**HỌC LARAVEL**

1. **Cài đặt composer:**

Cài đặt tại địa chỉ: <https://getcomposer.org/download/>

1. **Cài đặt Laravel:**

Mở cửa sổ cmd tại thư mục htdocs và gõ câu lệnh:

*Composser create-project –prefer-dist laravel/laravel TenProject*

1. **Route, định danh của route và nhóm route:**

Route:

Route::get('xinchao',  function(){

    return 'Xin chào các bạn';

});

Định danh của route:

Route::get('xinchao', ["as"=>"xinchao",function(){

    return 'Xin chào các bạn';

}]);

Gọi route khi đã được định danh:

Route::get('hi',function(){

    return redirect()->route('xinchao');

});

Giới hạn của route:

Route::get('hello/{ngay}',["as"=>"hello", function($ngay){

    echo "Hôm nay là ngày: ".$ngay;

}])->where(['ngay'=>'[0-9]+']);

Nhóm route (Route group):

Route::group(["prefix"=>"MyUser"],function(){

    Route::get("User1",function(){

        echo "User 1";

    });

    Route::get("User2",function(){

        echo "User 2";

    });

});

1. **Controller**

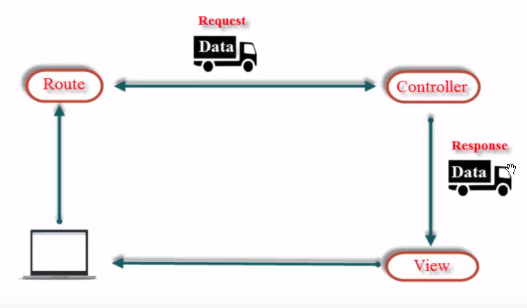
Tạo một controller mới: *artisan make:controller TenController.*

Gọi controller: *Route::phuongthuc(‘DuongDan’, ‘TenController@TenHam’);*

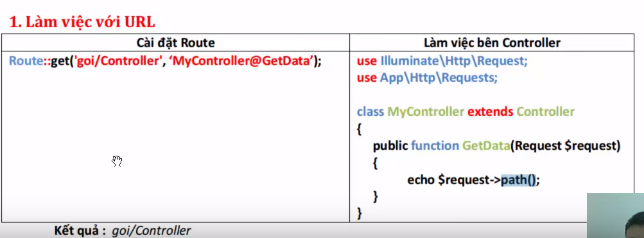
1. **Truyền dữ liệu cho controller**

Gọi route trong controller: return redirect()->route(‘TenRoute’);

1. **Làm việc với URL trên Request**



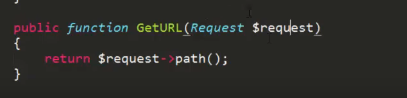
Hình 1- Sơ đồ mô tả mối quan hệ giữa route- controller- view



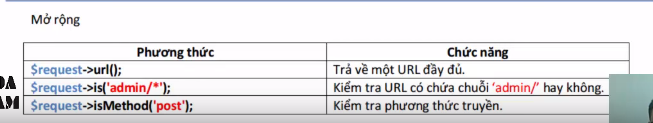
Hình 2- làm việc với URL



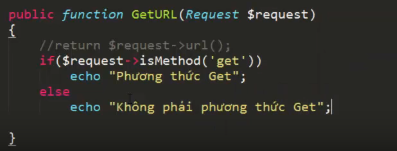
Hình 3- get URL (xử lý bên route)



Hình 4- get URL (xử lý bên controller)

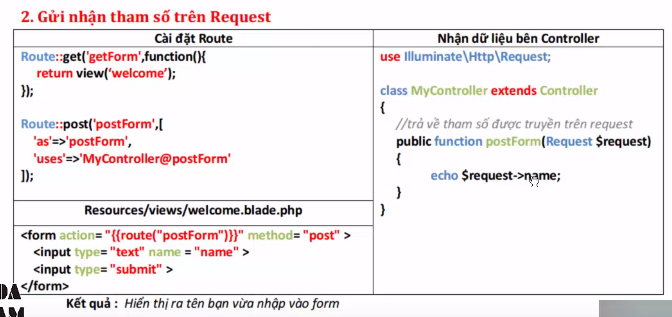


Hình 5- các phương thức request mở rộng

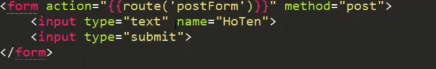


Hình 6- ví dụ kiểm tra phương thức get

1. **Gửi nhận tham số trên request**



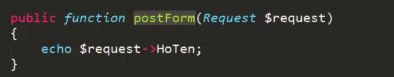
Hình 7- gửi nhận tham số trên request



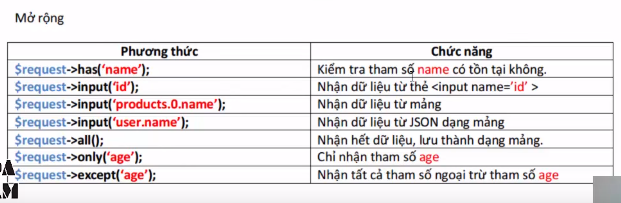
Hình 8- post (xử lý bên view).



