

Публикация на тему

# Enlighten - документация для Laravel

*Модуль Enlighten для Laravel позволяет убить сразу двух зайцев: тесты и документацию для RESTFull API*

## **Анотация**

Простой пакет для документирования Laravel API.

С этим пакетом нет необходимости добавлять бесконечные блоки документации к каждому методу API, поддерживать десятки файлов readme или писать обширные вики, чтобы ваши API были документированы и синхронизированы с базой! Осветите свои приложения Laravel красивой документацией, автоматически созданной из ваших тестовых наборов, благодаря чему ваша документация всегда будет обновляться с учетом текущей версии вашего приложения.

A seamless package to document your Laravel APIs.

There is no need to add endless docblocks to each API method, maintain dozens of readme files, or write extensive wikis to keep your APIs documented and in sync with your codebase!

Enlighten your Laravel applications with a beautiful documentation generated automatically from your test suites, by doing so, your documentation will always be updated with the current version of your app.

## **Автор**

[Михалькевич Александр Викторович](#)

## **Публикация**

**Наименование** Enlighten - документация для Laravel

**Автор** А.В.Михалькевич

**Специальность** Модуль Enlighten для Laravel позволяет убить сразу двух зайцев: тесты и документацию для RESTFull API,

## **Анотация**

Простой пакет для документирования Laravel API.

С этим пакетом нет необходимости добавлять бесконечные блоки документации к каждому методу API, поддерживать десятки файлов readme или писать обширные вики, чтобы ваши API были документированы и синхронизированы с базой! Осветите свои приложения Laravel красивой документацией, автоматически созданной из ваших тестовых наборов, благодаря чему ваша документация всегда будет обновляться с учетом текущей версии вашего приложения.

приложения.

## **Anotation in English**

A seamless package to document your Laravel APIs.

There is no need to add endless docblocks to each API method, maintain dozens of readme files, or write extensive wikis to keep your APIs documented and in sync with your codebase!

Enlighten your Laravel applications with a beautiful documentation generated automatically from your test suites, by doing so, your documentation will always be updated with the current version of your app.

**Ключевые слова** laravel, enlighten, php, phpdoc, test, тесты

**Количество символов** 7650

# **Содержание**

[Введение](#)

1 [Установка](#)

2 [Создание тестов](#)

3 [Конфигурирование enlighten](#)

4 [Тесты и результаты тестов](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

## **Введение**

### **1 Установка**

Пакет Enlighten устанавливается в проект Laravel с помощью менеджера зависимостей Composer

```
composer require styde/enlighten --dev
```

Полная информация по установке проекта доступна по ссылке:

<https://github.com/StydeNet/enlighten>

Помимо установки самого пакета, необходимо создать дополнительную базу данных для вашего проекта. Подключение к базе осуществляется в файле `config/database.php`. После чего необходимо выполнить миграции.

### **2 Создание тестов**

Enlighten тесно интегрирован с системой тестирования Laravel.

По умолчанию каталог тестов приложения содержит два каталога: Feature и Unit.

**Feature**, функциональное тестирование.

Тесты функций могут тестировать большую часть кода и даже полный HTTP-запрос к конечной точке JSON. Как правило, большинство тестов приложения должны быть

функциональными. Эти типы тестов обеспечивают максимальную уверенность в том, что система в целом работает должным образом.

**Unit**, модульное тестирование.

Модульные тесты - это тесты, которые фокусируются на очень небольшой изолированной части кода. Тесты из каталога «Unit» не загружают приложение Laravel и, следовательно, не могут получить доступ к базе данных вашего приложения или другим службам фреймворка.

Итак, создаём функциональный тест

```
php artisan make:test AccountTest
```

После чего в папке tests/Feature появится файл AccountTest.php

В одном классе тестов может быть несколько функций с тестами, название функции должно начинаться с префикса "test\_", например "test\_get" или "test\_post".

