



WEB разработка

Laravel CRUD

Работа с потребителски данни

Съдържание

Laravel resource controllers and CRUD

Създаване на resource контролери

Routes&resource контролери

post, get, put, delete etc

route:list, except, only /route/

Security

- CSRF protection

- data escaping

Добавяне на пакети -

- Laravel Collective - HTML&Forms

Задача 1

Отпечатайте всички лекции на страница лекции с бутон за промяна и изтриване

Опечатайте списък на всички студенти с бутон за изтриване и бутон към профила им.

*въведете предварително нужната информация в базата данни

Laravel resource controllers

Laravel resource controllers

Resource Controllers

Laravel позволява декларирането на "CRUD" пътищата `/routes/` свързани с даден контролер с един ред код.

Задържа Създайте контролер, обработващ всички HTTP заявки, свързани с лекциите, които се пазят във вашето приложение.

//създаваме контролер

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

//командата създава контролера в `app/Http/Controllers/LectureController.php`.

`LectureController` съдържа метод за всяка от CRUD операциите.

Next, you may register a resourceful route to the controller:

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

This single route declaration creates multiple routes to handle a variety of actions on the resource. The generated controller will already have methods stubbed for each of these actions, including notes informing you of the HTTP verbs and URIs they handle.

Laravel resource controllers

Следваща стъпка - регистриране на resourceful route за **LectureController** -

Route::resource('lectures', 'LectureController')

С тази декларация регистрираме нужните пътища, които ще направят възможни действията свързани с управляването на съответния ресурс - лекциите в приложението, в случая.

Използвайте командата -

php artisan route:list

И разгледайте съществуващите пътища, отнасящи се до **LectureController**.

Laravel resource controllers

Какви заявки
изпълнява
resource
контролера ?

Actions Handled By Resource Controller

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

Laravel resource controllers

Spoofing Form Methods

HTML формите не изпълняват PUT, PATCH, or DELETE заявки.

Добавяме скрито `_method` поле. С този трик формите разбират заявката и я изпълняват.

```
{{ method_field('PUT') }}
```


Laravel resource controllers

Частични Resource пътища

Можем да изключим част от пътищата, които се декларират по подразбиране с `Route::resource`.

Така ограничаваме resource контролера и той изпълнява само част от действията.

```
Route::resource('lectures', 'LectureController', ['only' => ['index', 'show']]);
```

//позволяваме само показването на списъка от всички лекции и показването на една лекция.

//Никакви други действия, свързани с лекциите не са достъпни.

```
Route::resource('lectures', 'LectureController', ['except' => ['create', 'store', 'update', 'destroy']]);
```

//не са достъпни действията по създаване, запазване, актуализиране и изтриване

Laravel resource controllers

Даване на имена на Resource Routes

Всички действия/заявки/, които изпълнява resource контролера имат име по подразбиране, което може да бъде променено.

```
Route::resource('lectures', 'LectureController', ['names' => ['create' => 'lecture.build']]);
```

//проверете с route:list как изглежда пътя, регистриран по този начин

Пътищата, чиито имена променят, се добавят като елементи в масива 'names'

```
['names' => ['create' => 'lecture.build',  
            'show' => 'yourCustomRouteNameHere']
```

Laravel resource controllers

Допълване на Resource Контролерите

Ако е необходимо, може да добавите пътища към основните.

Допълнителните пътища трябва да бъдат дефинирани преди декларацията `Route::resource`.

```
Route::get('lectures/additional', 'LectureController@methodName'); //допълнителен
```

```
Route::resource('lectures', 'LectureController'); //основни
```

CSRF и Ларавел

CSRF и Ларавел

CSRF/Cross-site request forgeries/ се отнасят за случаите, в които неоторизирани заявки се изпълняват от името на оторизиран потребител.

Предпазването от CSRF в Ларавел става с автоматично генериран CSRF "token" за всяка активна потребителска сесия.

С този token се удостоверява, че оторизирания потребител е този, който прави заявка към приложението.

