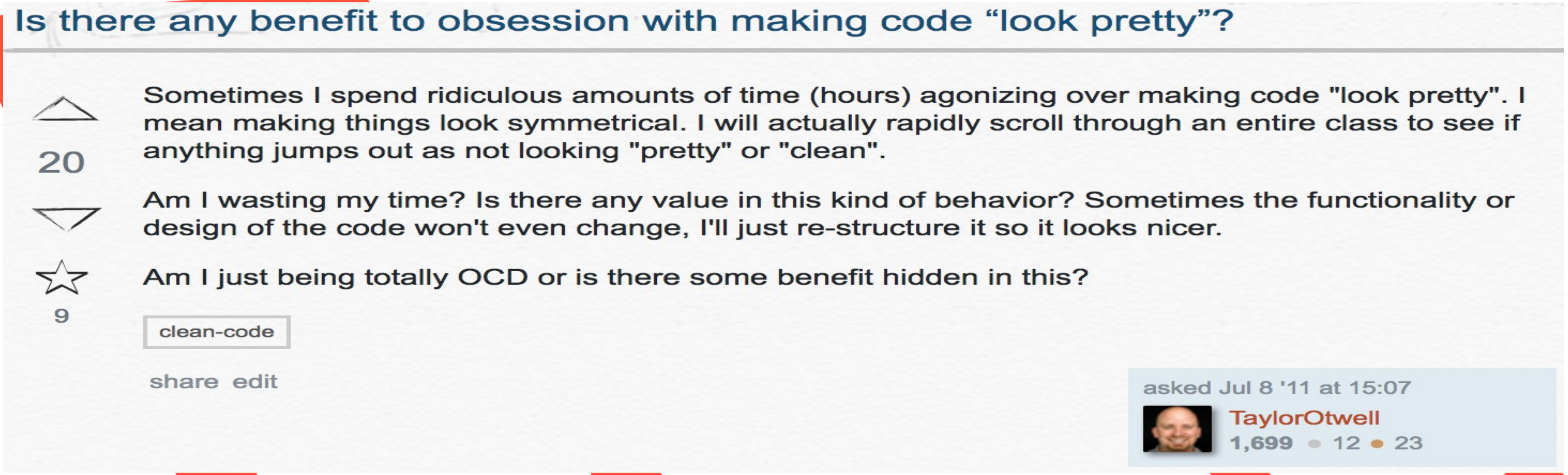
Mesut Vatansever - Software Developer

Github | @mvatansever
Twitter | @mesutsoft

What We'll Cover?

- What is Laravel?
- Shortly Laravel history
- Laravel Ecosystem
- Laravel Architecture
- How is work Laravel basically?
- Laravel Ecosystem tools like as Homestead
- Laravel toolset such as Eloquent, Cache, Artisan
- Build simple application works with RabbitMQ

What Is Laravel?



Is there any benefit to obsession with making code “look pretty”?

20

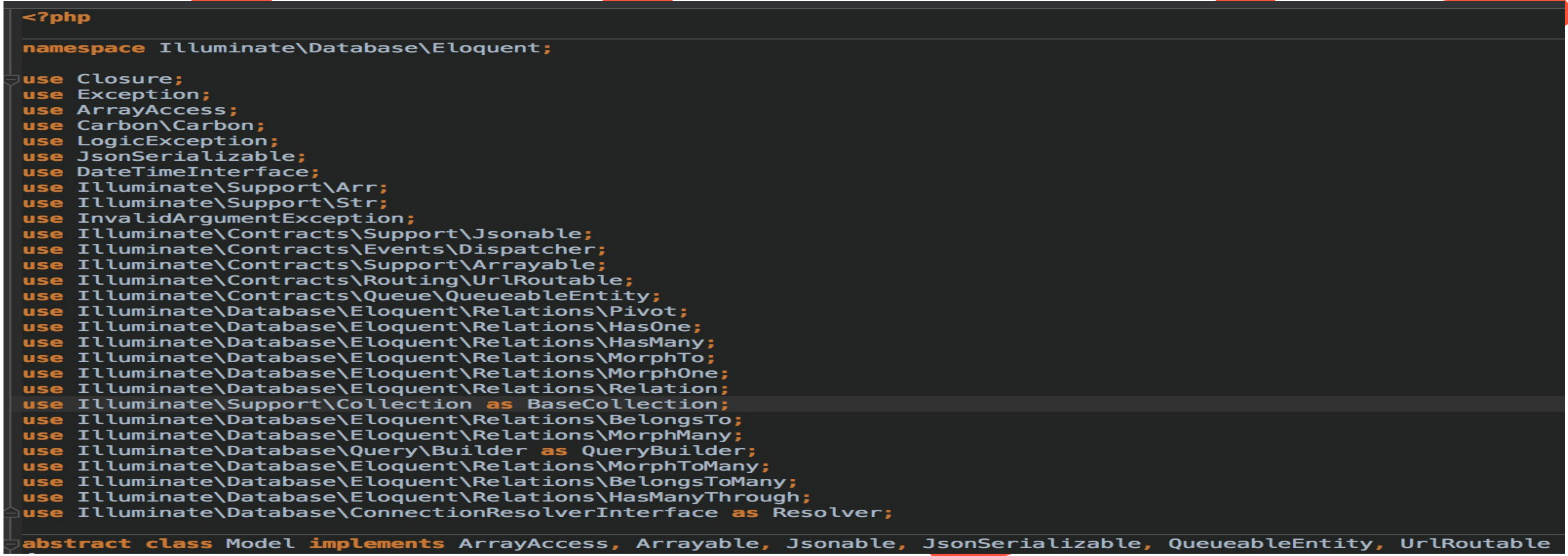
Sometimes I spend ridiculous amounts of time (hours) agonizing over making code "look pretty". I mean making things look symmetrical. I will actually rapidly scroll through an entire class to see if anything jumps out as not looking "pretty" or "clean".

