

Phát Triển Phần Mềm Mã Nguồn Mở

LARAVEL FRAMEWORK The PHP Framework for Web Artisans

NGUYỄN HẢI TRIỀU

Bộ môn Kỹ thuật phần mềm
Khoa Công nghệ thông tin
ĐH Nha Trang

Mục lục

- 1 Tổng quan về Laravel
- 2 Cài đặt Laravel

1 Tổng quan về Laravel

2 Cài đặt Laravel



- **Laravel** là một PHP Framework phổ biến nhất hiện nay được phát triển đầu tiên bởi Taylor Otwell. Ngoài ra, nó cũng là một framework mã nguồn mở (<https://laravel.com/>)
- Tính đến hiện tại 08/2020, Laravel phát hành tới phiên bản 8.x. Một số yêu cầu để cài đặt Laravel: [Laravel PHP >= 7.3](#), [BCMath PHP Extension](#), [Ctype PHP Extension](#), [Fileinfo PHP Extension](#), [JSON PHP Extension](#), [Mbstring PHP Extension](#), [OpenSSL PHP Extension](#), [PDO PHP Extension](#), [Tokenizer PHP Extension](#), [XML PHP Extension](#) (<https://laravel.com/docs/8.x>).

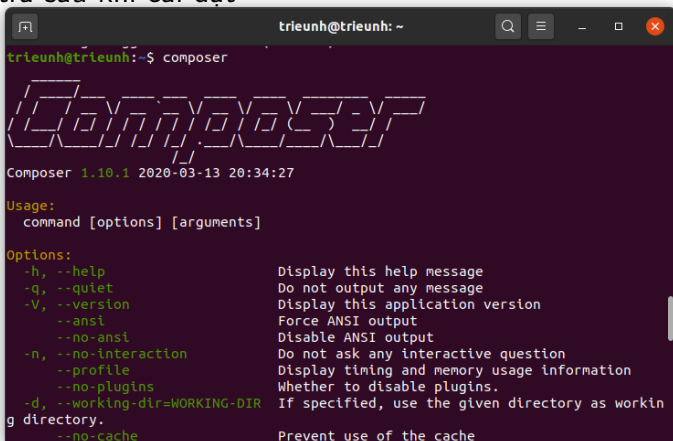
Laravel được viết bằng ngôn ngữ PHP chạy trên nền tảng web. Tương tự như PHP, môi trường để Laravel có thể chạy được gồm có

- Web server: Apache, Nginx
- PHP
- Hệ quản trị Cơ sở dữ liệu.

1 Tổng quan về Laravel

2 Cài đặt Laravel

- Laravel sử dụng **Composer (php package manager)** để quản lý các gói thư viện phụ thuộc trong project. Vì vậy, trước khi sử dụng Laravel, hãy đảm bảo đã cài đặt Composer.
- Cài đặt Composer bằng lệnh **sudo apt install composer** trên hệ điều hành Linux.
- Kiểm tra sau khi cài đặt



```
trieunh@trieunh: ~  
trieunh@trieunh:~$ composer  
  
Composer  
Composer 1.10.1 2020-03-13 20:34:27  
  
Usage:  
  command [options] [arguments]  
  
Options:  
  -h, --help                Display this help message  
  -q, --quiet               Do not output any message  
  -V, --version             Display this application version  
      --ansi                Force ANSI output  
      --no-ansi             Disable ANSI output  
  -n, --no-interaction      Do not ask any interactive question  
      --profile             Display timing and memory usage information  
      --no-plugins          Whether to disable plugins.  
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.  
      --no-cache            Prevent use of the cache
```

Cài đặt

Đầu tiên, cần chuyển đến thư mục cần cài Laravel. Cài đặt:

- Thông qua [Laravel Installer](#). Sử dụng các lệnh sau
 - ➊ `composer global require laravel/installer`
 - ➋ `laravel new blog`
- Thông qua [Composer Create-Project](#)
 - ➊ `composer create-project --prefer-dist laravel/laravel blog`

