



WEB разработка

Laravel CRUD

Работа с потребителски данни

Съдържание

- Laravel resource controllers and CRUD
- Create resource controllers
- Routes&resource контролери
- post, get, put, delete etc
- route:list, except, only /route/
- CRUD
 - Base tables
 - Related tables
- Security
 - CSRF protection
 - data escaping
- Добавяне на пакети -
 - Laravel Collective - HTML&Forms

Задача 1

А. Отпечатайте всички курсове на страница курсове с бутон за промяна и изтриване. Добавете бутон за добавяне на нов курс.
Направете CRUD за курсовете

**въведете предварително нужната информация в базата данни*

Задача 1

Tip

Controller

Model

View

Routes

/sends

/modifies data,

/displays data

/permits

Requests

operates with DB/

asks for data/

requests flow/

data/

Laravel resource controllers

Laravel resource controllers

Resource Controllers

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

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

*HTTP заявки -

- get = read resource,*
- post = create resource*
- put = fully update resource,*
- patch = partially update resource,*
- delete = delete resource*

[More about post, put, patch request](#)

Laravel resource controllers

//създаваме контролер

php artisan make:controller CoursesController --resource

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

CoursesController съдържа метод за всяка от CRUD операциите.

Laravel resource controllers

Следваща стъпка - регистриране на resourceful routes за **CoursesController** -

Route::resource('courses', 'CoursesController')

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

Използвайте командата -

php artisan route:list

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

Laravel resource controllers

Какви
заявки
изпълнява
resource
контролера ?

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 |

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('courses', CourseController, ['only' => ['index', 'show']]);
```

//позволяваме само показването на списъка от всички курсове и показването на един курс.

//Никакви други действия, свързани с курсовете не са достъпни

! дефинираните по подразбиране методи, от контролера, които не се използват, се премахват от него!

```
Route::resource('courses', CourseController, ['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('courses', 'CoursesController', ['names' => ['create' => 'course.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' => 'course.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(courses'/additional', CoursesController@methodName'); //допълнителен
```

```
Route::resource(courses', CoursesController'); //основни пътища
```

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>
```

CRUD methods

CRUD methods

Create / Store -> <https://laravel.com/docs/6.x/eloquent#inserting-and-updating-models>
-> <https://laravel.com/docs/6.x/eloquent#mass-assignment>

Edit/Update -> <https://laravel.com/docs/6.x/eloquent#updates>

Destroy -> <https://laravel.com/docs/6.x/eloquent#deleting-models>

Задача 1

Направете CRUD за

- Курсове /Отпечатайте всички курсове на страница курсове с бутон за промяна и изтриване. Добавете бутон за добавяне на нов курс./
- Зали, с които разполага ВСО /Отпечатайте всички зали на страница зали с бутон за промяна и изтриване. Добавете бутон за добавяне на нова зала/
- Тип събитие,
- Тип файлове - rar, zip,
- Випуск
- Тип сертификат

**забележка* Формите да бъдат направени с Laravel/Collective

Задача 1

Checklist

- Таблица в БД - миграции
- Модел
- Контролер
- Вюта
- Пътища/с параметри/

Добавяне на допълнителни пакети в Ларавел

LaravelCollective/Html

Инсталиране

Стъпка 0 Намираме, необходимия пакет и прочитаме документацията, започвайки от стъпките за добавянето му към проект.

[LaravelCollective/Html](#)

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**

!трябва да сте сигурни, че исканата версия е съвместима с версията на Laravel и другите, инсталирани пакети!

Инсталиране

Вариант 2 Добавяме допълнителните пакет чрез Composer.

Директно в конзолата с командата

composer require името-на-пакета

Командата добавя пакета в composer.json като преценява коя версия на пакета да инсталира

Забележка Възможно е и с тази команда да поискате инсталирането на конкретна версия

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

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

[Повече за останалите елементи на формата](#)

{{ }} vs {!! !!}

{{ }} vs {!! !!}

{{}} - ескейпва съдържанието

{!! !!} - не ескейпва съдържанието. Само при гарантиран източник на отпечатваното съдържание!!

Демо

CRUD на свързани таблици

Задача 3

Направете CRUD на

А. Модулите

Модулите имат

- Име
- Курс, към който принадлежат

Задача 3

В. Лекциите

Лекциите имат

- Име
- модул
- Дата на провеждане
- **ново** Място на провеждане - зала

С. Студентите

Студентите имат

- Потребителско име
- Профилна информация - първо, второ име
- Курс, в който са записани