Am I wasting my time? Is there any value in this kind of behavior? Sometimes the functionality or design of the code won't even change, I'll just re-structure it so it looks nicer.

Am I just being totally OCD or is there some benefit hidden in this?

9

asked Jul 8 '11 at 15:07
TaylorOtwell
1,699 ● 12 ● 23

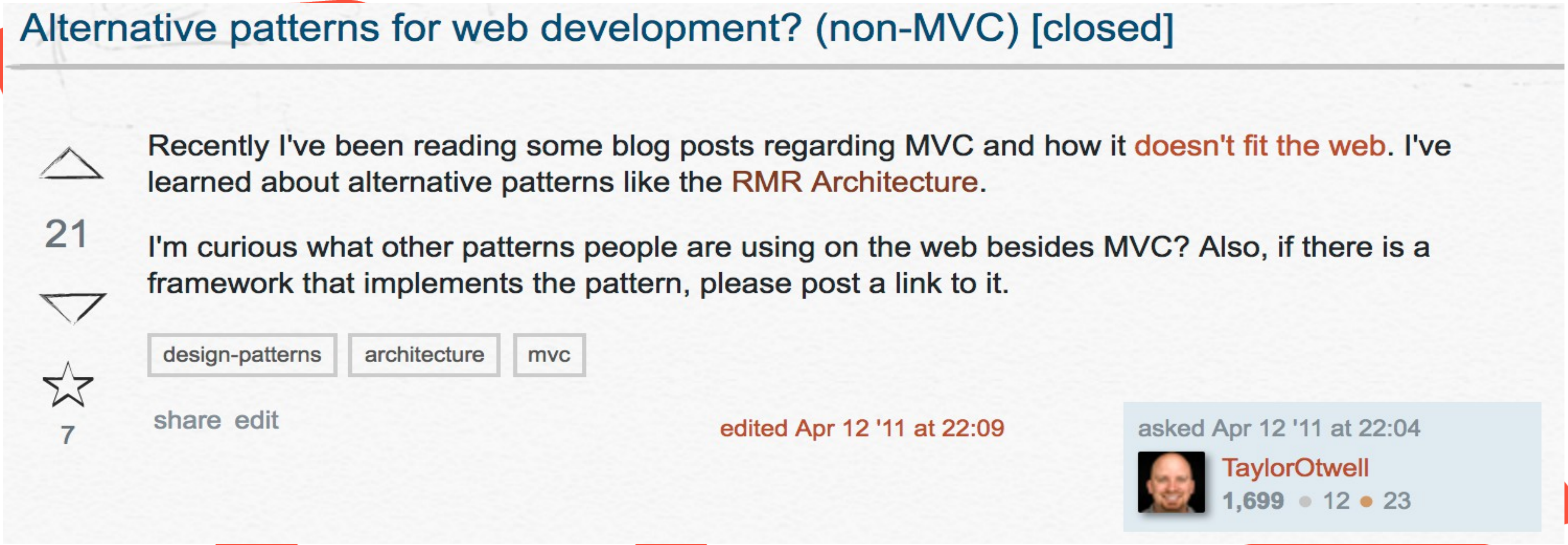


```php
<?php

namespace Illuminate\Database\Eloquent;

use Closure;
use Exception;
use ArrayAccess;
use Carbon\Carbon;
use LogicException;
use JsonSerializable;
use DateTimeInterface;
use Illuminate\Support\Arr;
use Illuminate\Support\Str;
use InvalidArgumentException;
use Illuminate\Contracts\Support\Jsonable;
use Illuminate\Contracts\Events\Dispatcher;
use Illuminate\Contracts\Support\Arrayable;
use Illuminate\Contracts\Routing\UrlRoutable;
use Illuminate\Contracts\Queue\QueueableEntity;
use Illuminate\Database\Eloquent\Relations\Pivot;
use Illuminate\Database\Eloquent\Relations\HasOne;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\MorphTo;
use Illuminate\Database\Eloquent\Relations\MorphOne;
use Illuminate\Database\Eloquent\Relations\Relation;
use Illuminate\Support\Collection as BaseCollection;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\MorphMany;
use Illuminate\Database\Query\Builder as QueryBuilder;
use Illuminate\Database\Eloquent\Relations\MorphToMany;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
use Illuminate\Database\Eloquent\Relations\HasManyThrough;
use Illuminate\Database\ConnectionResolverInterface as Resolver;

abstract class Model implements ArrayAccess, Arrayable, Jsonable, JsonSerializable, QueueableEntity, UrlRoutable
```

Laravel History

- Taylor Otwell researching how to develop a web app
- Laravel version 1 beta was released on June 9, 2011
- Solve the growing pains of using CodeIgniter PHP framework.
- Out of box Authentication
- Eloquent ORM
- Cache
- Routes
- Less than six months Laravel version 2 (MVC structure, IoC, Blade)

More Information:
https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/



Alternative patterns for web development? (non-MVC) [closed]

21

7

Recently I've been reading some blog posts regarding MVC and how it doesn't fit the web. I've learned about alternative patterns like the RMR Architecture.

