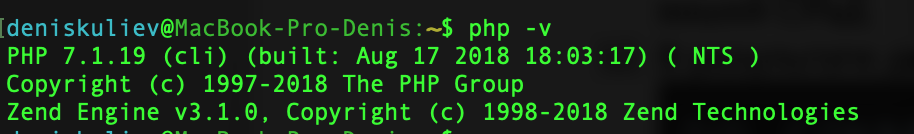
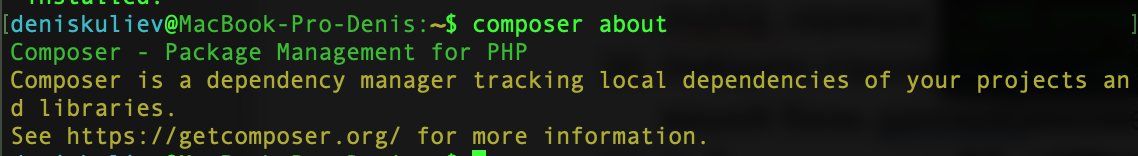
Пример приложения

Пример готового приложения вы можете увидеть [тут](https://github.com/CodeMonkeyInit/WorldskilllsApi/)

Установка Laravel

1. Установите/распакуйте интерпретатор php <http://php.net/downloads.php>
2. Если все прошло успешно, то вы можете написать в консоли [путь\_до\_php/]php -v и увидеть что-то подобное с поправкой на версию вашего интерпретатора (в моем случае это 7.1.19)  
   
3. Установите Composer <https://getcomposer.org/download/>
4. Если Composer установлен успешно, то вам станет доступна команда

composer about



Если команда вызовет ошибку вида:

PHP Fatal error: Declaration of Fxp\Composer\AssetPlugin\Repository\AbstractAssetsRepository::search($query, $mode = 0) must be compatible with Composer\Repository\ComposerRepository::search($query, $mode = 0, $type = NULL) in C:\Users\user\AppData\Roaming\Composer\vendor\fxp\composer-asset-plugin\Repository\AbstractAssetsRepository.php on line 334

то просто выполните следующую команду:

composer global remove fxp/composer-asset-plugin --no-plugins

1. Cкачайте Laravell Installer с помощью команды

composer global require laravel/installer

При скачивании возможно появления следующей ситуации: GitHub запросит у Вас значения токкена. Особенно это вероятно при ситуации, когда на компьютере уже использовался composer или установлено несколько версий PHP.

Для начала сгенерируйте токкен (<https://github.com/settings/tokens>).

Затем установите токкен.

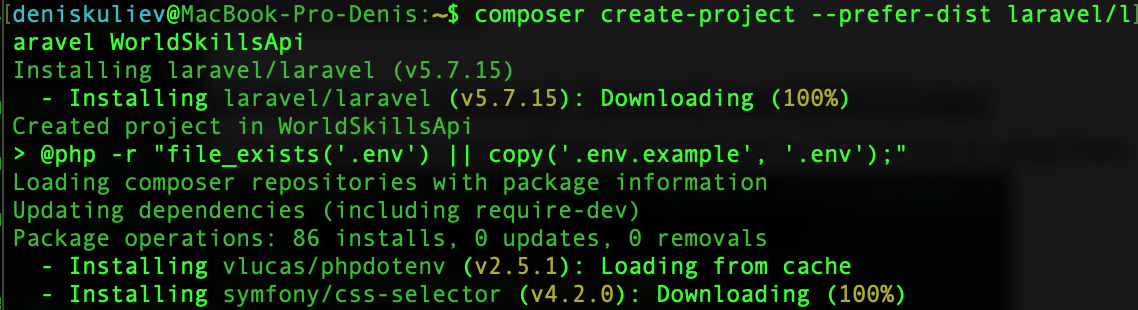
Вы можете задать значение токкена глобально:

