

Course - Laravel Framework

Các Contract

Các "contract" của Laravel là một tập hợp các interface xác định các service cốt lõi được cung cấp bởi framework.

Tags: laravel contract, laravel

Giới thiệu

Các "contract" của Laravel là một tập hợp các interface xác định các service cốt lõi được cung cấp bởi framework. Ví dụ, một contract `Illuminate\Contracts\Queue\Queue` sẽ xác định các phương thức cần thiết cho các công việc sắp xếp hàng chờ, trong khi contract `Illuminate\Contracts\Mail\Mailer` sẽ xác định các phương thức cần thiết để gửi e-mail.

Mỗi contract có một cách thực thi tương ứng được cung cấp bởi framework. Ví dụ: Laravel thực thi hàng chờ với nhiều driver khác nhau và triển khai bưu phẩm được hỗ trợ bởi `SwiftMailer`.

Tất cả các contract Laravel đều nằm trong kho lưu trữ GitHub của riêng họ. Điều này cung cấp một điểm tham chiếu nhanh cho tất cả các contract có sẵn, cũng như một package thống nhất, được tách biệt có thể được sử dụng khi xây dựng các package tương tác với các service của Laravel.

Contract Vs Facade

Các facade và hàm tiện ích của Laravel sẽ cung cấp phương pháp đơn giản để sử dụng các dịch vụ của Laravel mà không cần phải khai kiểu và resolve các contract ra khỏi service container. Trong hầu hết các trường hợp, mỗi facade sẽ có một contract tương đương.

Không giống như các facade, không yêu cầu bạn require chúng trong constructor của class, các contract cho phép bạn khai báo các thư viện một cách minh bạch cho các class của mình. Một số developer thích khai báo minh bạch các thư viện của họ theo cách này và do đó thích sử dụng contract, trong khi các developer khác thích sự tiện lợi của các facade. Nói chung, hầu hết các ứng dụng có thể sử dụng các facade mà không gặp bất kỳ vấn đề gì trong quá trình phát triển.

Khi nào sử dụng contract (thích thì dùng)

Quyết định sử dụng contract hoặc facade sẽ phụ thuộc vào sở thích cá nhân và thị hiếu của nhóm developer của bạn. Cả contract và facade đều có thể được sử dụng để tạo các ứng dụng Laravel mạnh mẽ, đã được thử nghiệm tốt. Contract và Facade không loại trừ lẫn nhau. Một số phần trong ứng dụng của bạn có thể sử dụng facade trong khi những phần khác phụ thuộc vào contract. Miễn là bạn tập trung vào trách nhiệm của lớp mình, bạn sẽ nhận thấy rất ít sự khác biệt thực tế giữa việc sử dụng contract và facade.

Nói chung, hầu hết các ứng dụng có thể sử dụng các facade mà không gặp vấn đề gì trong quá trình phát triển. Nếu bạn đang xây dựng một package tích hợp với nhiều framework

PHP, bạn có thể muốn sử dụng package **illuminate/contracts** để xác định sự tích hợp của bạn với các service của Laravel mà không cần yêu cầu sự thực thi cụ thể nào của Laravel trong *composer.json* của bạn.

Cách sử dụng contract

Vì vậy, làm thế nào để bạn có được một bản thực thi của một contract? Nó thực sự khá đơn giản.

Nhiều loại class trong Laravel được trả lại thông qua service container, bao gồm driver, event listener, middleware, các task được xếp hàng chờ và thậm chí cả route function. Vì vậy, để thực thi một contract, bạn có thể chỉ cần khai kiểu cho interface trong constructor của class đang được trả lại.

Ví dụ: hãy xem event listener này:

```
<?php

namespace App\Listeners;

use App\Events\OrderWasPlaced;
use App\Models\User;
use Illuminate\Contracts\Redis\Factory;

class CacheOrderInformation
{
    /**
     * The Redis factory implementation.
     *
     * @var \Illuminate\Contracts\Redis\Factory
     */
    protected $redis;

    /**
     * Create a new event handler instance.
     *
     * @param \Illuminate\Contracts\Redis\Factory $redis
     * @return void
     */
}
```

```

    */
    public function __construct(Factory $redis)
    {
        $this->redis = $redis;
    }

    /**
     * Handle the event.
     *
     * @param  \App\Events\OrderWasPlaced  $event
     * @return void
     */
    public function handle(OrderWasPlaced $event)
    {
        //
    }
}

```

Khi chương trình xử lý event được trả về, service container sẽ đọc các khai báo kiểu trên constructor của class và đưa vào giá trị thích hợp. Để tìm hiểu thêm về cách đăng ký mọi thứ trong service container, hãy xem tài liệu của nó.