Hình 9- post (xử lý bên route)

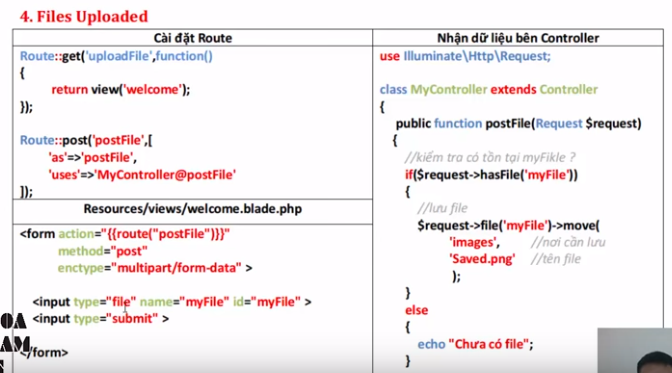


Hình 10- post (xử lý bên controller)

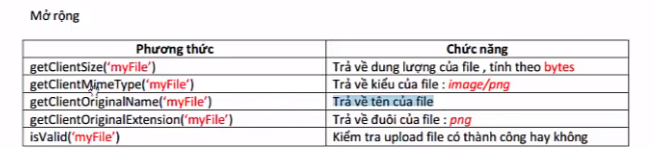


Hình 11 - các phương thức gửi nhận tham số trên request mở rộng

1. **Upload file**

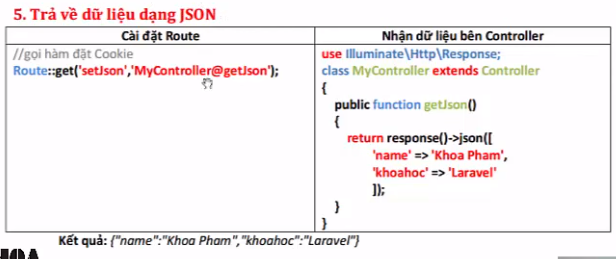


Hình 12- Upload file



Hình 13- các hàm mở rộng khi upload file

1. **Xuất dữ liệu dạng JSON.**



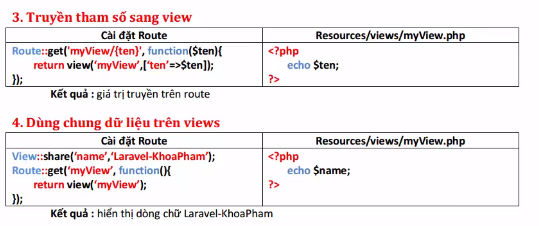
Hình 14 - Xuất dữ liệu dạng JSON

1. **Tìm hiểu về view.**

Trả về view có tên myview.php: return view(myview).

Trả về view có tên myview.php nằm trong thư mục view: return view(view.myview)

1. **Truyền dữ liệu sang view.**



Hình 15 - Truyền dữ liệu sang view và dùng chung dữ liệu trên view

Muốn truyền tham số trên view, trong controller, khi viết hàm cần truyền kèm tham số.

Ví dụ: Truyền tham số t từ route sang view:

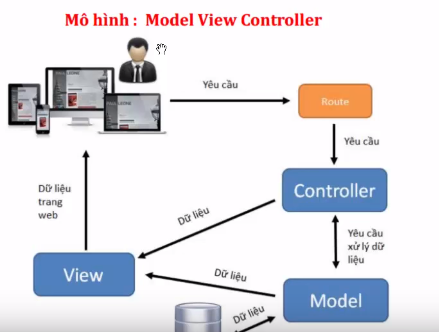
Return view(‘tenview’,[‘t’=>$t]);

Muốn dùng chung dữ liệu, trong route cần khai báo:

View::share(‘TenBien’,’GiaTri’);

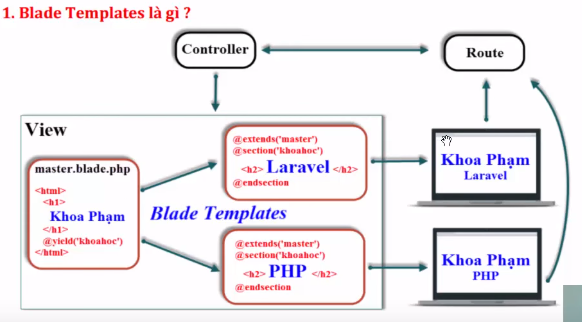
Sau khi khai bao, biến có thể sử dụng trong mọi view của project.

