



Technologie www

Cvičení 11 – A4TEW

Radek Vala a Milan Oulehla

Laravel – projekt autosalón - rozšíření

Na dnešním cvičení budeme pokračovat na projektu autosalón z minulého cvičení:

`/Applications/MAMP/htdocs/jmeno_studenta/autosalon`

Nejprve je nutné stávající projekt upravit:

1-Laravel_uprava_projektu_autosalon.txt

This single route declaration creates multiple routes to handle a variety of actions on the resource:

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

Actions Handled By Resource Controller

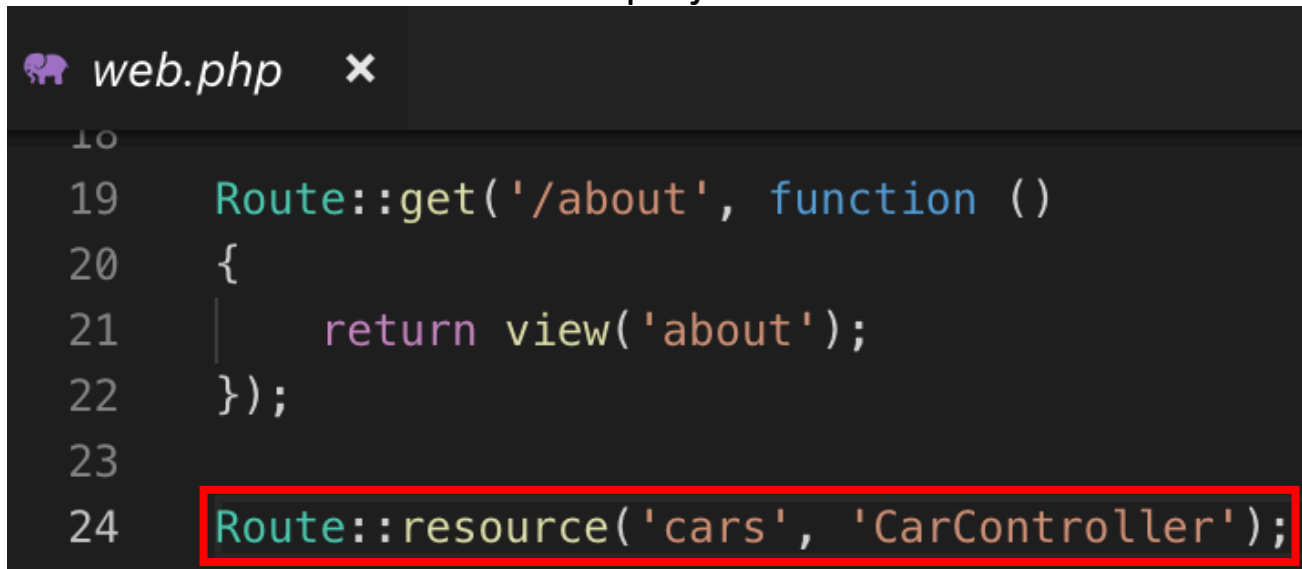
Verb	URI	Action	Route Name
GET	/photos	index	photos.index
GET	/photos/create	create	photos.create
POST	/photos	store	photos.store
GET	/photos/{photo}	show	photos.show
GET	/photos/{photo}/edit	edit	photos.edit
PUT/PATCH	/photos/{photo}	update	photos.update
DELETE	/photos/{photo}	destroy	photos.destroy

Route web.php v našem projektu

Dokumentace:

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

Náš projekt:



```
web.php x
10
19 Route::get('/about', function ()
20 {
21     return view('about');
22 });
23
24 Route::resource('cars', 'CarController');
```

Tzn., že můžeme využívat všechny URIs a Route Names z předchozího slajdu, jen místo **photos** bude použito klíčové slovo **cars**!

Celkové schéma:

http://localhost:8888/jmeno_studenta/autosalon/public/cars

web.php x

24 → `Route::resource('cars', 'CarController');`

GET	/photos	index	photos.index
GET	/photos/create	create	photos.create
POST	/photos	store	photos.store

OPEN EDITORS

index.blade.php resources...

x CarController.php app/Http...