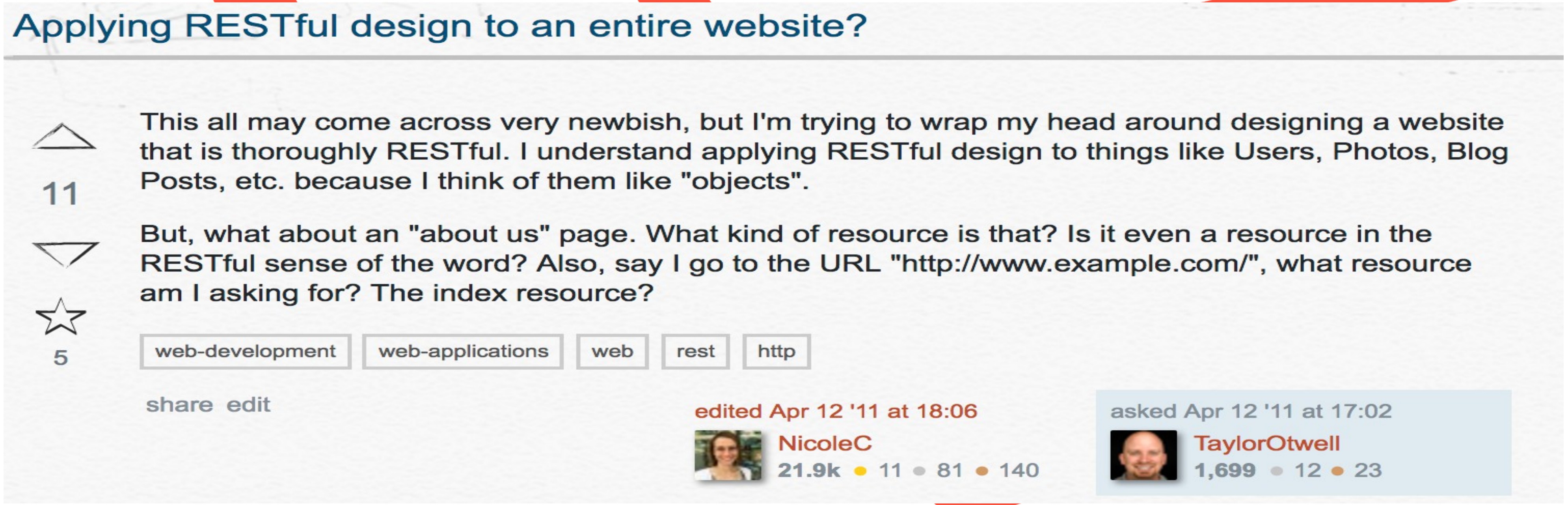
I'm curious what other patterns people are using on the web besides MVC? Also, if there is a framework that implements the pattern, please post a link to it.

design-patterns architecture mvc

share edit

edited Apr 12 '11 at 22:09

asked Apr 12 '11 at 22:04
TaylorOtwell
1,699 ● 12 ● 23



Applying RESTful design to an entire website?

11

5

This all may come across very newbish, but I'm trying to wrap my head around designing a website that is thoroughly RESTful. I understand applying RESTful design to things like Users, Photos, Blog Posts, etc. because I think of them like "objects".

But, what about an "about us" page. What kind of resource is that? Is it even a resource in the RESTful sense of the word? Also, say I go to the URL "http://www.example.com/", what resource am I asking for? The index resource?

web-development web-applications web rest http

share edit

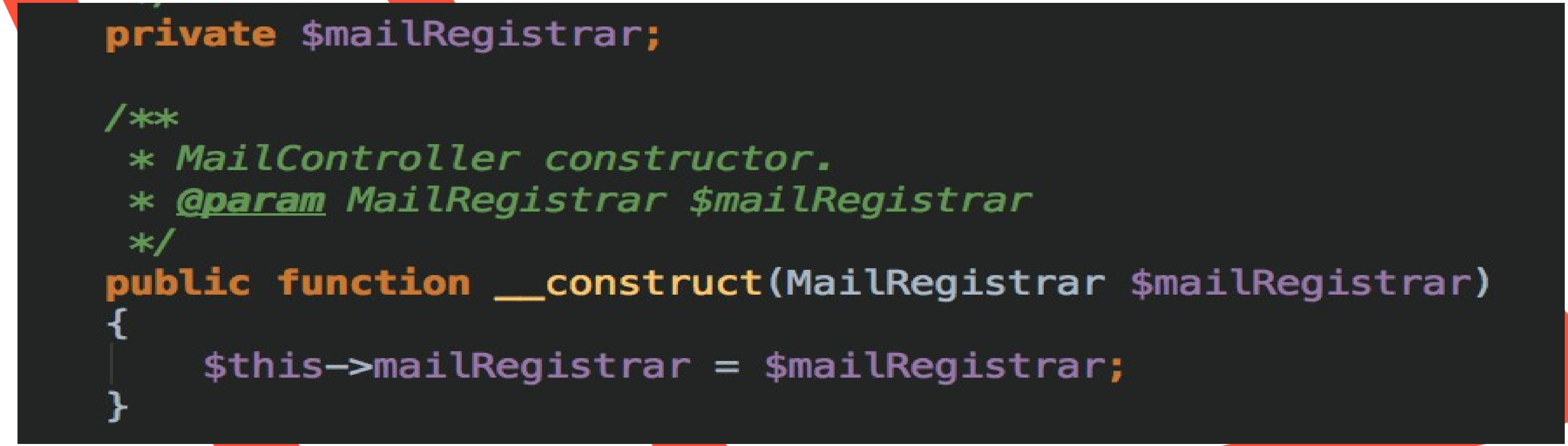
edited Apr 12 '11 at 18:06
NicoleC
21.9k ● 11 ● 81 ● 140