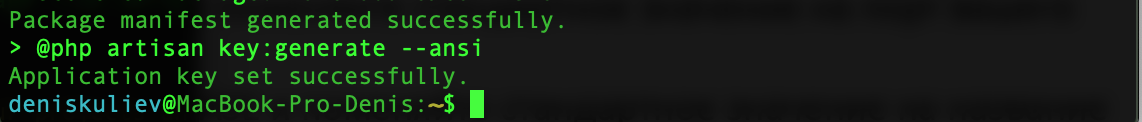
composer config --global github-oauth.github.com <TOKEN>

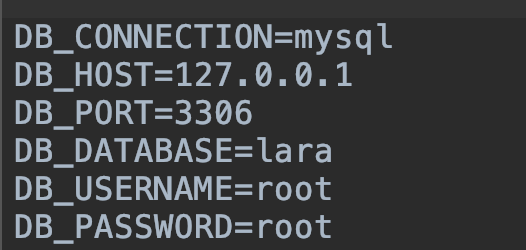
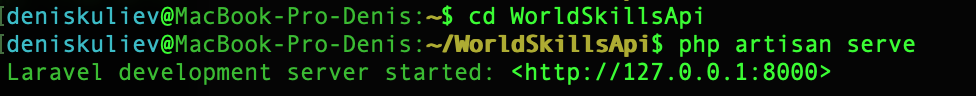
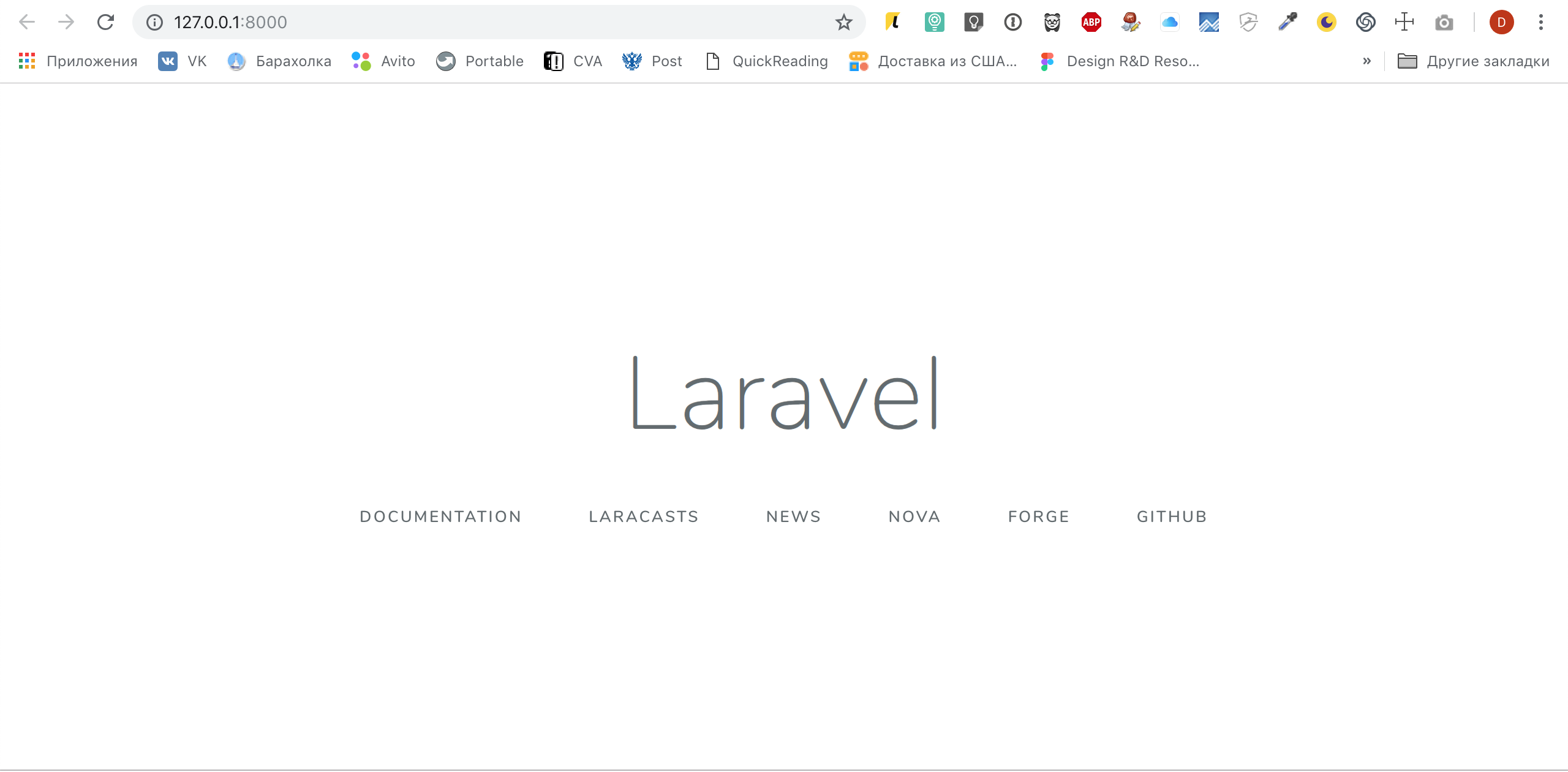
1. Создайте новый проект