Вносим следующие изменения в файл AccountTest.php:

```
namespace Tests\Feature;

use Illuminate\Foundation\Testing\RefreshDatabase;
use Illuminate\Foundation\Testing\WithFaker;
use Tests\TestCase;
/**
 * @description Информация по пользовательскому аккаунту. Модель Account
 */
class AccountTest extends TestCase
{
    /**
     * @description Аккаунт авторизованного пользователя
     * @method get
     * @return void
     */
    public function test_get()
    {
        $response =
$this->get('/account?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...');

        $response->assertStatus(200);
    }
    /**
     * @description Обновление аккаунта пользователя
     * @method post
     * @return void
     */
    public function test_post()
    {
        $response =
$this->post('/account?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...');

        $response->assertStatus(200);
    }
}
```

```
}
```

Обратите внимание на PHPdoc-блок @description, который мы прописали для метода и для всего класса. Enlighten будет его использовать для описания группы тестов (@description класса) и функции теста (@description метода).

## 3 Конфигурирование enlighten

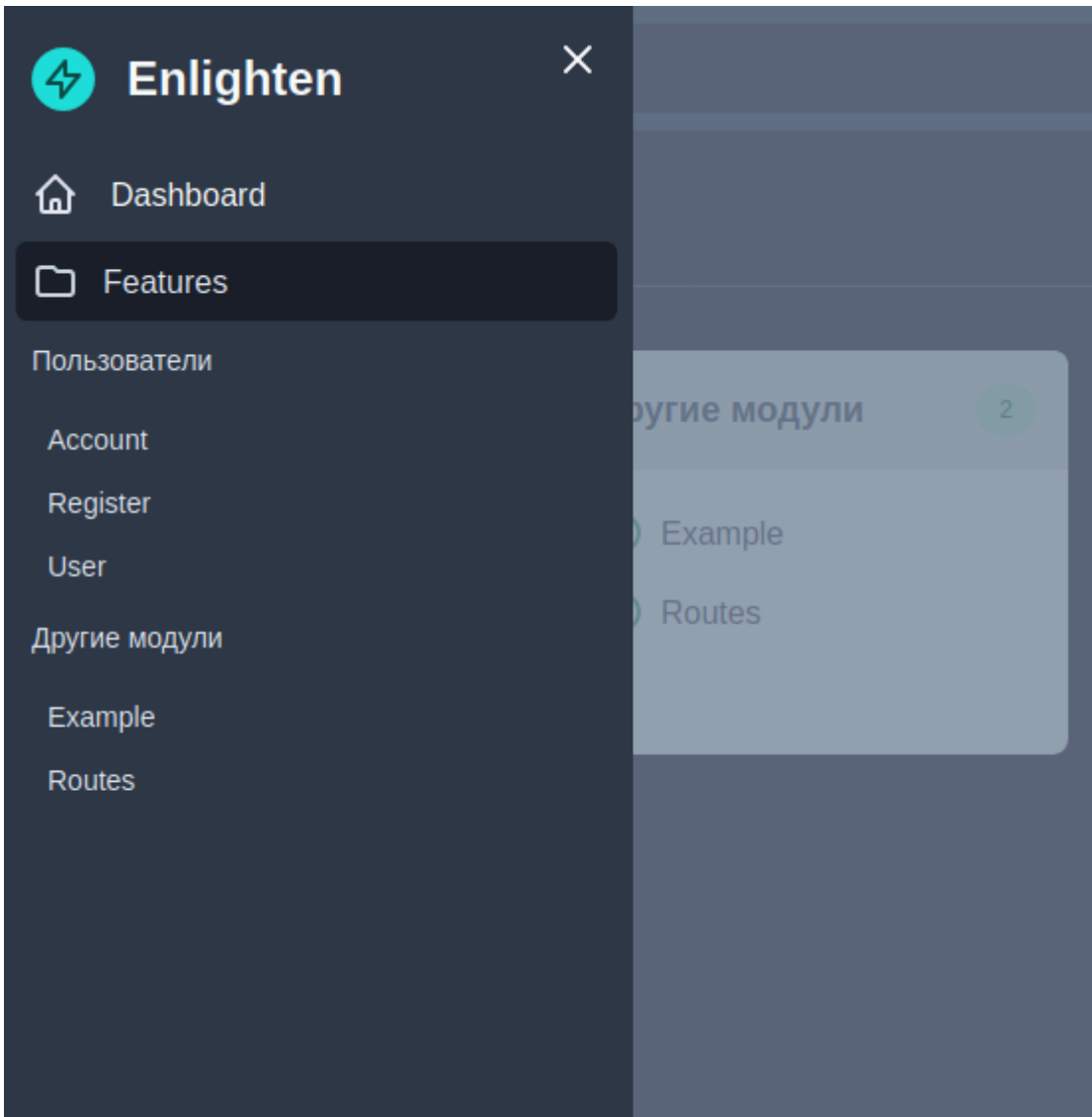
Настройки для Enlighten хранятся в файле config/enlighten.php.

