

Laravel Task Scheduler

Task Scheduler là gì?

Trong các công việc cần thực hiện trên một hệ thống website, đôi khi một số công việc chúng ta cần thực hiện theo một kế hoạch định trước, ví dụ như chúng ta cần xóa các dữ liệu tạm trong các bảng của cơ sở dữ liệu vào giữa đêm hoặc chúng ta muốn tổng hợp các báo cáo tốn rất nhiều tài nguyên vào cuối ngày để hôm sau có thể truy cập được ngay...

Laravel Task Scheduler cho phép bạn thực hiện các công việc theo một kế hoạch đặt trước.

Thiết lập thời gian biểu

Laravel Task Scheduling cho phép bạn lập thời gian biểu để chạy thực hiện các loại ứng dụng sau:

Sử dụng phương thức call()

```
$schedule->call(function () {  
    DB::table('temp_data')->delete();  
})->daily();
```

Sử dụng phương thức command()

```
$schedule->command('emails:send --force')->daily();
```

Sử dụng phương thức exec()

```
$schedule->exec('ipconfig -all')->daily();
```

Sử dụng phương thức job()

```
use App\Jobs\Heartbeat;  
  
$schedule->job(new Heartbeat)->everyFiveMinutes();
```

Các công việc cần thực hiện và thời gian biểu cần được định nghĩa trong phương thức `schedule()` của class `App\Console\Kernel`

Thiết lập Task Output

```
$schedule->command('inspire')  
    ->everyFiveMinutes()  
    ->appendOutputTo('C:\inspire.txt');
```

Đoạn code trên sẽ ghi các câu danh ngôn từ câu lệnh `php artisan inspire` vào file `inspire.txt`

Chạy Task Scheduler

Để chạy Task Scheduler trên Server, bạn dùng câu lệnh sau

```
php artisan schedule:run
```

Tuy nhiên để câu lệnh này chạy liên tục, bạn cần thiết lập cronjob trên Server

```
* * * * * cd /path-to-your-project && php artisan schedule:run >> /dev/null 2>&1
```

Để chạy Task Scheduler trên Local bạn sử dụng câu lệnh sau

```
php artisan schedule:work
```

Các tùy chọn đặt lịch trong Task Scheduler

Phương thức	Mô tả
->cron(' ');	Đặt lịch kiểu cron job
->everyMinute();	Thực hiện công việc mỗi phút một lần
->everyFiveMinutes();	Thực hiện công việc 5 phút một lần
->everyTenMinutes();	Thực hiện công việc 10 phút một lần
->everyThirtyMinutes();	Thực hiện công việc 30 phút một lần
->hourly();	Thực hiện công việc mỗi giờ một lần
->hourlyAt(15);	Thực hiện công việc mỗi giờ một lần vào phút thứ 15
->daily();	Thực hiện công việc hàng ngày vào đúng nửa đêm
->dailyAt('13:00');	Thực hiện công việc hàng ngày vào lúc 13:00
->twiceDaily(1, 13);	Thực hiện công việc 2 lần một ngày vào lúc 1:00 và 13:00
->weekly();	Thực hiện công việc hàng tuần
->monthly();	Thực hiện công việc hàng tháng
->monthlyOn(4, '15:00');	Thực hiện công việc vào 15:00 ngày thứ 4 của tháng hàng tháng
->quarterly();	Thực hiện công việc hàng quý
->yearly();	Thực hiện công việc hàng năm
->timezone('Asia/Ho_Chi_Minh');	Thiết lập thời gian từng vùng

Ví dụ:

```
$schedule->exec('ipconfig')
    ->weekdays()
    ->hourly()
    ->timezone('Asia/Ho_Chi_Minh')
    ->between('8:00', '17:00');
```

Ví dụ trên sẽ thực hiện câu lệnh ipconfig (câu lệnh xem các thông tin về địa chỉ IP trên máy) từ 8:00 đến 17:00 hàng giờ vào các ngày cuối tuần với múi giờ là giờ Việt Nam.

Thiết lập điều kiện thực hiện lịch

Phương thức	Mô tả
->weekdays();	Giới hạn thực hiện lịch chỉ vào ngày cuối tuần
->sundays();	Giới hạn thực hiện lịch chỉ vào Chủ nhật
->mondays();	Giới hạn thực hiện lịch chỉ vào thứ Hai
->tuesdays();	Giới hạn thực hiện lịch chỉ vào thứ Ba
->wednesdays();	Giới hạn thực hiện lịch chỉ vào thứ Tư
->thursdays();	Giới hạn thực hiện lịch chỉ vào thứ Năm
->fridays();	Giới hạn thực hiện lịch chỉ vào thứ Sáu
->saturdays();	Giới hạn thực hiện lịch chỉ vào thứ Bảy
->between(start, end);	Giới hạn thực hiện lịch ở giữa khoảng thời gian từ start đến end
->when(Closure);	Giới hạn thực hiện lịch chỉ khi nào Closure trả về true.

Laravel cho phép bạn thực hiện các điều kiện phức tạp trong thực hiện lịch công việc với việc sử dụng phương thức when và skip bằng cách truyền vào một Closure. Với phương thức when() nếu Closure trả về true thì lịch sẽ được thực hiện, phương thức skip() thì ngược lại, nếu Closure trả về true thì lịch sẽ được bỏ qua

```
$schedule->call(function () {
    DB::table('temp_users')->delete();
})->daily()
->when(function () {
    return (DB::table('logging_user')->get() != null)?false:true;
});
```

Ví dụ trên sẽ xóa bảng tạm chứa thông tin người dùng hàng ngày nếu tại thời điểm đó không có người dùng nào đang trong hệ thống.