1. **Mô hình MVC trong Laravel.**



Hình 16- Mô hình MVC trong Laravel

1. **Tìm hiểu về Blade Template.**



Hình 17- Blade template là gì.

Muốn sử dụng file blade template ta cần:

Viết dòng sau tại vị trí cần nhúng nội dung từ file blade template khác vào:

@yield(‘NoiDung’)

File blade templade được nhúng vào cần có các dòng sau:

@extends(‘TenFile’)

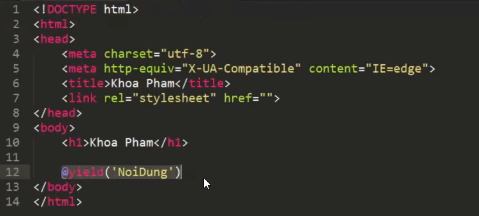
@section(‘NoiDung’)

..Bên trong đây là nội dung cần nhúng vào..

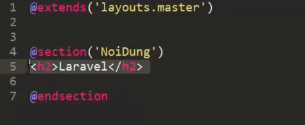
@endsection

Ví dụ:

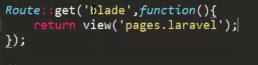
Tạo một file master.blade.php trong thư mục layouts.



Tạo một file laravel.blade.php trong thư mục pages



Gọi file laravel.blade.php trong route.



1. **Nhúng css vào blade template.**

Các file css, javascript được viết trong thư mục public.

Để nhúng file css vào blade templte, ta sử dụng thẻ link:

*<link rel=”stylesheet” href=”{{asset(‘css/style.css’)}}”>*

Hàm asset sẽ đưa chúng ta đến thư mục public, việc còn lại là phải đưa ra vị trí của file css trong thư mục public.

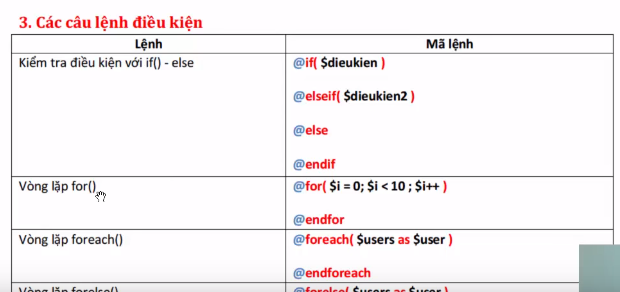
1. Hiển thị dữ liệu trên trang giao diện

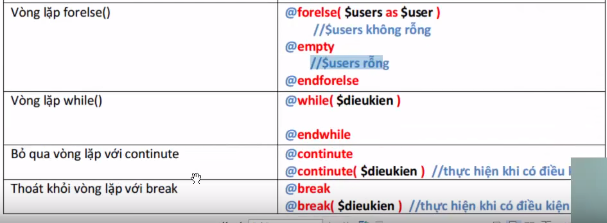


Hình 18- Hiển thị dữ liệu trên trang giao diện

Hai cặp dấu ngoặc nhọn không thể nhúng kèm các thẻ HTML, nhưng với cách còn lại thì có thể.

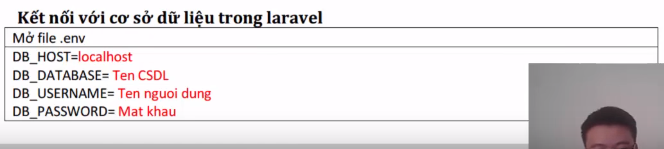
1. Câu lệnh điều kiện.