composer create-project --prefer-dist laravel/laravel WorldSkillsApi

1. Если все прошло успешно, то вы увидите подобное окно(часть пакетов я намеренно упустил)





1. Сделайте копию .env.example и переименуйте его в .env
2. Найдите строчку DB\_HOST и поменяйте стандартное значение на адрес вашего mySQL сервера (в моем случае это 127.0.0.1)
3. Найдите строчку DB\_PORT и поменяйте стандартное значение на порт вашего mySQL сервера
4. Найдите строчку DB\_DATABASE и поменяйте стандартное значение на название вашей базы данных (рекомендую создать новую и указать ее имя сюда)
5. Найдите строчку DB\_USERNAME и поменяйте стандартное значение на логин к вашей СУБД
6. Найдите строчку DB\_PASSWORD и поменяйте стандартное значение на пароль к вашей СУБД
7. В результате должно выглядеть как-то так
8. 
9. Теперь вы можете зайти в ваш проект с помощью cd “путь к вашему проекту” и запустить команду php artisan serve
10. Теперь перейдите по адресу который выведет вам Laravel (в моем случае это [http://127.0.0.1:8000](http://127.0.0.1:8000/))
11. Если все прошло успешно, то в своем браузере вы увидите подобное
12. Laravel успешно установлен, можно приступать к выполнению задания

API авторизации

1. Отредактируйте миграцию сгенерированную Laravel create\_users\_table. (путь – путь\_к\_проекту \database\migrations) [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/database/migrations/2014_10_12_000000_create_users_table.php). Подробнее [тут](https://laravel.com/docs/5.7/migrations)
2. ВЫПОЛНИТЬ МИРАЦИЮ php artisan migrate. Если ошибка в файл AppServiceProvider.php (путь – \app\Providers) добавить строку use Illuminate\Support\Facades\Schema; и в метод boot() строку Schema::defaultStringLength(191);(<https://laravel-news.com/laravel-5-4-key-too-long-error>)
3. ДОБАВИТЬ SEED (файл DatabaseSeeder.php путь – \database\seeds)

DB::table('users')->insert([

'login' => 'admin',

'password' => Hash::make('password'),

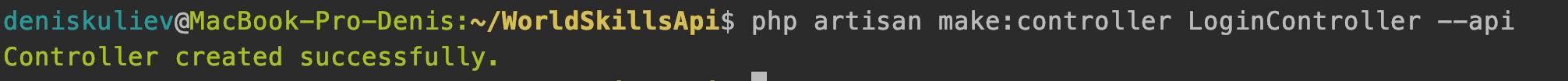
'token' => Hash::make('token'),

]);

Дополнительно необходимо добавить use Illuminate\Support\Facades\Hash;

ВЫПОЛНИТЬ SEED php artisan db:seed

1. Создайте API Controller с помощью команды php artisan make:controller LoginController –-api [Подробнее](https://laravel.com/docs/5.7/controllers)



1. Прописать путь к методу в файле routes/api.php. Пример: Route::post('/auth', 'LoginController@login'); [Подробнее](https://laravel.com/docs/5.7/routing)
2. Реализовать логин путем добавления кода в файл LoginController.php (путь – \app\Http\Controllers). Пример [тут](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/Auth/LoginController.php).