asked Apr 12 '11 at 17:02
TaylorOtwell
1,699 ● 12 ● 23

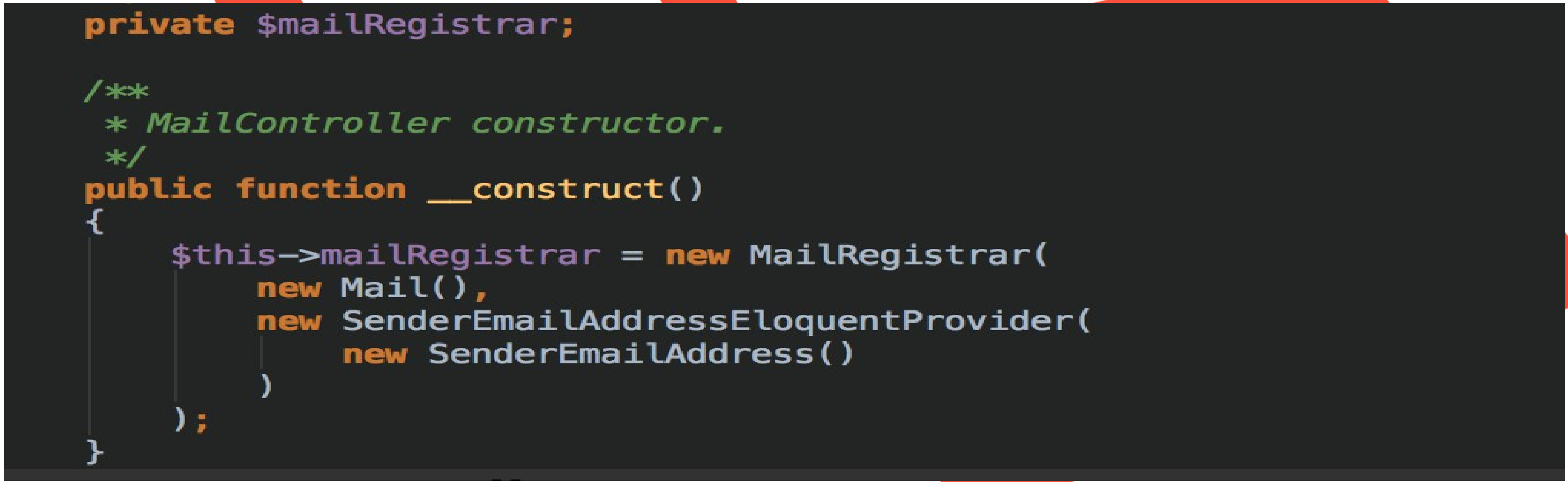
Laravel Ecosystem

- Laracasts daily video Laravel/PHP/Programming videos
- Laravel-news, new developments in Laravel
- Homestead (Powered by Vagrant)
- Laracon & Laracon EU, annual conferences about Laravel
- Lumen, micro-framework
- Official Packages (Passport, Scout etc..)
- Unofficial packages (Laravel Collective)
- LaraDock (Beautiful dockerize of web development)

Default Dependency Injection



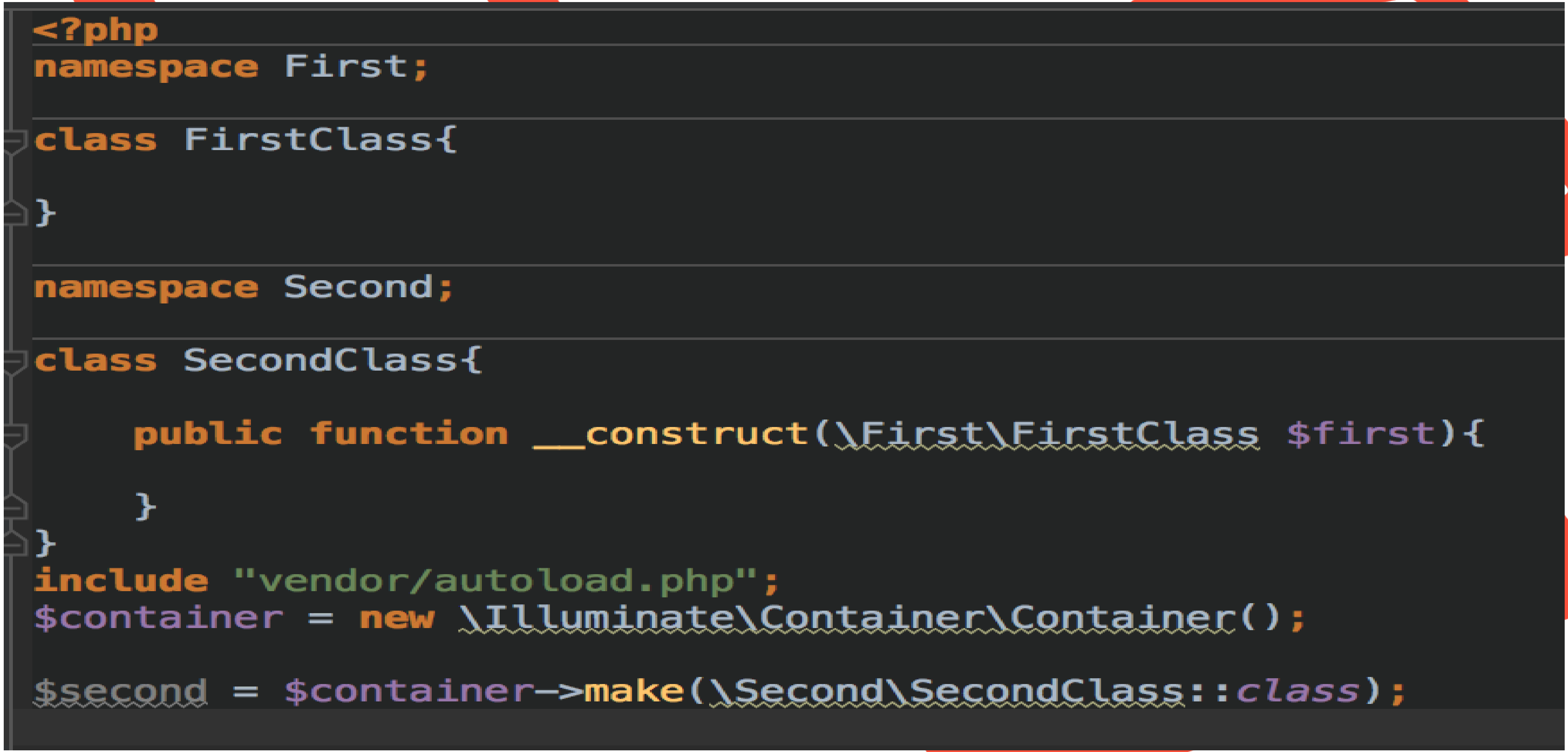
```php
private $mailRegistrar;

/**
 * MailController constructor.
 * @param MailRegistrar $mailRegistrar
 */
public function __construct(MailRegistrar $mailRegistrar)
{
    $this->mailRegistrar = $mailRegistrar;
}
```



```php
private $mailRegistrar;

/**
 * MailController constructor.
 */
public function __construct()
{
    $this->mailRegistrar = new MailRegistrar(
        new Mail(),
        new SenderEmailAddressEloquentProvider(
            new SenderEmailAddress()
        )
    );
}
```