AUTOSALON

app

Console

Exceptions

Http

Controllers

Auth

CarController.php

Controller.php

Middleware

Kernel.php

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class CarController extends Controller
8 {
9     /**
10      * Display a listing of the resource.
11      *
12      * @return \Illuminate\Http\Response
13      */
14     public function index()
15     {
16         return view('cars.index');
17     }
18 }
```

AUTOSALON

app

bootstrap

config

database

public

resources

js

lang

sass

views

cars

index.blade.php

welcome.blade.php

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

navbar.blade.php x

```
1 <nav class="navbar navbar-expand-sm navbar-light bg-light">
2   <a class="navbar-brand" href="#">Admin - autosalon</a>
3   <button class="navbar-toggler d-lg-none" type="button" data-toggle="collapse" data-target=
4     aria-expanded="false" aria-label="Toggle navigation">
5     <span class="navbar-toggler-icon"></span>
6   </button>
7   <div class="collapse navbar-collapse" id="collapsibleNavId">
8     <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
9       <li class="nav-item active">
10        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
11      </li>
12
13      <li class="nav-item dropdown">
14        <a class="nav-link dropdown-toggle" href="#" id="dropdownId" data-toggle="drop
15        <div class="dropdown-menu" aria-labelledby="dropdownId">
16          <a class="dropdown-item" href="{{ route('cars.index') }}">List all</a>
17          <a class="dropdown-item" href="{{ route('cars.create') }}">Create new</a>
18        </div>
19      </li>
20    </ul>
21    <form class="form-inline my-2 my-lg-0">
22      <input class="form-control mr-sm-2" type="text" placeholder="Search">
23      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
24    </form>
25  </div>
26 </nav>
```

adminLayout.blade.php x

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <title>Administrace</title>
5     <!-- Required meta tags -->
6     <meta charset="utf-8">
7     <meta name="viewport" content="w
8
9     <!-- Bootstrap CSS -->
10    <link rel="stylesheet" href="ht
11  </head>
12  <body>
13    <!-- navigace -->
14    @include('includes.navbar')
15
16    <!-- Optional JavaScript -->
17    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
18    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X-
19    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
20    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" in
21  </body>
22 </html>
```

index.blade.php x

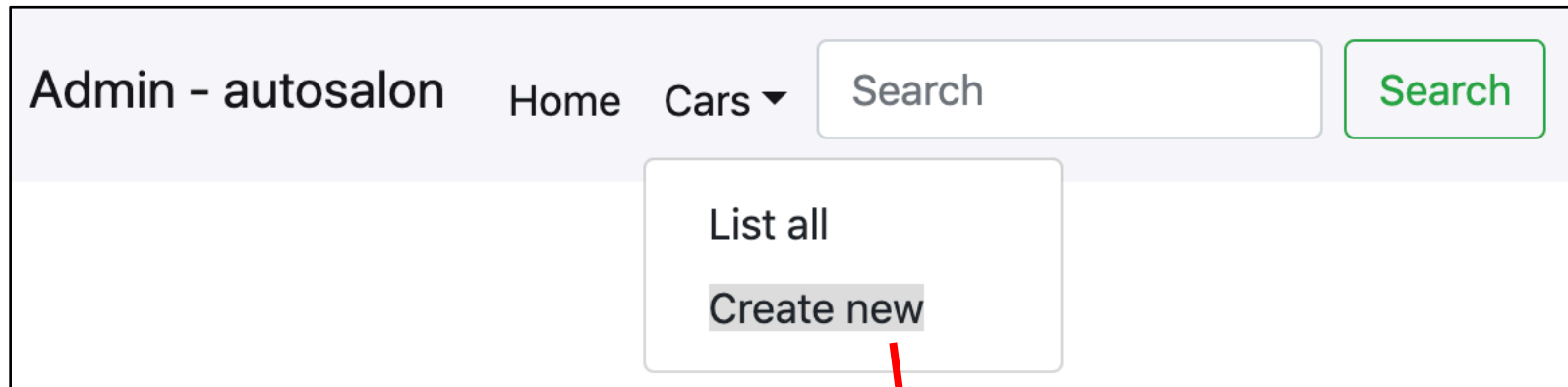
```
1 @extends('layouts.adminLayout')
```

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

Laravel – projekt autosalón - rozšíření



```
▼<div class="dropdown-menu show" aria-labelledby=
"dropdownId">
  <a class="dropdown-item" href="http://localhost:8888/
  oulehla/autosalon/public/cars">List all</a>
  <a class="dropdown-item" href="http://localhost:8888/
  oulehla/autosalon/public/cars/create">Create new</a> == $0
</div>
```

