# Get start

## Chuyển terminal sang git bash

"C:\Program Files\Git\bin\sh.exe" --login -i

## Tạo pj

laravel new new-pj

## Chạy server

php artisan serve

# Thao tác với db

## kết nối db

trong file .env

**đổi tên db và thông tin đăng nhập đúng với db đã tạo**

DB\_DATABASE=demoProject  
DB\_USERNAME=root  
DB\_PASSWORD=

## **Lưu ý**

Nếu đang dùng MySQL dưới 5.7.7 hoặc MariaDB dưới 10.2.2

Để migrate thành công phải sửa file App\Providers\AppServiceProvider

use Illuminate\Support\Facades\Schema;  
  
class AppServiceProvider extends ServiceProvider  
{  
 public function boot()  
 {  
 //  
 Schema::defaultStringLength(191);  
 }

}

## **Tạo bảng**

php artisan make:model Ten-model -m

trong database/migrations có 1 file mới được tạo có tên dạng 2021\_05\_29\_163801\_create\_ten-models\_table.php, chỉnh sửa func để định hình bảng

public function up()  
{  
 Schema::create('categories', function (Blueprint $table) {  
 $table->id();  
 $table->string('name');  
 $table->integer('parent\_id')->default(0);  
 $table->string('slug');  
 $table->timestamps();  
 });  
}

xem them syntax tại <https://laravel.com/docs/8.x/migrations>

sau khi chạy xong, chạy lệnh để migrate

**php artisan migrate**

# Thao tác View

## Get view

Tất cả các router đều chứa trong routes/web.php, cấu trúc trỏ đến view cơ bản:

Route::get('/', function () {  
 return view('viewName');  
});

View sẽ chứa trong resouces/views dưới tên viewName.blade.php, viết html vào file đấy là được

## Tham chiếu

Include dùng để tham chiếu 1 vew khác vào, ví dụ nếu muốn include view navBar trong thư mục partials vào trong div thì viết:

<div>  
 @include('partials.navBar')  
</div>

Khi muốn tạo một file template, ta cần dung yield kết hợp section:

Trong view template

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>  
 @yield('title')  
 </title>

…  
</head>  
<body class="hold-transition sidebar-mini">  
<div class="wrapper">  
 …  
 @yield('content')

…

</div>  
 …

</body>  
</html>

Trong view page, router sẻ lấy file này

Có 2 kiểu viết section:

@extends('layouts.admin')

{{--kiểu 1: chỉ text, không cần phải endsection--}}  
@section('title', 'Trang Chủ')  
  
{{--kiểu 2: html, phải có endsection--}}  
@section('content')  
 <p>viết html cho content ở đây</p>  
@endsection

# Thao tác controller

## Router đơn giản

tạo mới model có link categories/create

Tạo controller CategoryController

**php artisan make:controller CategoryController**

Setup router: trong routes/web.php, thêm mới router

Route::prefix('categories')->group(function () {  
 Route::get('/create', [  
// link router  
 'as' => 'categories.create',  
// đường dẫn tới controller, trong controller CategoryController gọi hàm create()  
// nếu điền CategoryController@create mà không hiện thì mới khai báo như dưới  
 'uses' => 'App\Http\Controllers\CategoryController@create'  
 ]);  
});

});

Controller mới được tạo ở app/Http/Controllers, trong CategoryController.php tạo func create

class CategoryController extends Controller  
{  
 public function create(){  
// in ra màn hình chữ created:  
// dd('created');  
// hoặc trả về vew  
 return view('category.add');  
 }  
}

trong view, để gán link vào thẻ a:

<a href="{{ route('categories.create') }}" class="btn btn-success m-2">Add</a>