Laravel Container

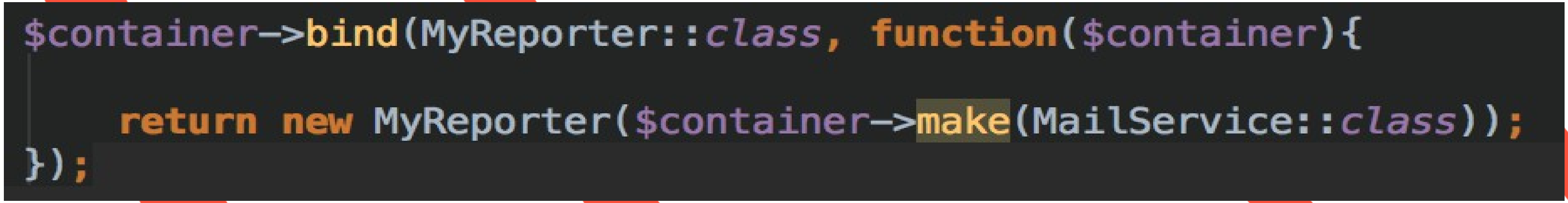
You may add your project with:
composer require illuminate/container



```php
<?php
namespace First;

class FirstClass{

}

namespace Second;

class SecondClass{

    public function __construct(\First\FirstClass $first){

    }
}
include "vendor/autoload.php";
$container = new \Illuminate\Container\Container();

$second = $container->make(\Second\SecondClass::class);
```

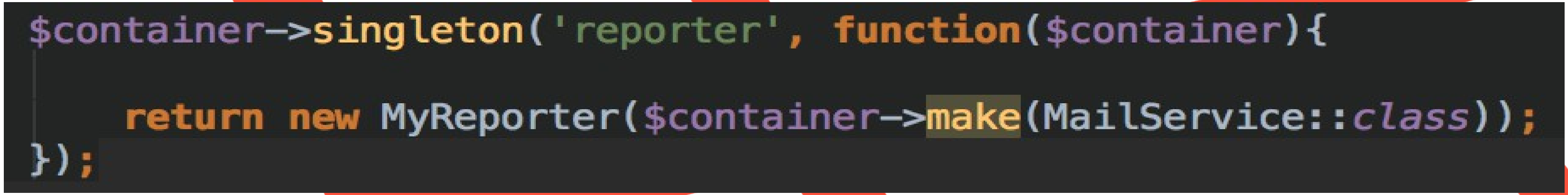
Container Binding

Closure Binding

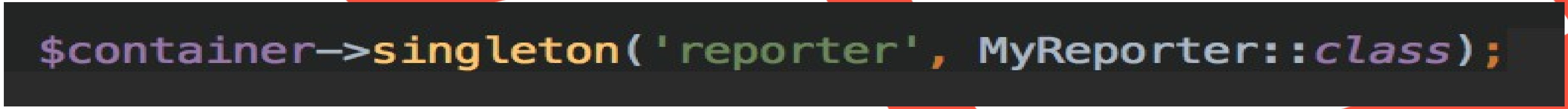


```php
$container->bind(MyReporter::class, function($container){

    return new MyReporter($container->make(MailService::class));
});
```