Группировать тесты по категориям можно с помощью элемента массива modules:

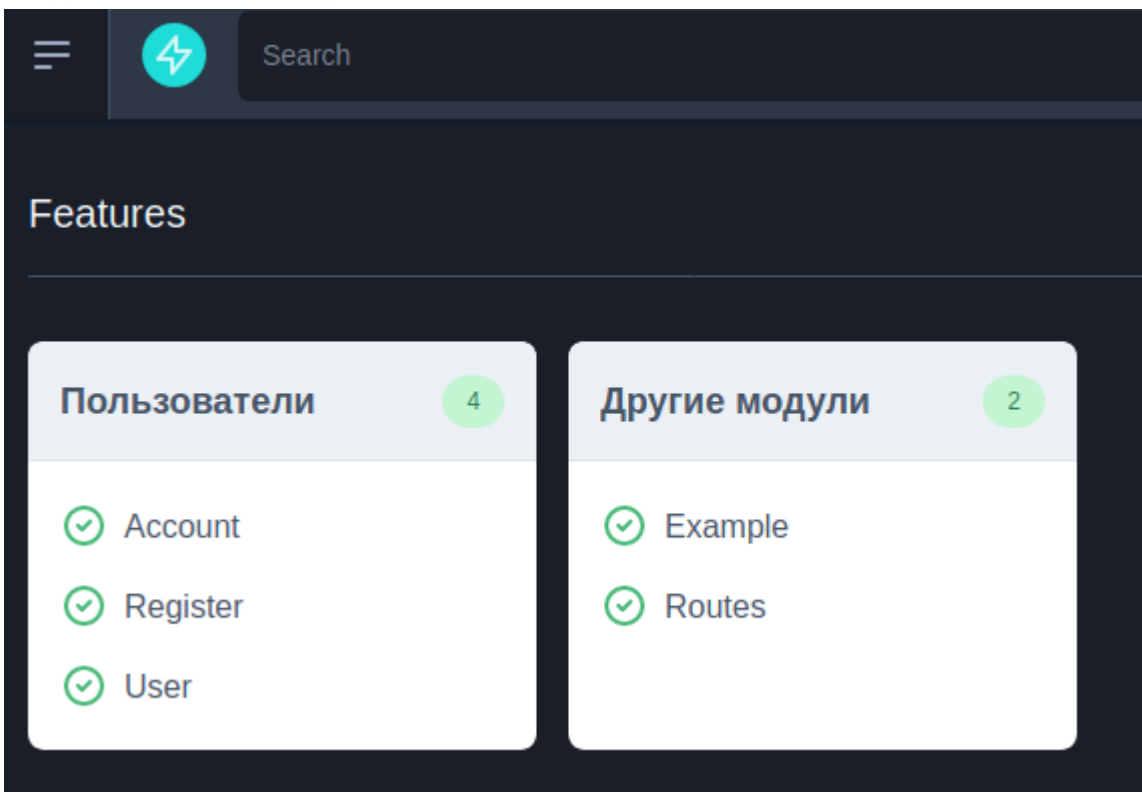
```
'modules' => [
    [
        'name' => 'Пользователи',
        'classes' => ['*User*', '*Register*', '*Account*'],
        'routes' => ['users/*'],
    ],
    [
        'name' => 'Другие модули',
        'classes' => ['*'],
    ],
]
```

Таким образом мы получили категории **Пользователи** и **Другие модули**. В каждой категории находятся группы (классы) тестов. В каждой группе - функция теста.