Cài đặt

- Sau khi hoàn thành quá trình cài đặt , Laravel sẽ tự động sinh ra **Application key**, khi chạy trên các môi trường khác nhau, thì sẽ sinh ra những key khác nhau.
- nếu clone mã nguồn từ một **nơi khác**, cần tạo lại **Application key** bằng lệnh
 - ▶ `php artisan key:generate`

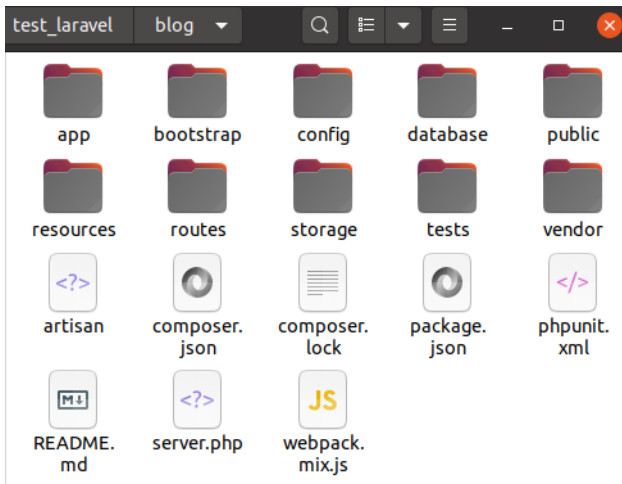
```

)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
facade/ignition suggests installing laravel/telescope (^3.1)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/php-code-coverage suggests installing ext-pcov (*)
phpunit/php-code-coverage suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
71 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan key:generate --ansi
Application key set successfully.
trieunh@trieunh: ~/NTU/ta_lieu_PHP/trieu_bai_giang/Chuong_06/test_laravel$

```

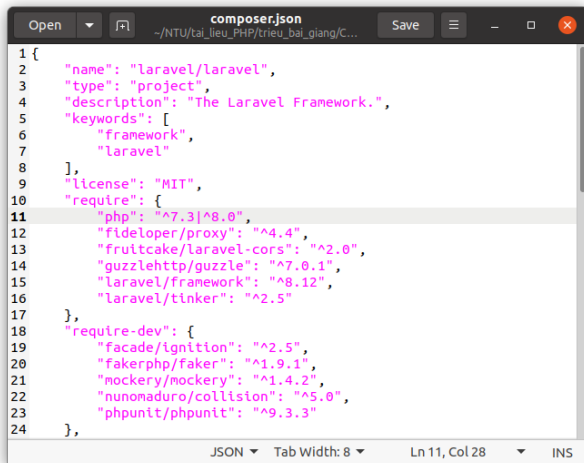
Kết quả sau khi cài đặt

Sau khi cài đặt, đi vào thư mục [blog](#) vừa tạo sẽ xuất hiện các file như hình dưới



File composer.json

Chứa các dependencies: danh sách tên các thư viện cài đặt cho project laravel.