Singleton Binding

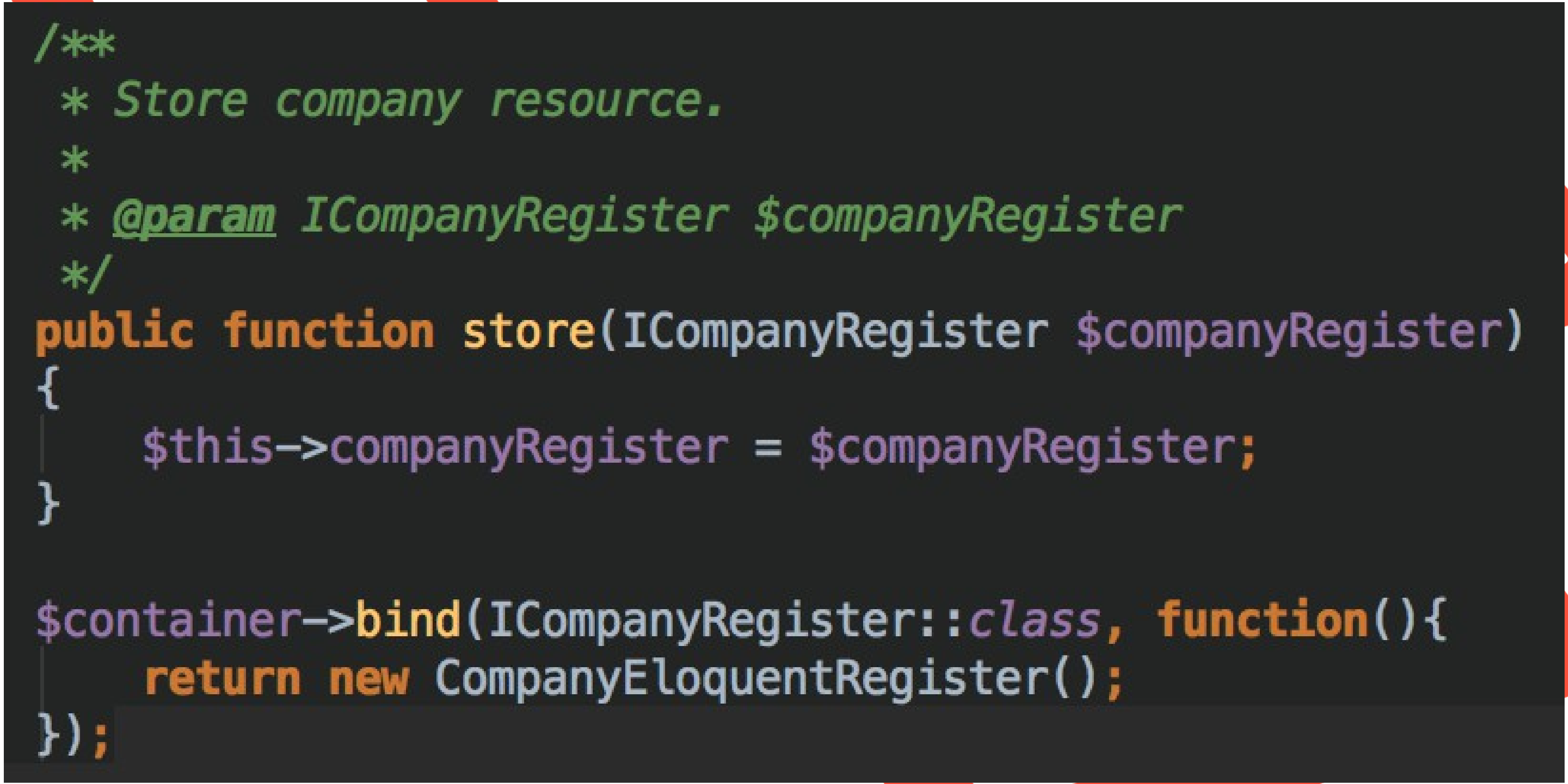


```php
$container->singleton('reporter', function($container){

    return new MyReporter($container->make(MailService::class));
});
```



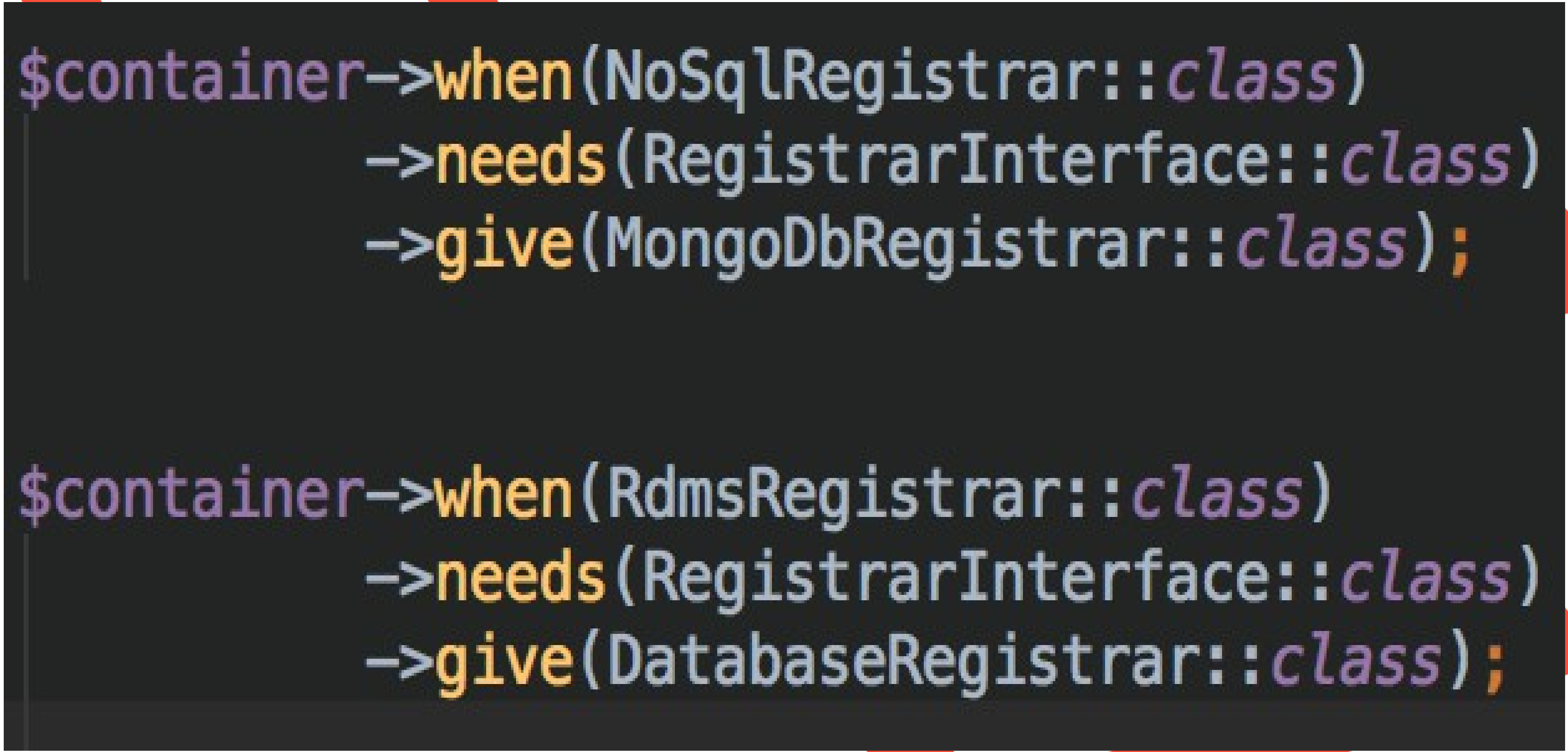
```php
$container->singleton('reporter', MyReporter::class);
```