1. Cấu hình cơ sở dữ liệu

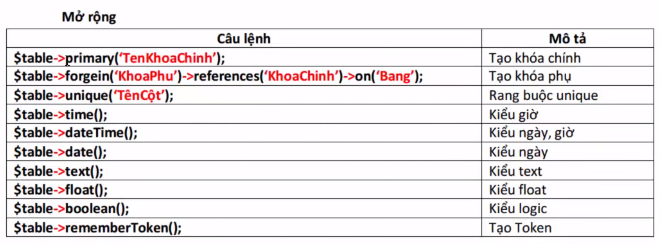




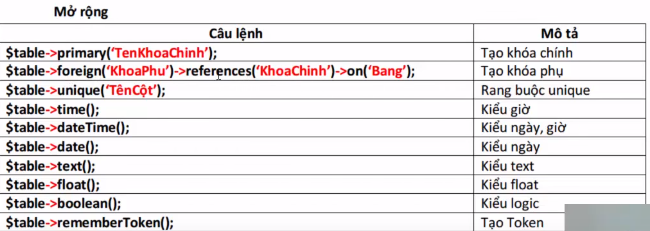
1. Tạo bảng với Schema



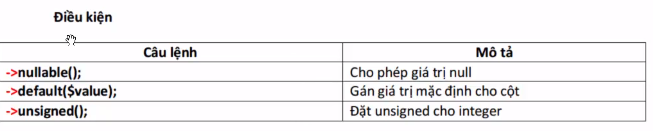
Hình 19 - Tạo bảng trong Schema



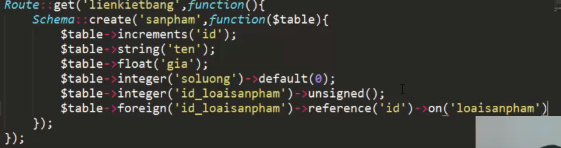
Hình 20- Tạo bảng trong Schema - mở rộng 1



Hình 21- Tạo bảng trong Schema - Mở rộng 2

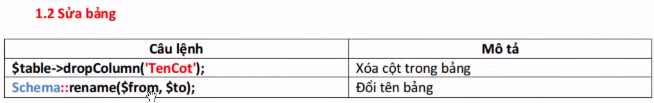


Hình 22- Tạo bảng trong Schema - Điều kiện

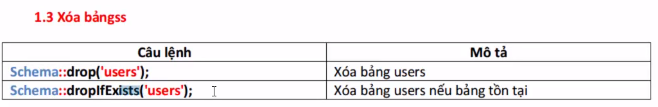


Hình 23- Ví dụ về tạo bảng bằng Schema trong Laravel

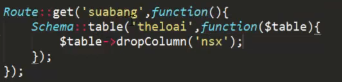
1. Chỉnh sửa bảng với Schema



Hình 24- Sửa bảng trong Schema



Hình 25- Xóa bảng trong Schema



Hình 26- Ví dụ về xóa cột trong Schema

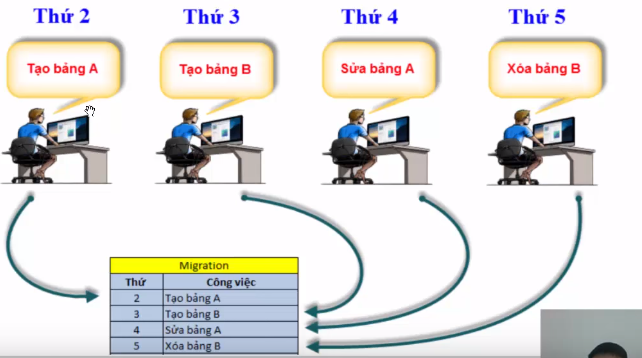


Hình 27- Ví dụ về thêm cột trong Schema

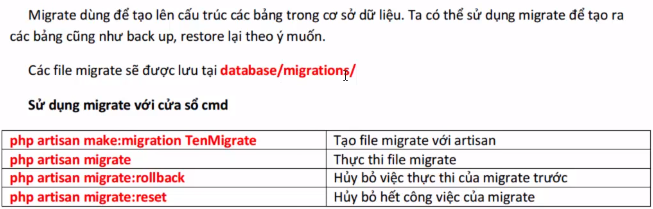


Hình 28- Ví dụ về sửa tên bảng trong Schema

1. Quản lý cơ sở dữ liệu với Migrate.

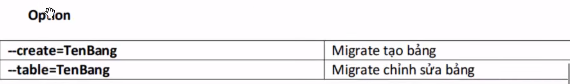


Hình 29- Mô tả về Migrate trong Laravel.



Hình 30- Cách sử dụng Migration trong Laravel

Migrate:refresh :Thực hiện rollback toàn bộ, sau đó thực thi lại toàn bộ file migration.



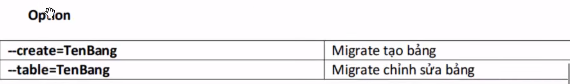
Ví dụ sử dụng option: php artisan amke:migration create\_student\_table –create=Student

Hình 31- option khi tạo migrate

1. Sử dụng Migrate để rollback dữ liệu.

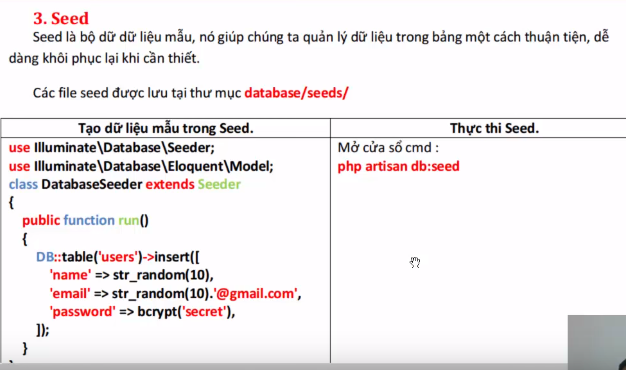
Sử dụng Migrate:rollback để rollbacl dữ liệu.