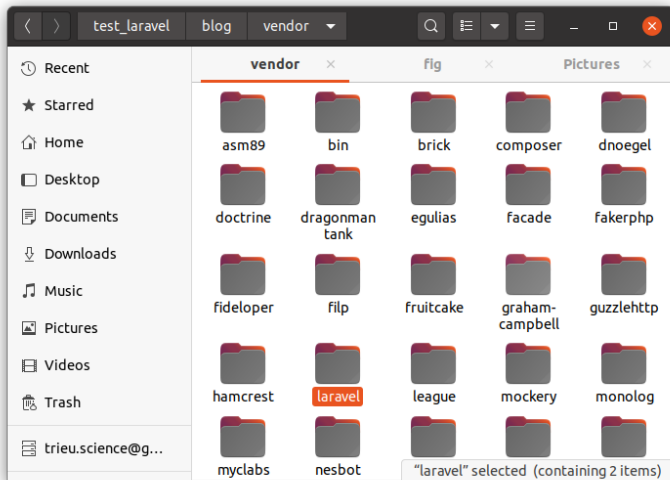


```
1 {
2     "name": "laravel/laravel",
3     "type": "project",
4     "description": "The Laravel Framework.",
5     "keywords": [
6         "framework",
7         "laravel"
8     ],
9     "license": "MIT",
10    "require": {
11        "php": "^7.3|^8.0",
12        "fideloper/proxy": "^4.4",
13        "fruitcake/laravel-cors": "^2.0",
14        "guzzlehttp/guzzle": "^7.0.1",
15        "laravel/framework": "^8.12",
16        "laravel/tinker": "^2.5"
17    },
18    "require-dev": {
19        "facade/ignition": "^2.5",
20        "fakerphp/faker": "^1.9.1",
21        "mockery/mockery": "^1.4.2",
22        "nunomaduro/collision": "^5.0",
23        "phpunit/phpunit": "^9.3.3"
24    },
25 }
```

JSON Tab Width: 8 Ln 11, Col 28 INS

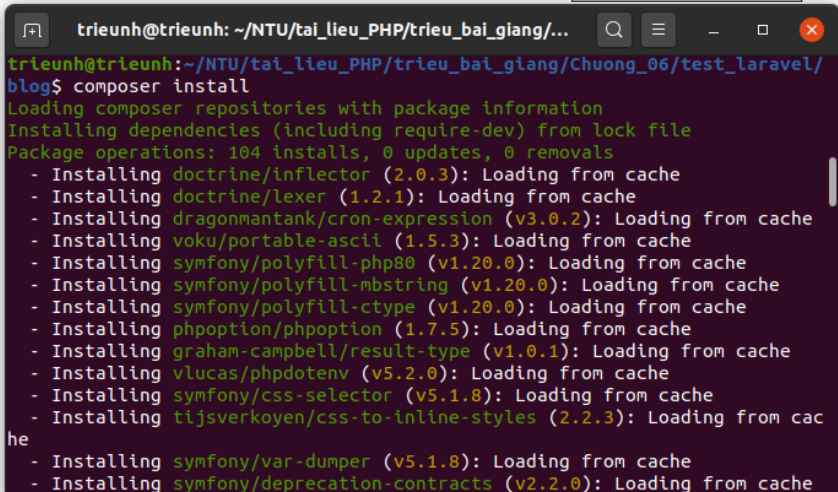
Thư mục Vendor

Chứa toàn bộ **source** của các thư viện từ danh sách thư viện trong file *composer.json*



Thư mục Vendor

Nếu clone một dự án Laravel đã có, cần phải cài đặt lại các thư viện theo danh sách *composer.json*. Sử dụng lệnh: `composer install`

