai
- Blade
- Controllers

Namų darbų aptarimas

Paskaitos tema:

Formų validatorių kūrimas ir taikymas. Failinės
sistemos, failų įkėlimas ir tvarkymas

Kontroleriai (angl. controllers)

- <https://laravel.com/docs/5.6/controllers>
- app\Http\Controllers direktorijoje

Pavyzdys

```
<?php
namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class PageController extends Controller
{
    public function testAction()
    {
        // your code
    }
}
```

Kontroleriai

- Kontrolerio panaudojimas routes faile (routes/web.php):
Route::get('/test', 'PageController@testAction');
 - Čia PageController Kontrolerio pavadinimas, testAction – metodo pavadinimas.
- Artisan komanda sukurti naujam kontrolieriui:
 - php artisan make:controller CarController
- Jei reikia perduoti papildomus parametrus, tuomet turėtumėme sukurti metodą su parametrais ir Route su parametru perdavimu. Pvz.:
Route::get('projects/{id}', 'ProjectController@showProjects');

Kontroleriai

- Taip pat kontrolierius kaip ir migracijas galime sukurti ir kuriant modelius:
- `php artisan make:model Modelis -mcr`
- `-m` sukurti migraciją
- `-c` sukurti kontrolierį
- `-r` modelis bus resurso tipas

Resursai

Mums įprastai reikia įprastinių routų kiekvienam veiksmui:
Peržiūrėti, ištrinti, koreguoti ...

Laravel tam turi sukūrusi route Resource.

Pavyzdžiui:

```
Route::resource('photos', 'PhotoController');
```

Arba keletą resursų iškart:

```
Route::resources([  
    'photos' => 'PhotoController',  
    'posts' => 'PostController'  
]);
```

Resurso Kontroleriai

Taip pat sukuriant Controllerį mes galime nurodyti jog šis kontroleris bus susietas su resursu:

```
php artisan make:controller PhotoController --resource
```

Taip pat galime jį susieti ir su modeliu:

```
php artisan make:controller PhotoController --resource --model=Photo
```

Šie kontroleriai gali turėti ir papildomų laukų.

Užduotis

Sukurkime Kontrolerį – Cars

Perkelkime automobilio atvaizdavimą ir pridėjimą iš route į CarsController.

logiką iš Route failo į CarsController

Perkelkite automobilio redagavimo ir ištrynimo metodus iš Route failo į kontrolerį.

Užduotis

- Sukurkite kontrolerį kaip resursą modeliui Owners ir perkeltkite visą valdymo logiką iš route į kontrolerį

Validatoriai

Validatoriai

<https://laravel.com/docs/5.6/validation>

Visi Controller'iai turi validatorius (paveldimi), todėl jie gali būti iškviečiami:

```
$this->validate();
```

Metodo naudojimas:

```
$this->validate($request, ['laukas'=>'taisykles']);
```

Taisyklės

Tipo:

Numeric, Date, Email ...

Dydžio:

Max, Min, Between ...

DB:

Unique ...

Visos taisyklės:

<https://laravel.com/docs/5.6/validation#available-validation-rules>

Pavyzdys

```
$this->validate($request, [  
    'email' => 'required|email',  
    'age' => 'required|numeric|max:110|min:18',  
    'name' => 'required|unique:projects|max:255',  
]);
```


Užduotis

Sukurkime validatorių pridedamiems automobiliams tikrinti.

Validatoriai turi tikrinti:

1. Numeris – provalomas, unikalus, ne ilgesnis nei 6 simboliai
2. Markė– privalomas, ne ilgesnis nei 64 simboliai
3. Modelis– privalomas, ne ilgesnis nei 64 simboliai

Pranešimų išvedimas

Norint išvesti pranešimus Blade:

```
@if (count($errors) > 0)
    <div class="alert alert-danger">
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
```

Užduotis

- Išveskime pranešimus apie įvedimo klaidą pridedant automobilį.

Pranešimai

```
$messages = [  
    'size' => 'Laukas :attribute turi būti trumpesnis  
nei: :value !'  
];
```

```
$messages = [  
    'email.required' => 'Prašome įvesti el. paštą!',  
];
```

Pranešimų siuntimas:

```
$this->validate($request,  
    'email' => 'required|email',  
    'age' => 'required|numeric|max:110|min:18',  
    $messages);
```

Užduotis

- Padarykime jog pridedant automobilį išvestų pranešimus apie klaidas lietuvių kalba.