Routes

Routes můžeme odvodit z této tabulky (photos změníme na cars):

Actions Handled By Resource Controller			
Verb	URI	Action	Route Name
GET	/photos	index	photos.index
GET	/photos/create	create	photos.create

`List all`
`http://localhost:8888/jmeno_studenta/autosalon/public/cars`

`Create new`
`http://localhost:8888/jmeno_studenta/autosalon/public/cars/create`

```
<a class="dropdown-item" href="{{ route('cars.create') }}">Create new</a>
```

http://localhost:8888/jmeno_studenta/autosalon/public/cars/create

Routovací systém pomocí web.php: `Route::resource('cars', 'CarController');`
zajistí, že se v `CarController.php` zavolá metoda **create**.

```
CarController.php:  
public function create()  
{  
    return view('cars.create');  
}
```

Musíme vytvořit: `resources/views/cars/create.blade.php`

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

```

adminLayout.blade.php x
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <title>Administrace</title>
5 <!-- Required meta tags -->
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8
9 <!-- Bootstrap CSS -->
10 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
11 </head>
12 <body>
13 <!-- navigace -->
14 @include('includes.navbar')
15
16 <!-- Optional JavaScript -->
17 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
18 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
19 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
20 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
21 </body>
22 </html>

```

```

create.blade.php x
1 @extends('layouts.adminLayout')

```

```

navbar.blade.php x
1 <nav class="navbar navbar-expand-sm navbar-light">
2 <a class="navbar-brand" href="#">Admin
3 <button class="navbar-toggler d-lg-none" type="button"
4     aria-expanded="false" aria-label="Toggle navigation">
5     <span class="navbar-toggler-icon"></span>
6 </button>
7 <div class="collapse navbar-collapse">
8     <ul class="navbar-nav mr-auto">
9         <li class="nav-item active">
10             <a class="nav-link" href="#">Home
11         </li>
12
13         <li class="nav-item dropdown">
14             <a class="nav-link dropdown-toggle" href="#">
15                 <div class="dropdown-menu">
16                     <a class="dropdown-item" href="#">
17                         <a class="dropdown-item" href="#">
18                     </div>
19                 </li>
20             </ul>
21         <form class="form-inline my-2 my-lg-0">
22             <input class="form-control mr-sm-2" type="text">
23             <button class="btn btn-outline-success" type="submit">Search
24         </form>
25     </div>
26 </nav>

```

Laravel – projekt autosalón - rozšíření

V create.blade.php budeme vytvářet formulář.

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

Actions Handled By Resource Controller

Verb	URI	Action	Route Name
GET	/photos	index	photos.index
GET	/photos/create	create	photos.create
POST	/photos	store	photos.store

{{ route('cars.store') }}

=> náš formulář, který je v resources/views/cars/create.blade.php

musí využít metodu **POST**:

```
create.blade.php x
1 @extends('layouts.adminLayout')
2
3 @section('content')
4 <div class="container">
5   <form action="{{ route('cars.store') }}" method="post">
6   @csrf
```

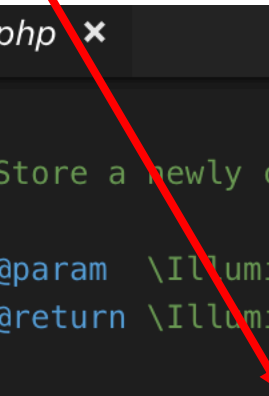
▼<form action="http://localhost:8888/oulehla/autosalon/public/cars" method="post">

POST	/photos	store	photos.store
------	----------------	-------	---------------------

Náš formulář v **create.blade.php** posílá data proti URL adrese **/cars**, aby si to odchytila metoda controlleru **CarController.php** s názvem **cars.store** (cars.store se zavolá, protože se jedná o POST požadavek, GET by zavolal cars.index).

```
create.blade.php ×  
1  @extends('layouts.adminLayout')  
2  
3  @section('content')  
4  <div class="container">  
5      <form action="{{ route('cars.store') }}" method="post">  
6          @csrf
```

```
CarController.php ×  
28  
29  /**  
30   * Store a newly created resource in storage.  
31   *  
32   * @param  \Illuminate\Http\Request  $request  
33   * @return \Illuminate\Http\Response  
34   */  
35  public function store(Request $request)  
36  {  
37      //  
38  }
```



Formuláře – bezpečnostní problém

Může web server věřit datům, která přicházejí a tvrdí, že jsou z formuláře odpovídající stránky?