Migrate:refresh :Thực hiện rollback toàn bộ, sau đó thực thi lại toàn bộ file migration.



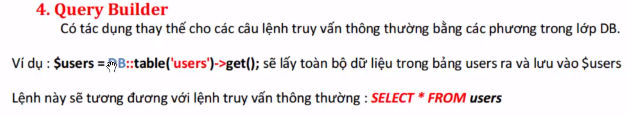
Ví dụ sử dụng option: php artisan make:migration create\_student\_table –create=Student

1. Tạo dữ liệu mẫu với Seed.

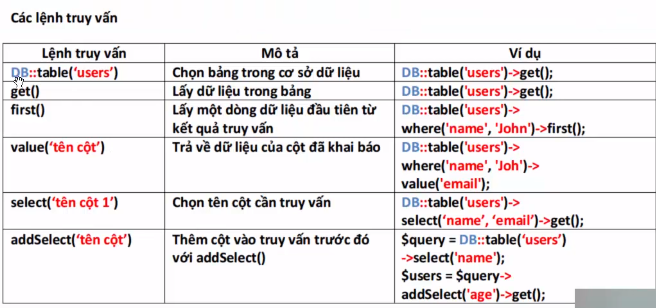


Hình 32- seed trong laravel

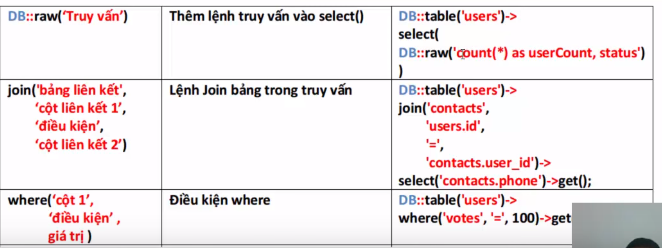
1. Truy vấn dữ liệu với Jquery Builder.



Hình 33- Cơ bản về Query Builder.



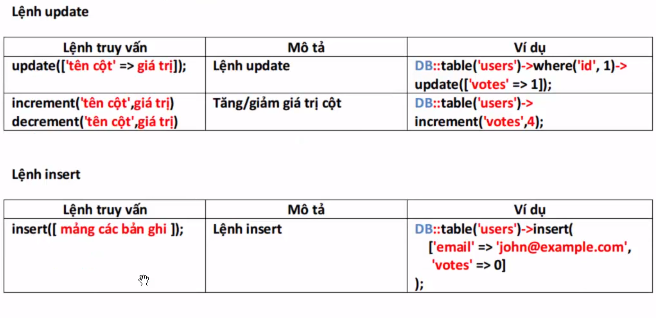
Hình 34- Các lệnh truy vấn trong laravel p1



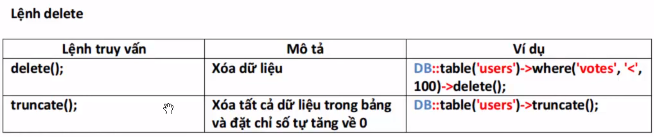
Hình 35- các lệnh truy vấn trong laravel p2



Hình 36- các lệnh truy vấn trong laravel p3



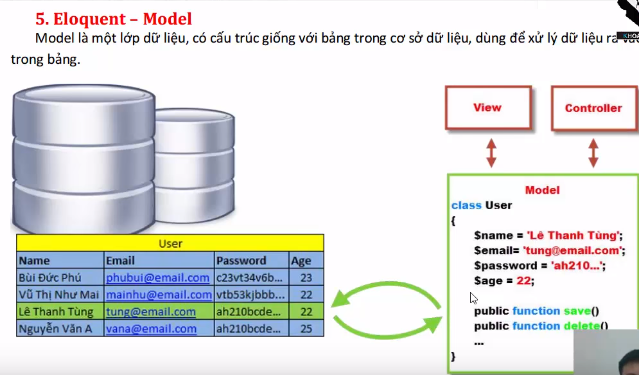
Hình 37- các lệnh truy vấn trong laravel p4