Interface Binding



```
/**
 * Store company resource.
 *
 * @param ICompanyRegister $companyRegister
 */
public function store(ICompanyRegister $companyRegister)
{
    $this->companyRegister = $companyRegister;
}

$container->bind(ICompanyRegister::class, function(){
    return new CompanyEloquentRegister();
});
```

Contextual Binding

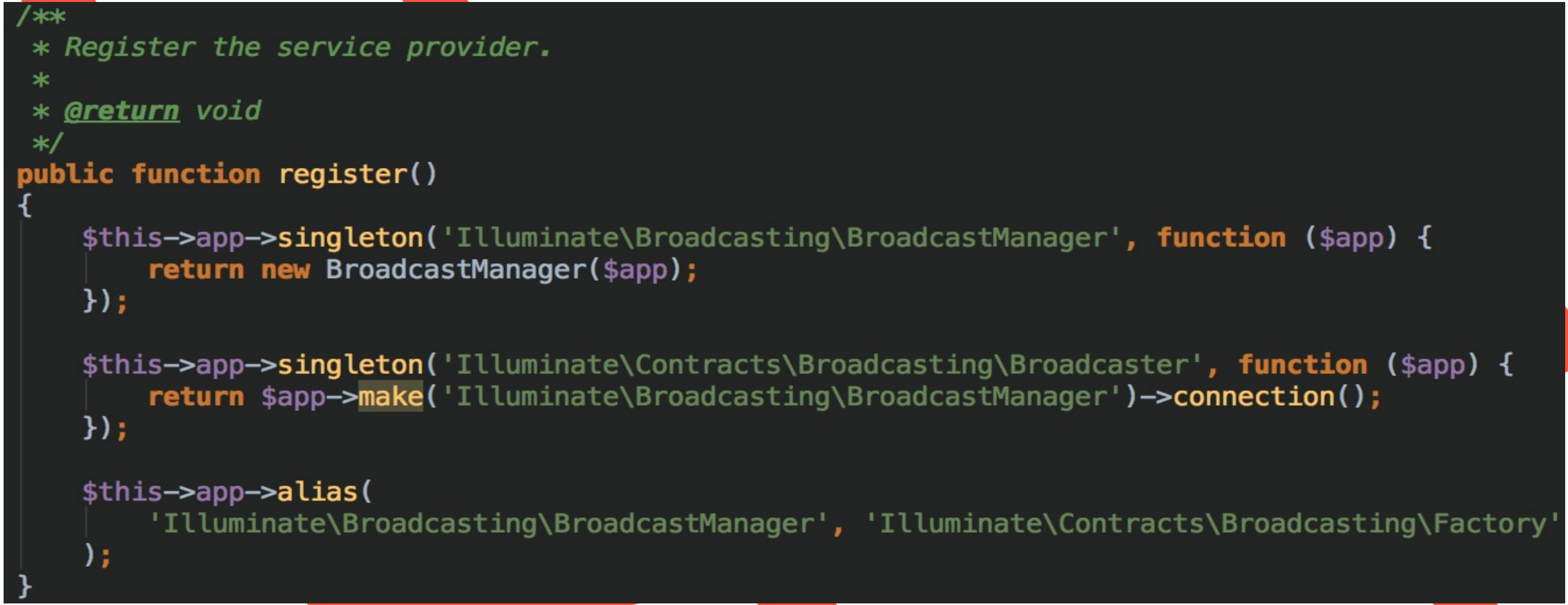


```
$container->when(NoSqlRegistrar::class)
                ->needs(RegistrarInterface::class)
                ->give(MongoDbRegistrar::class);


$container->when(RdmsRegistrar::class)
                ->needs(RegistrarInterface::class)
                ->give(DatabaseRegistrar::class);
```

