**Cập nhật Timestamps trong Model Relation**

Trick này cho phép bạn cập nhật lại timestamps của **parent** khi **child model** được update.

Ví dụ bạn có 2 model là **Post** và **Comment** được liên kết với nhau theo quan hệ cha con, khi một **Comment** được update thì bạn cũng muốn đồng thời update thời gian đó vào **updated\_at** cho parent **Post** chẳng hạn, bạn có thể dễ dàng làm điều này bằng việc thêm thuộc tính **touches** vào **child model**

Ví dụ:

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Comment extends Model

{

protected $touches = ['post'];

public function post()

{

return $this->belongsTo('App\Post');

}

}

**Specify Eager Loading**

Khi sử dụng **Eager Loading** một model, bạn có thể chỉ ra chính xác trường dữ liệu mà bạn cần lấy của model đó:

$users = App\Book::with('author:id,name')->get();

**Authenticate một lần**

Đôi khi bạn sẽ muốn thực hiện một lần xác thực duy nhất cho một cú call API, điển hình là các stateless API thì bạn sẽ muốn sử dụng phương thức **Auth::once()** đó.

Phương thức này sẽ vẫn cho phép request đó được thực hiện, vượt Middleware nhưng sẽ không lưu lại sessions hay cookies nào hết

if (Auth::once($credentials)) {

// làm gì đó...

}

**Redirect sang Controller khác kèm tham số**

Bình thường bạn hay sử dụng phương thức **redirect()** để điều hướng URL hoặc **routes**, nhưng ngoài ra phương thức này còn có thể sử dụng để điều hướng dữ liệu qua **Controller** khác, thậm chí có thể gán thêm cả tham số vào

return redirect()->action('SomeController@method', ['param' => $value]);

**Bỏ qua Fatal Errors khi không tìm thấy Relation**

Khi bạn gọi một Mode Relation mà nó không có tồn tại, và bạn không muốn ứng dụng bị crash tóe loe trước mắt khác hàng thì hãy sử dụng **withDefault()**

public function user()

{

return $this->belongsTo('App\User')->withDefault();

}

**Biến $loop trong vòng lặp kép**

Khi sử dụng **foreach** trong **Blade Template**, bạn có thể sử dụng biến **$loop** để truy xuất đến dữ liệu của vòng lặp cha

@foreach ($users as $user)

@foreach ($user->posts as $post)

@if ($loop->parent->first)

This is first iteration of the parent loop.

@endif

@endforeach

@endforeach

**dd() with Ease**

Thay vì truyền giá trị vào **dd()** thì bạn có thể trỏ đến nó ngay tại kết thúc câu truy vấn

// Thay vì

$users = User::where('name', 'Stark')->get();

dd($users);

// làm như này

$users = User::where('name', 'Stark')->get()->dd();

**Dùng hasMany để saveMany**

Nếu trong một Model của bạn có **hasMany()** và bạn muốn lưu một lúc nhiều bản ghi vào model con thì bạn có thể sử dụng phương thức **saveMany** như dưới đây:

$post = Post::find(1);

$post->comments()->saveMany([

new Comment(['message' => 'First comment']),

new Comment(['message' => 'Second comment']),

]);

**Specify Columns trong phương thức all()**

Khi sử dụng **Model::all()**, bạn có thể chỉ ra chính xác những field mà bạn cần lấy trong phương thức này

$users = User::all(['id', 'name', 'email']);

**@auth trong Blade**

Nếu trong Blade Template bạn cần một câu lệnh điều kiện để kiểm tra **user** đã đăng nhập chưa và hiển thị một tag ra cho **user** đó, thay vì sử dụng **if else** thì bạn có thể sử dụng **@auth** như này:

// thay vì

@if(auth()->user())

//hãy làm...

@auth

<h2>User đã login</h2>

@endauth

**Xem trước Email mà không cần gửi**

Nếu bạn đang sử dụng mailables để gửi emails, bạn có thể xem trước những emails này mà không cần gửi chúng đi, khá là tiện khi phải debug nhiều lần

Route::get('/mailable', function () {

$invoice = App\Invoice::find(1);

return new App\Mail\InvoicePaid($invoice);

});

**hasMany với điều kiện cụ thể**

Trong Eloquent **hasMany()** bạn có thể lọc các bản ghi trả về với điều kiện các bản ghi child phải thỏa mãn một điều kiện gì đó

Ví dụ: Lấy ra các **Author** có số lượng sách xuất bản lớn hơn 5 quyển

// Author -> hasMany(Book::class)

$authors = Author::has('books', '>', 5)->get();

**Khôi phục nhiều bản ghi bị Soft Deleted cùng lúc**

Khi sử dụng Soft Deletes, bạn có thể khôi phục nhiều bản ghi cùng lúc thõa mản điều kiện như này

Post::withTrashed()->where('author\_id', 1)->restore();

**Migrate Columns kèm theo Time zones**

Bạn có biết là migrations không chỉ có **timestamps()** mà còn có cả **timestampsTz()** để lưu lại luôn cả time zone nữa, nếu cần thiết đừng quên sử dụng nhé

Schema::create('employees', function (Blueprint $table) {

$table->increments('id');

$table->string('name');

$table->string('email');

$table->timestampsTz();

});

**Kiểm tra xem một View có tồn tại hay không**

Như title, đơn giản là làm như này:

if (view()->exists('custom.page')) {

// Load the view

}

**Điều kiện ngày tháng trong Eloquent**

Tron Eloquent bạn có thể thực hiện các mệnh đề điều kiện kiểm tra ngày tháng trực tiếp bằng các phương thức: **whereDay()**, **whereDate()**, **whereTime()**, **whereMonth()**, **whereYear()**

$posts = Post::whereDate('created\_at', '2018-01-31')->get();

$posts = Post::whereMonth('created\_at', '12')->get();

$posts = Post::whereDay('created\_at', '31')->get();

$posts = Post::whereYear('created\_at', date('Y'))->get();

$posts = Post::whereTime('created\_at', '=', '14:13:58')->get();

**Sử dụng orderBy() ngay trong Eloquent Relationships**

Bạn có thể trực tiếp sử dụng phương thức này ngay tại relation trong Model của bạn, các bản ghi con sau này được lấy ra sẽ tự động được order theo quy định

public function posts()

{

return $this->hasMany(Post::class);

}

public function postsByName()

{

return $this->hasMany(Post::class)->orderBy('name');

}