Т.к. мы создавали функциональный тест, в Enlighten меню всё это находится в разделе **Features**.



Раздел Features выглядит так:



## 4 Тесты и результаты тестов

После того как тесты созданы и конфигурирование Enlighten-а настроено, можно приступать к тестированию.

Перед запуском тестов, необходимо почистить кэш конфига. Запустить тесты можно командой

```
php artisan enlighten
```

```
mikhailkevich@mikhailkevich:/var/www/tattoo/api2$ php artisan config:clear
Configuration cache cleared!
mikhailkevich@mikhailkevich:/var/www/tattoo/api2$ php artisan enlighten
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git

  1/3 Tests\Unit\ExampleTest
  ✓ basic test
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git

  2/3 Tests\Feature\AccountTest
  ✓ get
  ✓ post
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git

  3/3 Tests\Feature\ExampleTest
  ✓ basic test
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git

  WARN Tests\Feature\RegisterTest
  ! example → This test did not perform any assertions /var/www/tattoo/api2/tests/Feature/RegisterTest.php:16
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git

  4/3 Tests\Feature\RoutesTest
  ✓ example
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
```

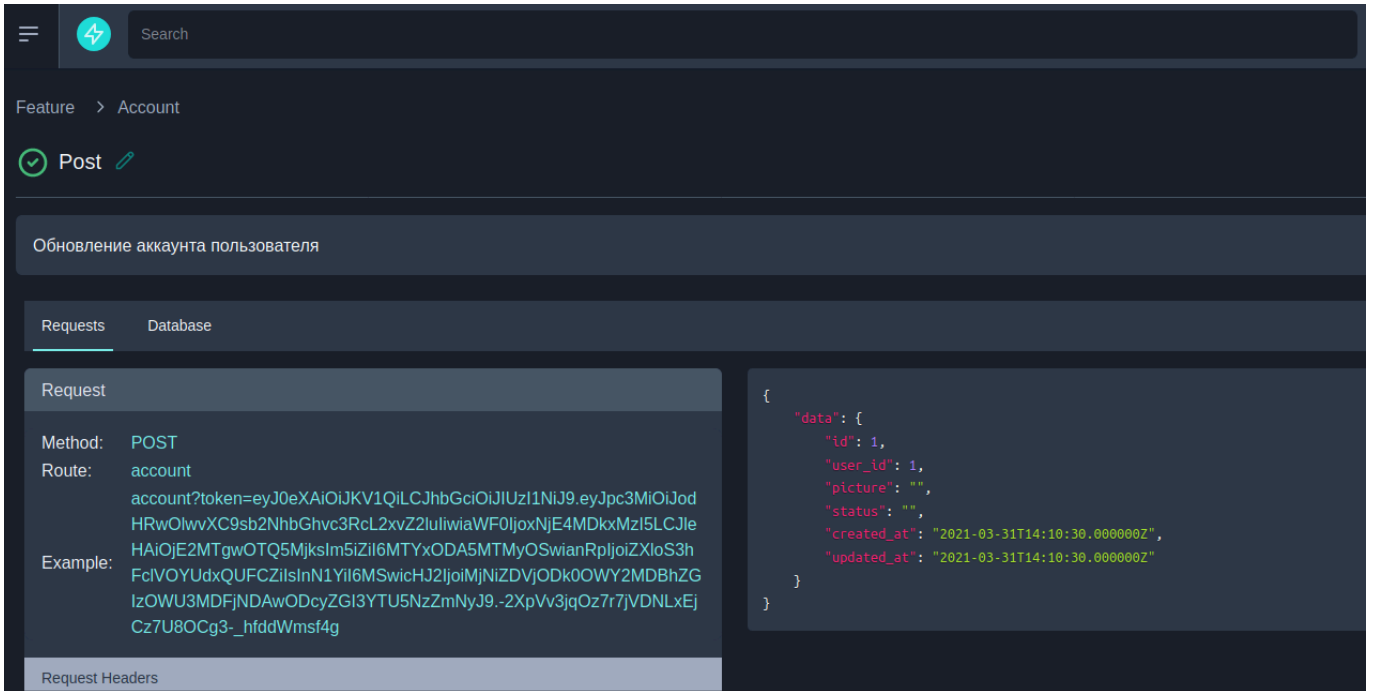
При этом успешные тесты сохраняются в базу данных Enlighten-а.