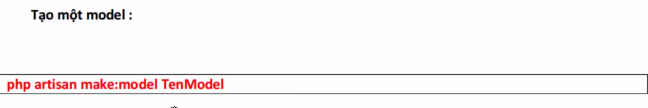
Hình 38- các lệnh truy vấn trong laravel p5

|  |  |  |
| --- | --- | --- |
| **Mục cơ bản** | **Câu truy vấn SQL** | **Truy vấn trong Laravel** |
| **Select all** | **Select \* from users** | **DB::table(‘users’)->get()** |
| **Select có điều kiện** | **Select \* from users where id=2** | **DB::table(‘users’)->where(‘id’,’=’,2)->get()** |
| **Select trường chỉ định** | **Select id, name, email from users where id=2** | **DB:table(‘users’)->select([‘id’,’name’,’email’])->where(‘id’,’=’,2)->get()** |
| **Đổi tên trường select** | **select id,name as hoten,email from users** | **DB::table(‘users’)->select(DB::raw(‘id’,‘name as hoten’,’email’))->get()** |
| **Order by** | **select \* from users order by id desc** | **DB::table(‘users’)->orderBy(‘id’,’desc’)->get()** |
| **Limit: Giới hạn trường trả về** | **Select \* from users limit 2,3** | **DB::table(‘users’)->skip(1)->take(3)->get()** |
| **count: Đếm dòng** | **select count(\*) from users** | **DB::table(‘users’)->count()** |
| **update dữ liệu** | **update users set name=’Long’ where id=2** | **DB::table(‘users’)->where(‘id’,’=’,2)->update([‘name’=>’Long’])** |
| **delete dữ liệu** | **delete from users where id=2** | **DB::table(‘’users)->where(‘id’=’2’)->delete()** |

1. Tìm hiểu về Model.



Hình 39- tìm hiểu về model

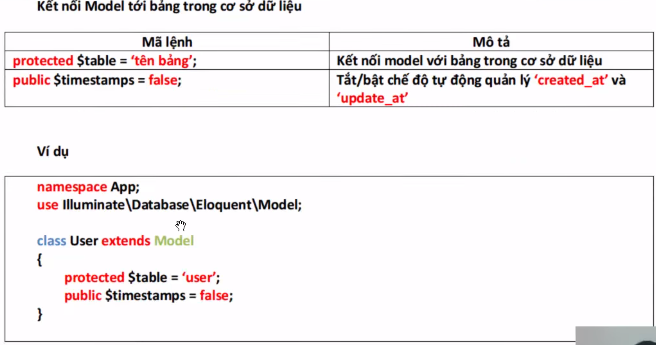


Hình 40- tạo model

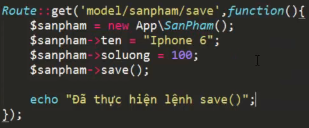


Hình 41- tạo model và một migrate tương ứng với nó

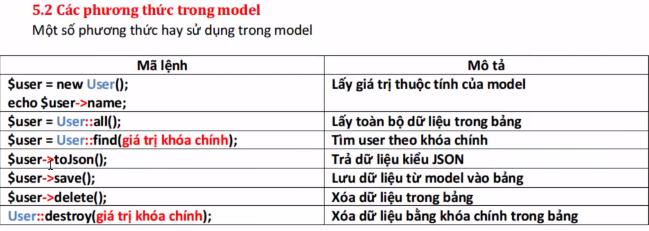
1. Truy vấn dữ liệu với Model.