Apipavidalinimas

Blaid'e galima panaudoti f-ją kuri išves senas reikšmes

```
{{ Request::old('lauko pavadinimas') }}
```

Norint patikrinti ar laukas turi klaidą galime tai patikrinti f-jos has pagalba:

```
{{ $errors->has('lauko_pavadinimas') }}
```

 - gražins 1 jei yra klaida, 0 jei nėra klaidos

Užduotis

- Padarykime jog pridedant automobilį ir įvykus klaidai būtų išsaugotos įvestos reikšmės, bei paryškinti langeliai kuriuose padarytos klaidos.

Užduotis

- Sukurkite validatorių automobilio savininkų pridėjimui
- Validatorius turi tikrinti visus įvedamus laukus ir išvesti pranešimus
- Perdarykite pranešimus taip, kad jie būtų išvedami būtent prie to laukelio kuriame padaryta klaida
- Sukurkite lietuviškus pranešimus apie įvedimo klaidas, sudarykite pranešimus taip kad būtų išsaugotos buvusios reikšmės, klaidingi laukeliai pažymėti, o pranešimai rodomi prie klaidingų laukelių.

Užduotis

- Tikrinimą perkeltkite iš kontrolių į Requests
- Sukurkite tikrinimus tiek pridėdant tiek ir redaguojant automobilį ir jo savininką.
- Apie Request klasės praplėtimą:
- <https://laravel.com/docs/5.6/validation#form-request-validation>

Formų tikrinimas “Requests”

Naujo Request sukūrimas:

Sukūrimas iš artisan: **php artisan make:request StoreBlogPostRequest**

```
namespace App\Http\Requests;
use App\Http\Requests\Request;

class FormValidator extends Request
{
    public function authorize()
    {
        return false;
    }

    public function rules()
    {
        return [
            // Taisyklės
        ];
    }
}
```

Request'ų panaudojimas Controller'iuose

Norint panaudoti Request'ą kontroleryje mes tiesiog jį nurodysime parametrų sąraše.

```
public function display(FormValidator $request)  
{  
    return view('home');  
}
```

Failų įkėlimas

<https://laravel.com/docs/5.6/filesystem>

Vartotojų failai (viešieji)

Vieši failai turėtų būti talpinami:

`/storage/app/public`

Konfigūracinis failinės sistemos failas:

`/config/filesystems/`

Tačiau norint pasiekti šiuos failus reiktų sukurti simbolinę nuorodą:
`php artisan storage:link`

Failo atvaizdavimas

- Kai failas įkeliamas į katalogą `/storage/app/public/`
- Failas pasiekiamas nuoroda `[URL]/storage/[failo_pavadinimas]`
- Arba nuoroda gali būti sugeneruojama:
`{{ asset('storage/[failas]') }}`

Failų įkėlimas

Norint sukurti failo įkėlimą, pirmiausia turėtume sukurti formą.

```
<form action="" method="post" enctype="multipart/form-data">  
  {{ csrf_field() }}  
  <input type="file" name="failas">  
  <input type="submit" value="Išsiusti">  
</form>
```

Failų įkėlimo vykdymas

Visi atsiųsti failai pasiekiami:

```
$request->file('failo pavadinimas');
```

F-ja grąžins failo objektą.

```
request()->file('failas')->store('test');
```


Užduotis

- Pamėginkime sukurti formą kurios pagalba galėtumėme įkelti failą ir tas failas atsivaizduotų puslapyje.

Užduotis 2

Galerijos sukūrimas. Sukurkite migraciją ir modelį:

Lentelė: gallery

id (auto increment)

name (string, 128)

....

Pamėginkite padaryti kad visi įkeliami failai taip pat pasitalpintų ir į šią lentelę.

Failų atvaizdavimas

Route taip pat gali gražinti failo informaciją ir patį failą vartotojui:

```
return response()->download($pathToFile);
```

```
return response()->download($pathToFile, $name, $headers);
```

```
return response()->download($pathToFile)->deleteFileAfterSend(true);
```

Taip pat ir išvedimo:

```
return response()->file($pathToFile);
```

```
return response()->file($pathToFile, $headers);
```

Kelias iki failo gali būti paimamas:

```
storage_path('app/file.txt');
```

Route kaip failų pavadinimai

- Mes galime Laravel sistmoje susikurti atskirus route kurie būtų skirti failų paėmimui. Pvz.:

```
Route::get('/failas/{file}',function(){  
    // failo gražinimas  
});
```

- Tuomet route galime apsaugoti su prisijungimo ar kitokiais mechanizmais

Užduotis

- Sukurkite route kurie atvaizduoja failus (juos išveda)
- Padarykite taip, kad failą galėtų matyti tik prisijungę vartotojai

Santrauka

- Kontroleriai (angl. Controllers)
- Formų validatoriai
- Requests
- Failinė sistema