Всяка HTML форма в приложението трябва да съдържа скрито CSRF token поле, чрез което да се валидира заявката.

```
<form method="POST" action="/lectures">  
  {{ csrf_field() }}  
  ...  
</form>
```

Задача 2

Направете CRUD за

- Студенти - техните профили
 - Администраторът добавя, изтрива и променя профилите – до всяко име в списък със студенти има бутони за промяна и изтриване. Над списъка има бутон за добавяне на нови студенти.
- Курсовете и техните лекции
- Домашни

Добавяне на информация

Добавяне на информация

Стъпки

1. Форма с нужните полета
2. Добавяме `{{ csrf_field() }}`
3. В контролера метода `create()` служи за зареждане на формата
 1. Заявката е `get` – посочваме я в пътя в `web.php`
4. Регистрираме път или ползваме регистрираните `resource` пътища.

Добавяне на информация

Стъпки

5. В контролера метода `store()` служи за съхраняване на въведената информация. Заявката е `post`. Посочваме я в пътя и в отварящия таг на формата.
6. Регистрираме пътя

Променяне на информация

Променяне на информация

Стъпки

1. Тръгваме към контролера за промяна на информация с id на информацията, която ще променяме –
 1. `id) }}" class="btn btn-info">Edit`
2. В контролера метода edit() служи за зареждане на формата за промяна на информация. Подаваме id на информацията, като параметър.

Променяне на информация

Стъпки

3. Регистрираме пътя – в пътя фигурира id на информацията, която променяме

```
Route::get('user/{$id}/edit', 'UserController@edit')->name('edit_user_info');
```

Добавяне на допълнителни пакети в Ларавел

Инсталиране

Стъпка 1 Добавяме допълнителните пакет чрез Composer.

Вариант 1

Променяме `composer.json` - добавяме декларация, с която да бъде добавен `laravelcollective/html` към приложението ни.

```
"require": {  
    "laravelcollective/html": "~5.0" //пакета и необходимата версия  
}
```

Актуализираме Composer в конзолата с командата **composer update**

Има и втори начин - `composer require името-на-пакета` /прави обратното - сам добавя пакета в `composer.json`/

Инсталиране

Стъпка 1 Добавяме допълнителните пакет чрез Composer.

Вариант 2

Директно в конзолата с командата

composer require **името-на-пакета**

Командата добавя пакета в composer.json

Инсталиране

Стъпка 2 Добавяме новия **provider** в масива **providers** в **config/app.php**:

```
'providers' => [  
    ...  
    'Collective\Html\HtmlServiceProvider',  
    ...  
],
```

//всеки пакет, обикновено, съдържа инструкция за добавяне към Ларавел

Инсталиране

Finally, add two class aliases to the aliases array of config/app.php:

```
'aliases' => [  
    // ...  
    'Form' => 'Collective\Html\FormFacade',  
    'Html' => 'Collective\Html\HtmlFacade',  
    // ...  
],
```

Инсталиране

Стъпка 3 Добавяме синоним/и, с който ще извикваме класа/класовете в масива `aliases` в `config/app.php`:

```
        'aliases' => [  
            // ...  
            'Form' => 'Collective\Html\FormFacade',  
            'Html' => 'Collective\Html\HtmlFacade',  
            // ...  
        ],
```

Генериране на форми чрез `laravelcollective/html`

форми

Отварящ и затварящ form - тагове

```
{!! Form::open(array('url' => 'foo/bar')) !!}  
  //  
{!! Form::close() !!}
```

По подразбиране, тези декларации зареждат POST метод.

Може да зададете друг, чрез - `Form::open(array('url' => 'foo/bar', 'method' => 'put'))`

Обърнете внимание: `Laravelcollective/html` автоматично добавя

- GET, PUT и DELETE методите, както и
- скрито CSRF token поле

За останалите елементи на формата -
<https://laravelcollective.com/docs/5.0/html#text>

форми

Генериране на форми -

<https://laravelcollective.com/docs/5.0/html#text>