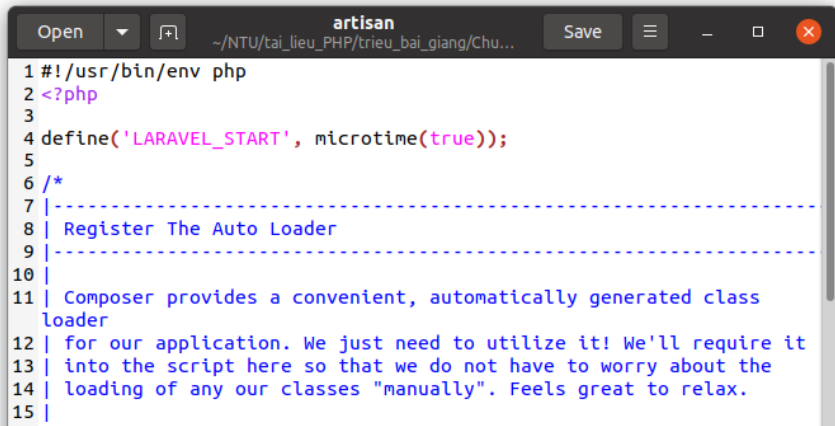


```
trieunh@trieunh: ~/NTU/tai_lieu_PHP/trieu_bai_giang/...
trieunh@trieunh:~/NTU/tai_lieu_PHP/trieu_bai_giang/Chuong_06/test_laravel/
blog$ composer install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 104 installs, 0 updates, 0 removals
- Installing doctrine/inflector (2.0.3): Loading from cache
- Installing doctrine/lexer (1.2.1): Loading from cache
- Installing dragonmantank/cron-expression (v3.0.2): Loading from cache
- Installing voku/portable-ascii (1.5.3): Loading from cache
- Installing symfony/polyfill-php80 (v1.20.0): Loading from cache
- Installing symfony/polyfill-mbstring (v1.20.0): Loading from cache
- Installing symfony/polyfill-ctype (v1.20.0): Loading from cache
- Installing phpoption/phpooption (1.7.5): Loading from cache
- Installing graham-campbell/result-type (v1.0.1): Loading from cache
- Installing vlucas/phpdotenv (v5.2.0): Loading from cache
- Installing symfony/css-selector (v5.1.8): Loading from cache
- Installing tijsverkoyen/css-to-inline-styles (2.2.3): Loading from cac
he
- Installing symfony/var-dumper (v5.1.8): Loading from cache
- Installing symfony/deprecation-contracts (v2.2.0): Loading from cache
```

artisan

Là một công cụ dòng lệnh được Laravel cung cấp sẵn nhằm thực thi một số tác vụ và được viết bằng php. Cú pháp sử dụng artisan như sau:

php artisan lệnh_cần_thực_thi

A screenshot of a code editor window titled 'artisan'. The window shows the contents of the file located at '~/.NTU/tai_lieu_PHP/trieu_bai_giang/Chu...'. The code is a PHP script that defines a constant 'LARAVEL_START' and includes a comment about registering the auto loader. The code is as follows:

```
1#!/usr/bin/env php
2<?php
3
4define('LARAVEL_START', microtime(true));
5
6/*
7|-----|
8| Register The Auto Loader
9|-----|
10|
11| Composer provides a convenient, automatically generated class
12| loader
13| for our application. We just need to utilize it! We'll require it
14| into the script here so that we do not have to worry about the
15| loading of any our classes "manually". Feels great to relax.
16|
```

artisan

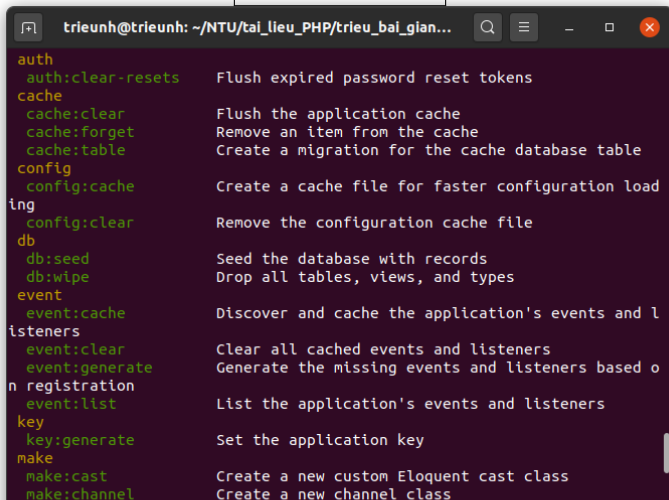
Khi sử dụng các lệnh **artisan**, công việc sẽ được xử lý tự động, nhanh chóng:

- **xử lý cơ sở dữ liệu** như migrate, đưa dữ liệu vào database, tạo dữ liệu kiểm thử.
- **tạo các template** là các Class trong lập trình theo các mẫu khác nhau như tạo ra các **Model, Controller, Event ...**
- các công việc liên quan đến **bảo mật** như cài đặt xác thực người dùng, sinh key mã hóa và các việc liên quan đến sử dụng **OAuth2**.
- cài đặt các package
- tối ưu hệ thống (optimize)

artisan

Để xem danh sách các câu lệnh trong artisan, sử dụng lệnh sau:

```
php artisan list
```



```
trieunh@trieunh: ~/NTU/tai_lieu_PHP/trieu_bai_gian...  
auth  
  auth:clear-resets    Flush expired password reset tokens  
cache  
  cache:clear          Flush the application cache  
  cache:forget         Remove an item from the cache  
  cache:table          Create a migration for the cache database table  
config  
  config:cache         Create a cache file for faster configuration load  
ing  
  config:clear         Remove the configuration cache file  
db  
  db:seed              Seed the database with records  
  db:wipe              Drop all tables, views, and types  
event  
  event:cache          Discover and cache the application's events and l  
isteners  
  event:clear          Clear all cached events and listeners  
  event:generate       Generate the missing events and listeners based o  
n registration  
  event:list           List the application's events and listeners  
key  
  key:generate         Set the application key  
make  
  make:cast            Create a new custom Eloquent cast class  
  make:channel         Create a new channel class
```


artisan

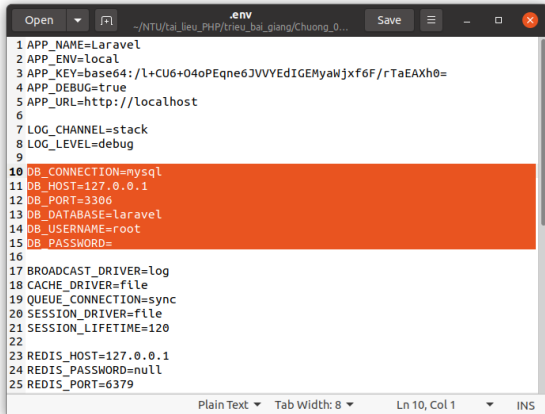
Một số lệnh artisan cơ bản:

- Khởi động `webserver` để chạy app laravel
 - ▶ `php artisan serve`
- Khi bảo trì hoặc bảo trì xong cần chuyển trạng thái hoạt động của hệ thống
 - ▶ `php artisan down` // `php artisan up`

File .env

File `.env` là file chứa các thiết lập môi trường

- Thiết lập **cấu hình Cơ sở dữ liệu**



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:/l+CU6+04oPEqne6JVYEdIGEMyaWjxf6F/rTaEAXh0=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=laravel
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 QUEUE_CONNECTION=sync
20 SESSION_DRIVER=file
21 SESSION_LIFETIME=120
22
23 REDIS_HOST=127.0.0.1
24 REDIS_PASSWORD=null
25 REDIS_PORT=6379
```

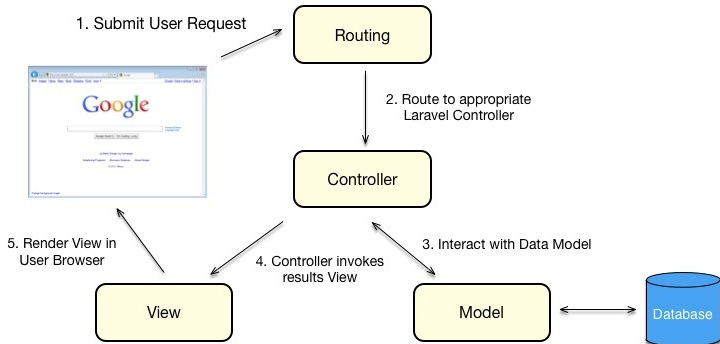
- Ngoài ra còn có thể thiết lập các biến môi trường như: host, mail ...

Định nghĩa

- **Controller:** Đóng vai trò điều phối, kiểm soát luồng thực thi của ứng dụng. Nhận lệnh từ người dùng, gửi lệnh đến cho **Model** để tiếp nhận, xử lý việc trả về dữ liệu/đối tượng, từ đó truyền lệnh đến **View** để hiển thị.
- **Model:** Tiếp nhận, Xử lý việc trả về dữ liệu/đối tượng khi nhận được yêu cầu từ **Controller** (nếu có) và trả thông tin về cho **Controller**.
- **View:** **Controller** sau khi nhận được thông tin trả về từ Model nó sẽ trả về cho View. View sẽ trả về định dạng hiển thị (HTML) trên màn hình user.

Định nghĩa

Hình 1: Laravel MVC <http://grigorov.website/blog/how-to-organize-your-project-with-php-and-laravel-to-get-the->



Auth

Laravel hỗ trợ sẵn các chức năng đăng nhập, đăng ký, xác thực địa chỉ email, xác thực 2 lớp (two-factor authentication), quản lý session, hỗ trợ API. Sử dụng lệnh sau đối với vesion 8.x