Odpovědí je: CSRF Protection

<https://laravel.com/docs/5.8/csrf>

Formuláře – bezpečnostní problém

```
create.blade.php x
1  @extends('layouts.adminLayout')
2
3  @section('content')
4  <div class="container">
5      <form action="{{ route('cars.store') }}" method="post">
6          @csrf
```

Do formuláře je potřeba přidat `@csrf`, což zajistí, že formulář bude mít token, který si může serverová strana kontrolovat, zda souhlasí.

!!! Bez `@csrf` nebude formulář fungovat !!!

Formuláře – input typu date

```
<div class="form-group">  
  <label for="produced_on">Produced on</label>  
  <input type="date"  
    class="form-control" name="produced_on" id="produced_on" aria-describedby="helpId" placeholder="">  
  <small id="helpId" class="form-text text-muted">Production date...</small>  
</div>
```

Chrome OK:

Produced on

duben 2019 ▾

po	út	st	čt	pá	so	ne
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Safari KO:

Produced on

Production date...

Submit

Administrace

localhost:8888/oulehla/autosalon/public/cars/create

Admin - autosalon Home Cars ▾

Create new car

Make

Car producer...

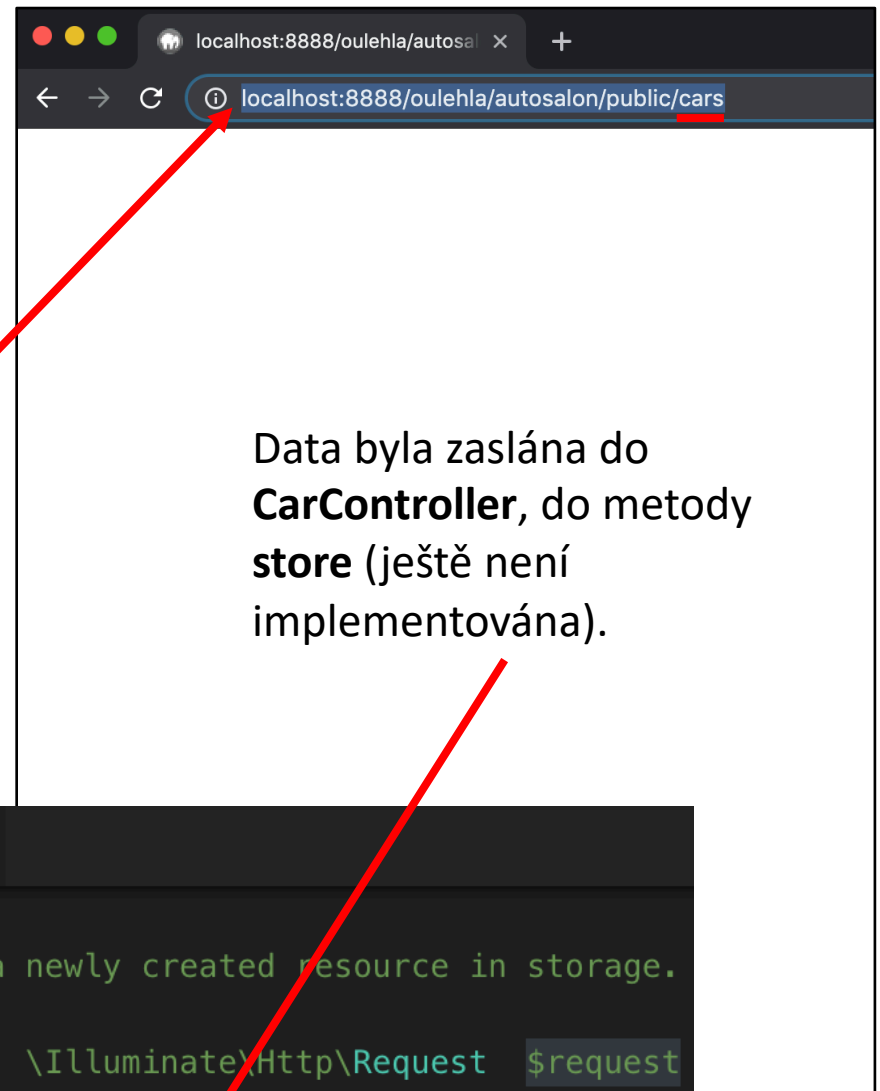
Model

Car model...