Метод jsonResponse($data, $statusCode, $statusMessage) реализован в файле Controller.php (путь – \app\Http\Controllers). Пример [тут](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/Controller.php) (не забываем про use Illuminate\Http\JsonResponse;)).

1. Проверить, что все работает, передав логин и пароль в POST запрос на путьКСайту/api/auth, это можно сделать с помощью Postman.

**API постов**

1. Создайте API Controller с помощью команды php artisan make:controller PostsController --api
2. Создайте прослойку (middleware) для валидации, что текующий пользователь администратор с помощью php artisan make:middleware CheckAdmin [Подробнее](https://laravel.com/docs/5.7/middleware)
3. Опишите псевдоним (alias) для вашего middleware в файл app/Http/Kernel.php. Путем добавления строки 'checkAdmin' => CheckAdmin::class в массив protected $routeMiddleware. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Kernel.php)(62 строчка).
4. Добавьте конфигурацию для хранения файлов, по пути config/filesystems.php. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/config/filesystems.php) (56 строчка). Пример использования: $imagePath = Storage::disk('post\_images')->put('', $request->image);
5. Опишите модель поста в файле \app\Post.php (можно использовать команду php artisan make:model Post). [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Post.php). [Подробнее](https://laravel.com/docs/5.7/eloquent)
6. Добавьте миграцию которая сгенерирует таблицу в вашей бд с помощью команды php artisan make:migration create\_posts\_table
7. Отредактируйте миграцию. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/database/migrations/2018_11_14_090354_create_posts_table.php). Выполните миграцию php artisan migrate
8. Опишите в методе store добавление нового поста. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (37 строчка)
9. Опишите в методе update для редактирования поста. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (93 строчка)
10. Опишите в методе destroy для удаления поста. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (127 строчка)
11. Опишите в методе index для вывода всех постов. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (21 строчка)
12. Опишите в методе show для вывода одного поста. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (70 строчка)
13. Опишите путь к методам store, update, destroy и middleware для проверки на администратора в файле routes/api.php.   
    Пример: Route::middleware('checkAdmin')->post('/posts', 'PostsController@store'); Route::middleware('checkAdmin')->post('/posts/{id}', 'PostsController@update'); Route::middleware('checkAdmin')->delete('/posts/{id}', 'PostsController@destroy');
14. Опишите путь к методам index и show в файле routes/api.php. Пример: Route::get('/posts', 'PostsController@index');

Route::get('/posts/{id}', 'PostsController@show');

1. Опишите модель комментария [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Comment.php). Можно использовать команду php artisan make:model Comment –m, которая автоматически создаст миграцию create\_comments\_table *(если надо сразу создать и контроллер можно использовать ключ –mc, или ключ –all(-a), чтобы создать дополнительно еще и ресурс)*
2. Отредактируйте миграцию, а затем выполните её. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/database/migrations/2018_11_14_090857_create_comments_table.php)
3. Добавьте метод для добавления комментария. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (149 строчка)
4. Опишите путь к методу и middleware для добавления комментария routes/api.php. Пример: Route::post('/posts/{id}/comments', 'PostsController@addComment');
5. Добавьте метод для удаления комментария. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (180 строчка)
6. Опишите путь к методу и middleware для удаления комментария в файле routes/api.php. Пример: Route::middleware('checkAdmin')->delete('/posts/{id}/comments/{commentId}', 'PostsController@removeComment');
7. Добавьте метод поиска постов по тегу. [Пример](https://github.com/CodeMonkeyInit/WorldskilllsApi/blob/master/app/Http/Controllers/PostsController.php) (201 строчка)
8. Опишите путь к методу для поиска по тегу в файле routes/api.php. Пример: Route::get('posts/tag/{tagName}', 'PostsController@searchByTag');