Подробную информацию (документацию проекта) по тестам и группам тестов, с учётом PHPdoc-блоков, можно получить по адресу:

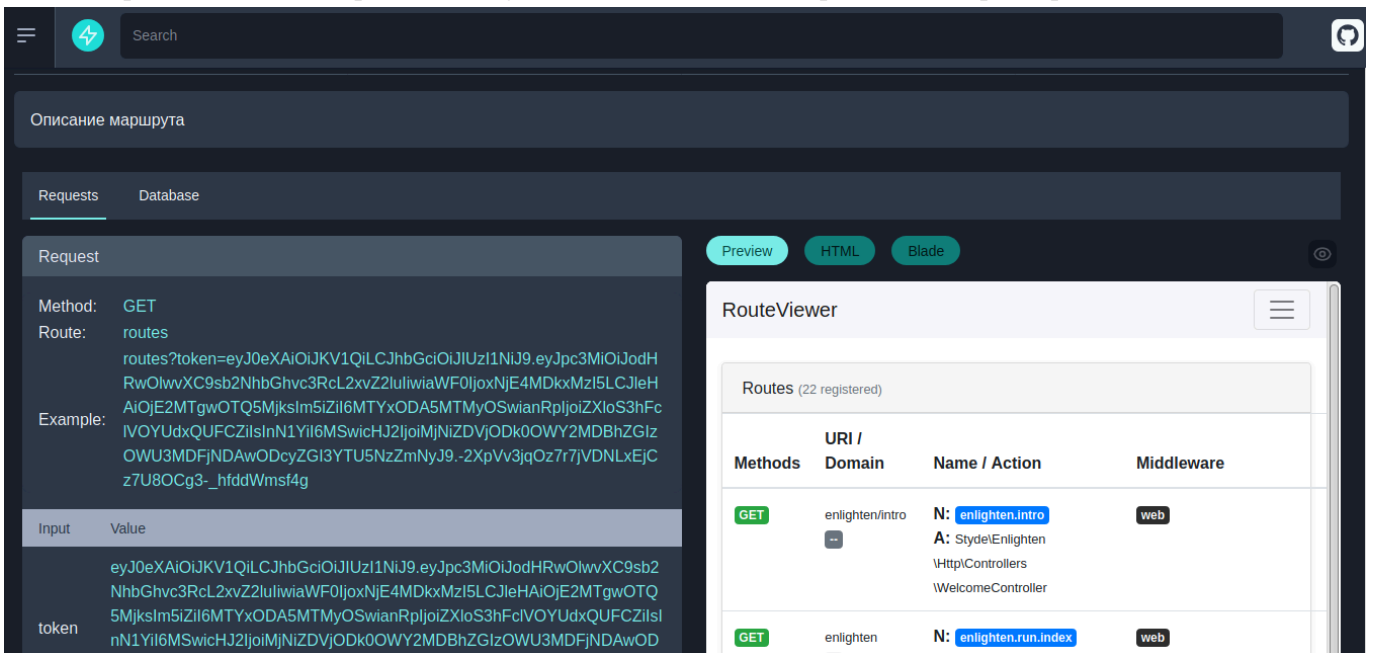
<http://localhost:8000/enlighten>

Чтобы получить полную информацию по тесту, заново его проходить не потребуется. Для этого достаточно перейти по определённой ссылке.

Так выглядит один из тестов группы Account (класса AccountTest.php)



Если response теста возвращает не JSON, а view(), то возвращается примерно такой ответ:



Так же имеется возможность экспортировать тесты в html-файлы, с помощью команды

```
php artisan enlighten:export
```

По сути, это та же документация, но доступна без сервера.

## Заключение

## Список использованных источников

## Приложения