Hình 42- kết nối model tới bảng trong csdl



Hình 43- ví dụ insert dữ liệu bằng model

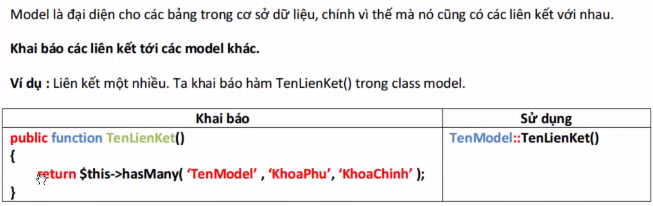


0Hình 44- các phương thức trong model

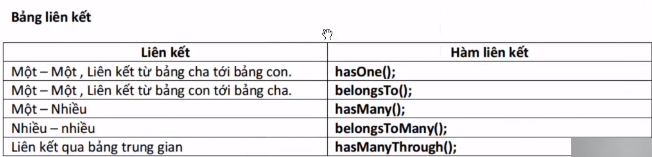


Hình 45- kết hợp model với query builder

1. Liên kết dữ liệu trong model.

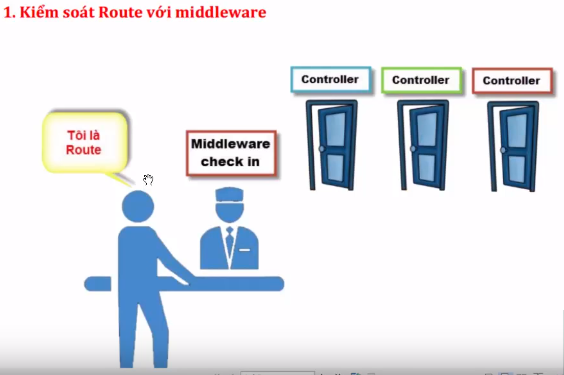


Hình 46 - Khái niệm liên kết dữ liệu trong model

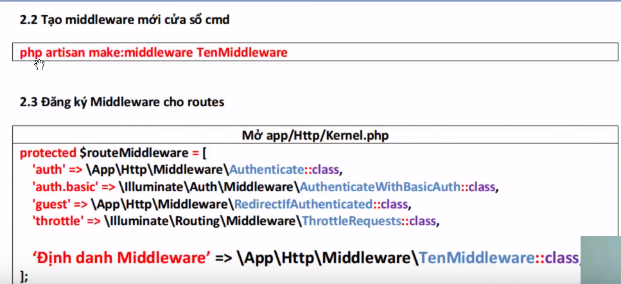


Hình 47- Bảng liên kết dữ liệu trong model

1. Bảo mật với Middleware



Hình 48- Khái niệm về Middleware trong Laravel



Hình 49 - Tạo middleware và đăng ký middleware cho route

Các bước thao tác với middleware:

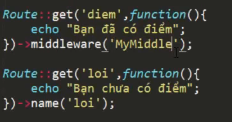
B1: Tạo middleware.

B2:Viết code xử lý trong Middleware.



B3: Đăng ký middleware trong file Kernel.php

B4: Gọi định danh Middleware khi sử dụng route.

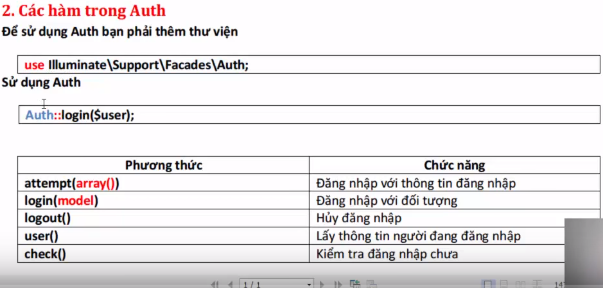


1. Tìm hiểu về Auth.

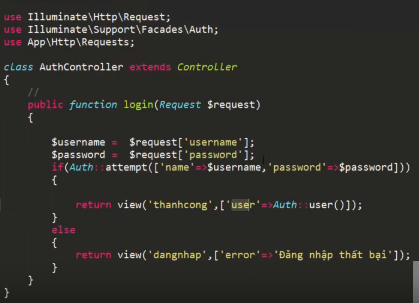


Hình 50 - Auth là gì

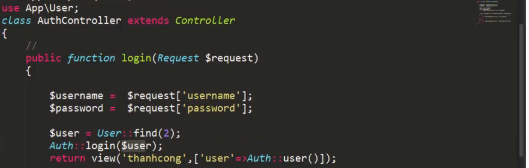
1. Quản lý đăng nhập với Auth.



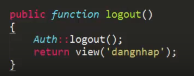
Hình 51 - Các hàm trong Auth



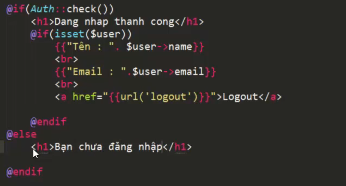
Hình 52 - Đăng nhập với hàm attempt



Hình 53 - Đăng nhập với hàm login (Đăng nhập với model User)



Hình 54 - Hàm logout



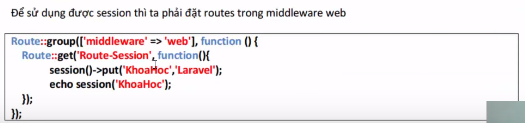
Hình 55 - Kiểm tra đăng nhập với hàm check

1. Làm việc cùng Session.

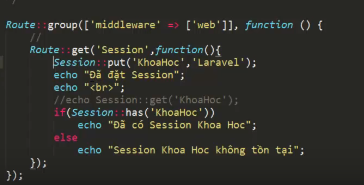
Cấu hình session.



Hình 56 - Cấu hình session

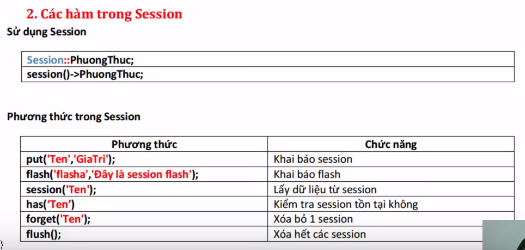


Hình 57 - Sử dụng session

**

Hình 58 - Ví dụ về sử dụng session

Các hàm trong session.

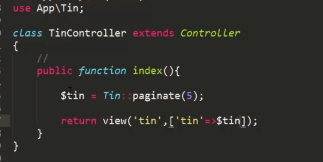


Hình 59 - Các hàm trong session

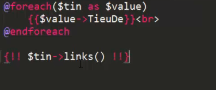
1. Phân trang với Pagination



Hình 60 - Sử dụng Paginate.