```
composer require laravel/jetstream
```

```
php artisan jetstream:install livewire
```

```
php artisan jetstream:install inertia
```

Sau đó cài lệnh theo yêu cầu `npm install && npm run dev` -> Lệnh này có nhiệm vụ build thư mục `resources/js` vào trong `public/(js/css)`

Auth

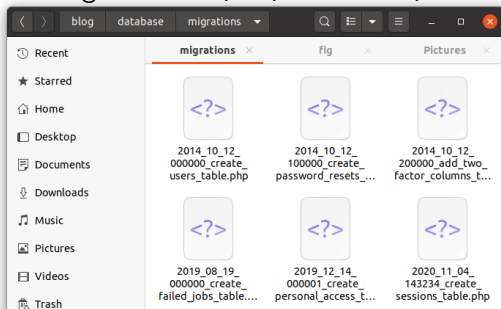
Sau khi cài đặt đăng nhập, bước tiếp theo cần thực hiện là:

- ❶ Tạo một cơ sở dữ liệu mới để lưu các thông tin liên quan đến Auth.
- ❷ Khai báo kết nối CSDL trong file `.env`
- ❸ Chạy lại lệnh `php artisan migrate`

Laravel Migration

- Laravel cung cấp công cụ tự động trong việc **quản lý các phiên bản cơ sở dữ liệu (database version control)**.
- Nhờ đó mà các công việc như tạo cấu trúc bảng, cột, index, khóa chính, khóa ngoại, tạo dữ liệu test ... trở nên đơn giản
- Migration là các file php được laravel sinh tự động, và được đặt trong thư mục **database/migrations**

Hình 2: Các file migrations được tạo khi cài đặt Auth ở phần trước



Laravel Migration


Cấu trúc của một file php migration gồm có

- Khai báo thư viện sử dụng
- Câu lệnh SQL
- Trạng thái là tạo mới hay xóa table.



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->string('password');
22             $table->rememberToken();
23             $table->foreignId('current_team_id')->nullable();
24             $table->text('profile_photo_path')->nullable();
25             $table->timestamps();
26         });
27     }
28
29     /**
```


Kết quả Auth



Email


admin@gmail.com

Password

••••••••

☐ Remember me

[Forgot your password?](#) **LOGIN**



Name

|

Email

Password

Confirm Password

[Already registered?](#) **REGISTER**

Định nghĩa

- Laravel route định tuyến
- Laravel Middleware cơ chế bộ lọc trung gian
- HTTP Request trong Laravel

Reference

- [1] Lê Thị Bích Hằng, Bài giảng *Phát Triển Phần Mềm Mã Nguồn Mở*.
- [2] <https://www.w3schools.com/php/>
- [3] <https://www.allaravel.com/>
- [4] <https://laravel.com/docs/8.x>
- [5] <https://www.codehub.com.vn/>