Produced on

Production date...

Submit



CarController.php

```
29  /**
30   * Store a newly created resource in storage.
31   *
32   * @param \Illuminate\Http\Request $request
33   * @return \Illuminate\Http\Response
34   */
35  public function store(Request $request)
36  {
37      //
38  }
```

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

create.blade.php:

```
<h1 class="my-5">Create new car</h1>

<form action="{{ route('cars.store') }}" method="post">
    @csrf
```

@csrf zajistí vložení tokenu, type je hidden

http://localhost:8888/jmeno_studenta/autosalon/public/cars/create

```
<h1 class="my-5">Create new car</h1>
▼<form action="http://localhost:8888/oulehla/autosalon/public/cars"
method="post">
    <input type="hidden" name="_token" value=
    "6EwWKDUKHFfnAo7g78Af0T3c4VjhuBNfHHnf9eUjk"> == $0
```

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

Validace formulářových dat na straně serveru

Na stránce Validation:

<https://laravel.com/docs/5.8/validation>

Je podkapitola: **Writing The Validation Logic**,
obsahující kód:

```
public function store(Request $request)
{
    $validatedData = $request->validate([
        'title' => 'required|unique:posts|max:255',
        'body' => 'required',
    ]);

    // The blog post is valid...
}
```


Validace formulářových dat na straně serveru

```
public function store(Request $request)
{
    $validatedData = $request->validate([
        'title' => 'required|unique:posts|max:255',
        'body' => 'required',
    ]);

    // The blog post is valid...
}
```

Do metody store přichází parametr, který je pojmenován `$request` a v němž je tzv. Request objekt.

Request objekt obsahuje http hlavičky ale především **formulářová data**.

Request objekt má metodu **validate**.

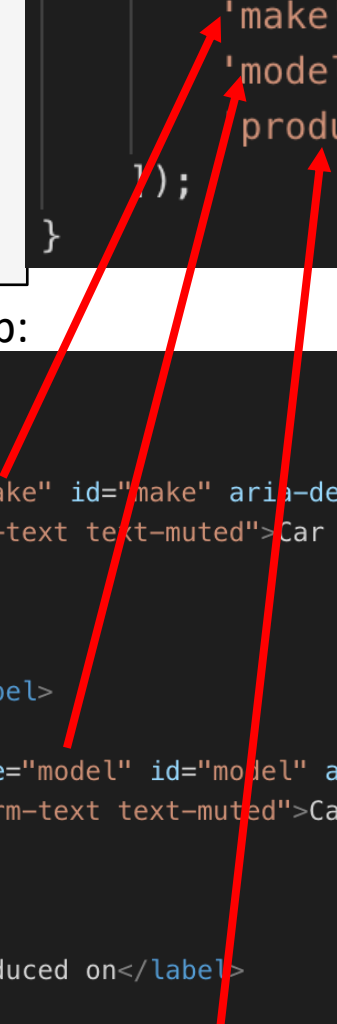
Dokumentace:

```
public function store(Request $request)
{
    $validatedData = $request->validate([
        'title' => 'required|unique:posts|max:255',
        'body' => 'required',
    ]);

    // The blog post is valid...
}
```

CarController.php - metodu **validate** upravíme:

```
public function store(Request $request)
{
    // server side validation
    $validatedData = $request->validate([
        'make' => 'required|max:255',
        'model' => 'required|max:255',
        'produced_on' => 'required|date',
    ]);
}
```



Formulář v create.blade.php:

```
<div class="form-group">
    <label for="make">Make</label>
    <input type="text"
        class="form-control" name="make" id="make" aria-describedby="helpId" />
    <small id="helpId" class="form-text text-muted">Car producer...</small>
</div>

<div class="form-group">
    <label for="model">Model</label>
    <input type="text"
        class="form-control" name="model" id="model" aria-describedby="helpId" />
    <small id="helpId" class="form-text text-muted">Car model...</small>
</div>

<div class="form-group">
    <label for="produced_on">Produced on</label>
    <input type="date"
        class="form-control" name="produced_on" id="produced_on" />
    <small id="helpId" class="form-text text-muted">Production date...</small>
</div>
```