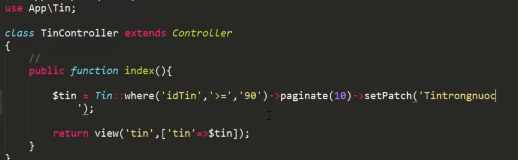


Hình 61 - Ví dụ sử dụng Paginate (Code bên phía controller)

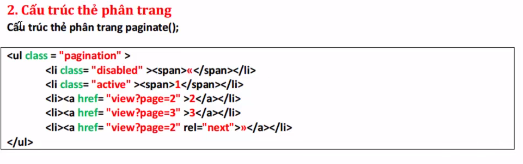


Hình 62 - Ví dụ sử dụng Paginate (Code bên phía view)

1. Chỉnh CSS trong Pagination.



Hình 63 - Sử dụng hàm setPath trong Paginatez



Hình 64 - Cấu trúc thẻ phân trang



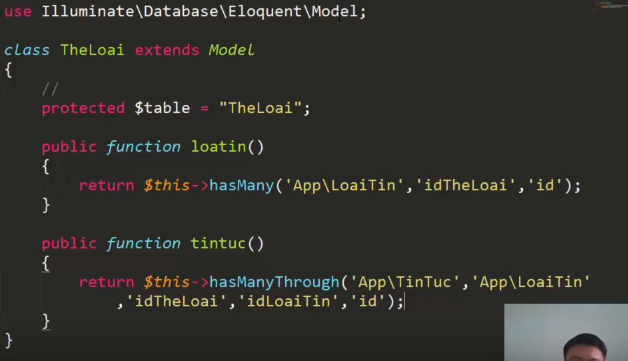
Hình 65- Cấu trúc thẻ phân trang simplePaginate

HỌC LARAVEL QUA DỰ ÁN THỰC TẾ.

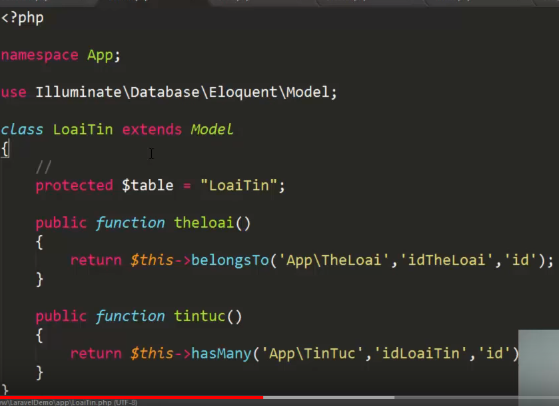
1. Phân tích cơ sở dữ liệu.
2. Tạo cơ sở dữ liệu.
3. Tạo model.

Tạo model băng câu lệnh: php artisan make:model TenModel.

Tạo liên kết giữa model với bảng dữ liệu, tạo liên kết giữa các model.



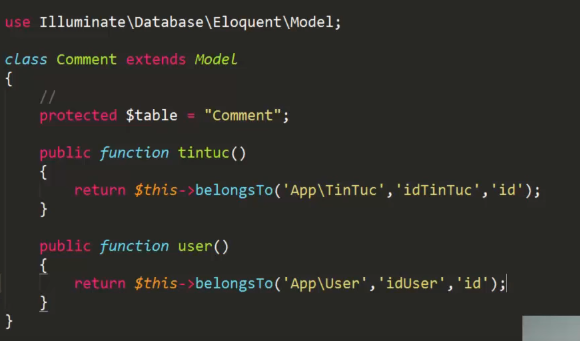
Hình - Tạo liên kết giữa các model (bảng thể loại).



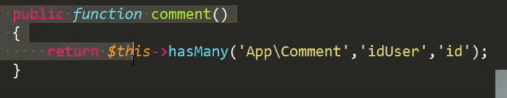
Hình - Tạo liên kết giữa các bảng (Bảng loại tin)



Hình - Tạo liên kết giữa các bảng (bảng tin tức)



Hình - Tạo liên kết giữa các model (bảng comment)

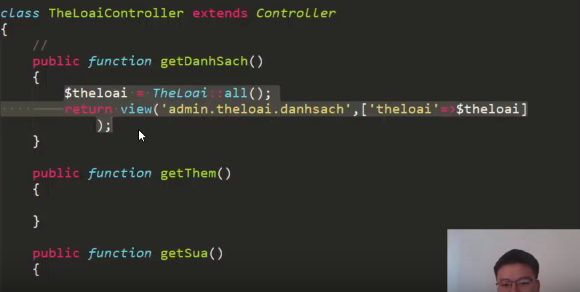


Hình - Tạo liên kết giữa các model (bảng User)

1. Tạo giao diện admin.
2. Tạo route cho admin.

Tạo route.

Tạo controller theo câu lệnh: php artisan make:controller TenController.



Hình - Tạo controller thể loại