Tham chiếu contract

Bảng này cung cấp một tham chiếu nhanh đến tất cả các contract Laravel và các giao diện tương đương của chúng:

Contract	Facade tương ứng
Illuminate\Contracts\Auth\Access\Authorizable	
Illuminate\Contracts\Auth\Access\Gate	Gate
Illuminate\Contracts\Auth\Authenticatable	
Illuminate\Contracts\Auth\CanResetPassword	
Illuminate\Contracts\Auth\Factory	Auth

Contract	Facade tương ứng
Illuminate\Contracts\Auth\Guard	Auth::guard()
Illuminate\Contracts\Auth\PasswordBroker	Password::broker()
Illuminate\Contracts\Auth\PasswordBrokerFactory	Password
Illuminate\Contracts\Auth\StatefulGuard	
Illuminate\Contracts\Auth\SupportsBasicAuth	
Illuminate\Contracts\Auth\UserProvider	
Illuminate\Contracts\Bus\Dispatcher	Bus
Illuminate\Contracts\Bus\QueueingDispatcher	Bus::dispatchToQueue()
Illuminate\Contracts\Broadcasting\Factory	Broadcast
Illuminate\Contracts\Broadcasting\Broadcaster	Broadcast::connection()
Illuminate\Contracts\Broadcasting\ShouldBroadcast	
Illuminate\Contracts\Broadcasting\ShouldBroadcastNow	
Illuminate\Contracts\Cache\Factory	Cache
Illuminate\Contracts\Cache\Lock	
Illuminate\Contracts\Cache\LockProvider	
Illuminate\Contracts\Cache\Repository	Cache::driver()
Illuminate\Contracts\Cache\Store	
Illuminate\Contracts\Config\Repository	Config
Illuminate\Contracts\Console\Application	
Illuminate\Contracts\Console\Kernel	Artisan

Contract	Facade tương ứng
Illuminate\Contracts\Container\Container	App
Illuminate\Contracts\Cookie\Factory	Cookie
Illuminate\Contracts\Cookie\QueueingFactory	Cookie::queue()
Illuminate\Contracts\Database\ModelIdentifier	
Illuminate\Contracts\Debug\ExceptionHandler	
Illuminate\Contracts\Encryption\Encrypter	Crypt
Illuminate\Contracts\Events\Dispatcher	Event
Illuminate\Contracts\Filesystem\Cloud	Storage::cloud()
Illuminate\Contracts\Filesystem\Factory	Storage
Illuminate\Contracts\Filesystem\Filesystem	Storage::disk()
Illuminate\Contracts\Foundation\Application	App
Illuminate\Contracts\Hashing\Hasher	Hash
Illuminate\Contracts\Http\Kernel	
Illuminate\Contracts\Mail\MailQueue	Mail::queue()
Illuminate\Contracts\Mail\Mailable	
Illuminate\Contracts\Mail\Mailer	Mail
Illuminate\Contracts\Notifications\Dispatcher	Notification
Illuminate\Contracts\Notifications\Factory	Notification
Illuminate\Contracts\Pagination\LengthAwarePaginator	
Illuminate\Contracts\Pagination\Paginator	

Contract	Facade tương ứng
Illuminate\Contracts\Pipeline\Hub	
Illuminate\Contracts\Pipeline\Pipeline	
Illuminate\Contracts\Queue\EntityResolver	
Illuminate\Contracts\Queue\Factory	Queue
Illuminate\Contracts\Queue\Job	
Illuminate\Contracts\Queue\Monitor	Queue
Illuminate\Contracts\Queue\Queue	Queue::connection()
Illuminate\Contracts\Queue\QueueableCollection	
Illuminate\Contracts\Queue\QueueableEntity	
Illuminate\Contracts\Queue\ShouldQueue	
Illuminate\Contracts\Redis\Factory	Redis
Illuminate\Contracts\Routing\BindingRegistrar	Route
Illuminate\Contracts\Routing\Registrar	Route
Illuminate\Contracts\Routing\ResponseFactory	Response
Illuminate\Contracts\Routing\UrlGenerator	URL
Illuminate\Contracts\Routing\UrlRoutable	
Illuminate\Contracts\Session\Session	Session::driver()
Illuminate\Contracts\Support\Arrayable	
Illuminate\Contracts\Support\Htmlable	
Illuminate\Contracts\Support\Jsonable	
Illuminate\Contracts\Support\MessageBag	

Contract	Facade tương ứng
Illuminate\Contracts\Support\MessageProvider	
Illuminate\Contracts\Support\Renderable	
Illuminate\Contracts\Support\Responsable	
Illuminate\Contracts\Translation\Loader	
Illuminate\Contracts\Translation\Translator	Lang
Illuminate\Contracts\Validation\Factory	Validator
Illuminate\Contracts\Validation\ImplicitRule	
Illuminate\Contracts\Validation\Rule	
Illuminate\Contracts\Validation\ValidatesWhenResolved	
Illuminate\Contracts\Validation\Validator	Validator::make()
Illuminate\Contracts\View\Engine	
Illuminate\Contracts\View\Factory	View
Illuminate\Contracts\View\View	View::make()