Validace formulářových dat na straně serveru

```
public function store(Request $request)
{
    // server side validation
    $validatedData = $request->validate([
        'make' => 'required|max:255',
        'model' => 'required|max:255',
        'produced_on' => 'required|date',
    ]);
}
```

required = daný input formuláře je povinný, uživatel ho musí vyplnit

max:255 = maximální počet znaků daného vstupu je 255

date = vstup musí být typu datum

Více se lze dočíst v Available Validation Rules:

<https://laravel.com/docs/5.8/validation#available-validation-rules>

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

Chybová zpráva pro uživatele

Displaying The Validation Errors:

<https://laravel.com/docs/5.8/validation>

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

Chybová zpráva pro uživatele

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

Pokud dojde k validační chybě, tak se do session uloží pole, které obsahuje všechny chyby, jež nastaly.

V šabloně (create.blade.php) je přes lokální proměnnou \$errors možné pole chyb vypsát.

Pokud je v poli chyb nějaká chyba: \$errors->any(), pak @foreach (direktiva šablonovacího systému blade) projde všechny chyby a postupně je vloží do proměnné \$error. Obsah proměnné \$error se postupně vkládá do HTML seznamu ().

Celý chybový výpis je vložen do divu Bootstrapu <div class="alert alert-danger">. Výsledkem je červený styl vhodný pro zobrazování chyb.

```

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

```

Kód zkopírujeme sem:

create.blade.php:

```

<div class="container">
    <h1 class="my-5">Create new car</h1>
    <form action="{{ route('cars.store') }}" method="post">
        @csrf

        <div class="form-group">
            <label for="make">Make</label>
            <input type="text"
                class="form-control" name="make" id="make" aria-
            <small id="helpId" class="form-text text-muted">Ca
        </div>
    </form>
</div>

```

Pokračovat v: 1-Laravel_uprava_projektu_autosalon.txt

Create new car

- The model field is required.
- The produced on field is required.

Make

Car producer...

Model

Car model...

Produced on

Production date...

Submit

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

Create new car

Make

MAKEMM

Car producer...

Model

MODELMM

Car model...

Produced on

18.04.2019

Production date...

Submit

Server: localhost:8889 » Database: autosalon_db » Table: cars

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

SELECT * FROM `cars`

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

				id	make	model	produced_on	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	make01	model01	2010-10-01	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	make02	model02	2011-11-02	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	make03	model00	2012-12-03	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	MAKEMM	MODELMM	2019-04-18	2019-04-18 17:20:45	2019-04-18 17:20:45

OK

localhost:8888/oulehla/autosalon/public/cars

=> potřebujeme redirection, aby se uživatel nenacházel na prázdné stránce

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

jméno proměnné, která bude dostupná v šabloně
index.blade.php (bude mít k dispozici data z pole)

data, která vkládáme do pole

```
CarController.php x
15 public function index()
16 {
17     $allCars = Car::all();
18     return view('cars.index', [ 'allCars' => $allCars ]);
19 }
```

Pole všech automobilů si pošleme
z CarController.php z metody index().
Nyní je ve foreach v proměnné \$allCars
pole všech automobilů.

```
index.blade.php x
1 @extends('layouts.adminLayout')
2
3 @section('content')
4     <ul class="list-group">
5         @foreach($allCars as $car)
6             <li class="list-group-item">{{ $car->make }} {{ $car->model }}</li>
7         @endforeach
8     </ul>
9 @endsection
```

Administrace

×

+

←

→

↺

ⓘ

localhost:8888/OULEHLA/autosalon/public/cars

☆

⋮

Admin - autosalon

Home

Cars ▾

Search

Search

make01 model01

make02 model02

make03 model00

MAKEMM MODELMM

MAKENN MODELNN

MAKEO MODELO

Laravel – projekt autosalón - rozšíření

Pokračovat v:

1-Laravel_uprava_projektu_autosalon.txt

Laravel – Doporučení pro samostudium

Doporučení pro samostudium:

- V dokumentaci si projděte sekci The Basics <https://laravel.com/docs/5.8>.
- Doma si ještě jednou projděte poslední 2 cvičení, které se týkají FW Laravel.

Kontakt

- Ing. Milan Oulehla
- Email: oulehla@fai.utb.cz
- Kancelář: D215 (penetrační laboratoř)

Děkuji Vám za pozornost.

Máte nějaké otázky?