

SaleController

Напишем контроллер **SaleController**, обрабатывающий список объявлений, с действием `index()`, которое в будущем станет выводить страницу с перечнем объявлений, а сейчас - временную «заглушку» в виде обычного текста.

1. Создадим контроллер `SaleController`:

```
kebagina@INT-A209-3 W:\domains\localhost\kebagina_laravel\kebagina
$ php artisan make:controller SaleController

INFO Controller [W:\domains\localhost\kebagina_laravel\kebagina\app\Http/Co
ntrollers/SaleController.php] created successfully.
```

Команда `make:controller` утилиты `artisan` создает модуль с классом контроллера, чье имя указано после команды через пробел.

2. Откроем только что сгенерированный модуль `app\Http\Controllers\SaleController.php` и посмотрим на его код.

```
kebagina_laravel > kebagina > app > Http > Controllers > SaleController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class SaleController extends Controller
8  {
9      //
10 }
```

Контроллер объявлен в пространстве имен `App\Http\Controllers` и является производным от суперкласса `Controller`. Изначально он «пуст» - не содержит ни одного метода-действия.

3. Объявим в контроллере-классе `SaleController` действие `index()`, выводящее временную текстовую «заглушку»:

```
class SaleController extends Controller
{
    public function index() {
        return response('Здесь будет перечень объявлений')
            ->header('Content-Type', 'text/plain');
    }
}
```

Функция `response()` генерирует серверный ответ, представленный объектом класса `Illuminate\Http\Response`, на основе строки, переданной в параметре, и возвращает его в качестве результата.

Метод `header()` класса `Illuminate\Http\Response` помещает в текущий серверный ответ заголовок с заданными в параметрах именем и значением. Мы используем этот метод, чтобы поместить в ответ заголовок `content-Type` со значением `text/plain`, тем самым указав веб-обозревателю, что ответ содержит обычный текст.

Готовый ответ следует вернуть из метода-действия в качестве результата - чтобы Laravel смог отправить его клиенту.

4. Для просмотра результата в браузере исправим маршрут на главной странице

```
kebagina_laravel > kebagina > routes > 🐼 web.php > ...  
16  
17 //это удаляем:  
18 // Route::get('/', function () {  
19 //     return 'Я начинаю изучать фреймворк';  
20 // });  
21  
22 Route::get('/', [SaleController::class, 'index']);  
23
```

Не забудьте подключить контроллер в начале файла

```
kebagina_laravel > kebagina > routes > 🐼 web.php > ...  
2  
3 use App\Http\Controllers\RandomController;  
4 ✓ use App\Http\Controllers\SaleController;  
5 use Illuminate\Support\Facades\Route;  
6
```

5. Смотрим итог в браузере (тёмный фон установлен в самом браузере, не на сайте):