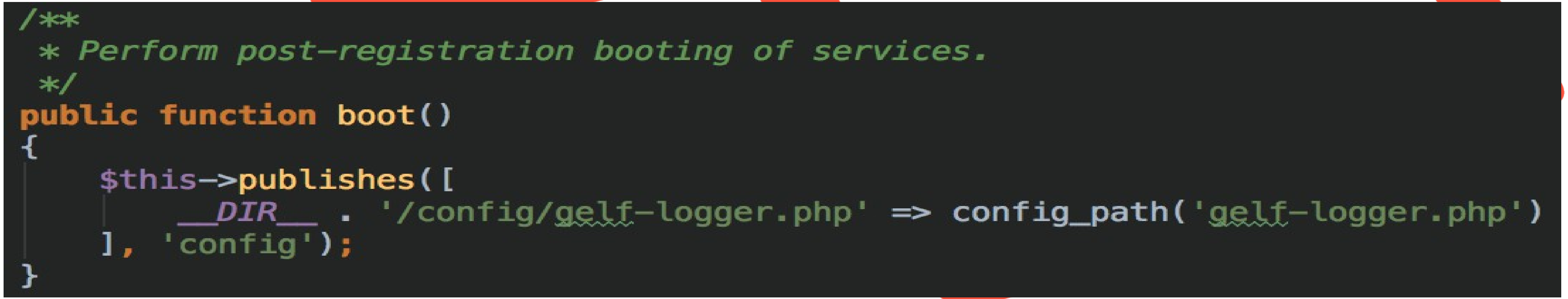
Service Provider

- Bootstrapping the all of Laravel core services via Service Providers such as Mail, Queue, Database etc.
- In the **config/app.php** file included **providers** array you might see all of the services of our application.
- Providing advantage for binding services to Service Container by grouping.
- May deferred instantiating classes until needed it.

* The Service Provider class must be extend the **Illuminate\Support\ServiceProvider** class.



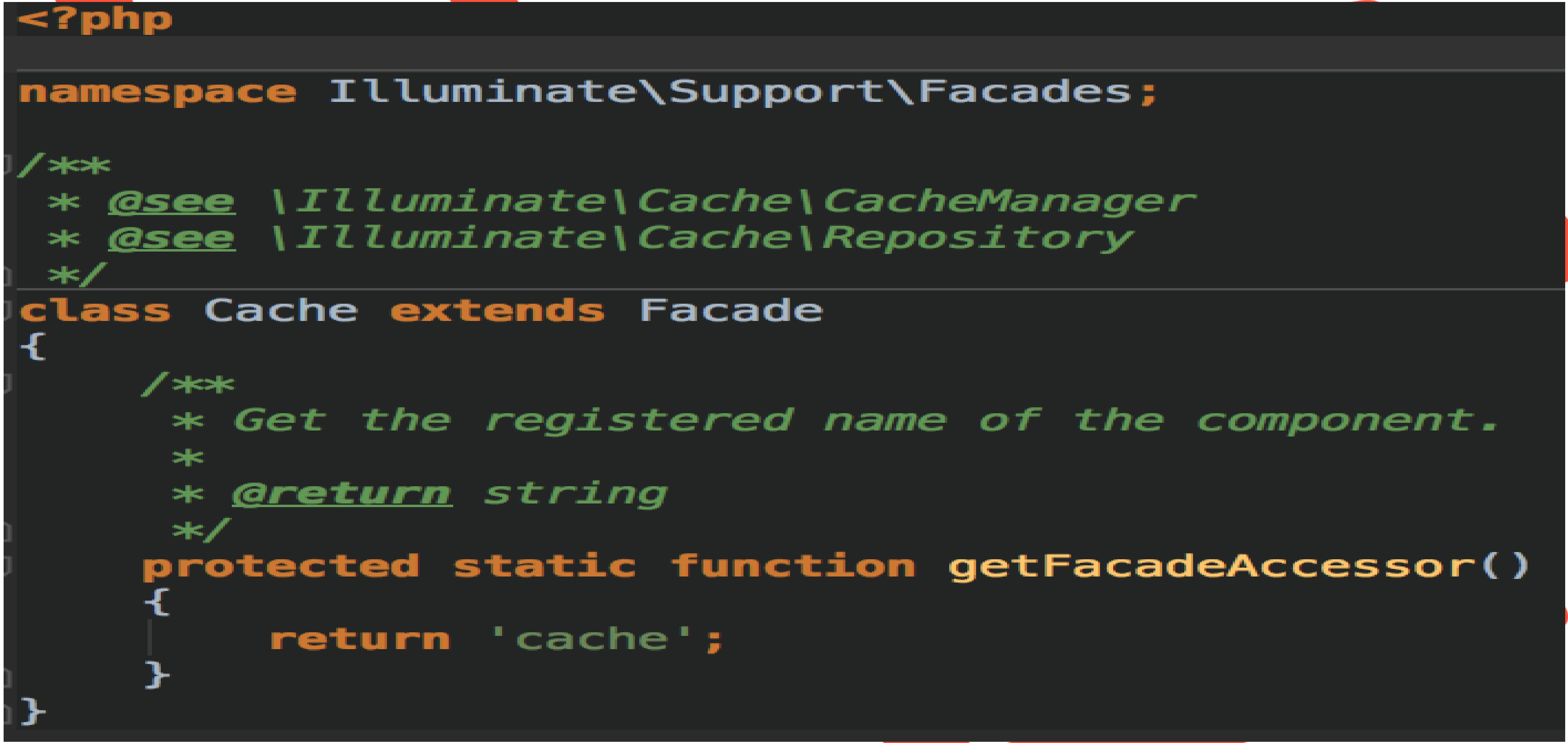
```php
/**
 * Register the service provider.
 *
 * @return void
 */
public function register()
{
    $this->app->singleton('Illuminate\Broadcasting\BroadcastManager', function ($app) {
        return new BroadcastManager($app);
    });

    $this->app->singleton('Illuminate\Contracts\Broadcasting\Broadcaster', function ($app) {
        return $app->make('Illuminate\Broadcasting\BroadcastManager')->connection();
    });

    $this->app->alias(
        'Illuminate\Broadcasting\BroadcastManager', 'Illuminate\Contracts\Broadcasting\Factory'
    );
}
```



```php
/**
 * Perform post-registration booting of services.
 */
public function boot()
{
    $this->publishes([
        __DIR__ . '/config/gelf-logger.php' => config_path('gelf-logger.php')
    ], 'config');
}
```

Facades

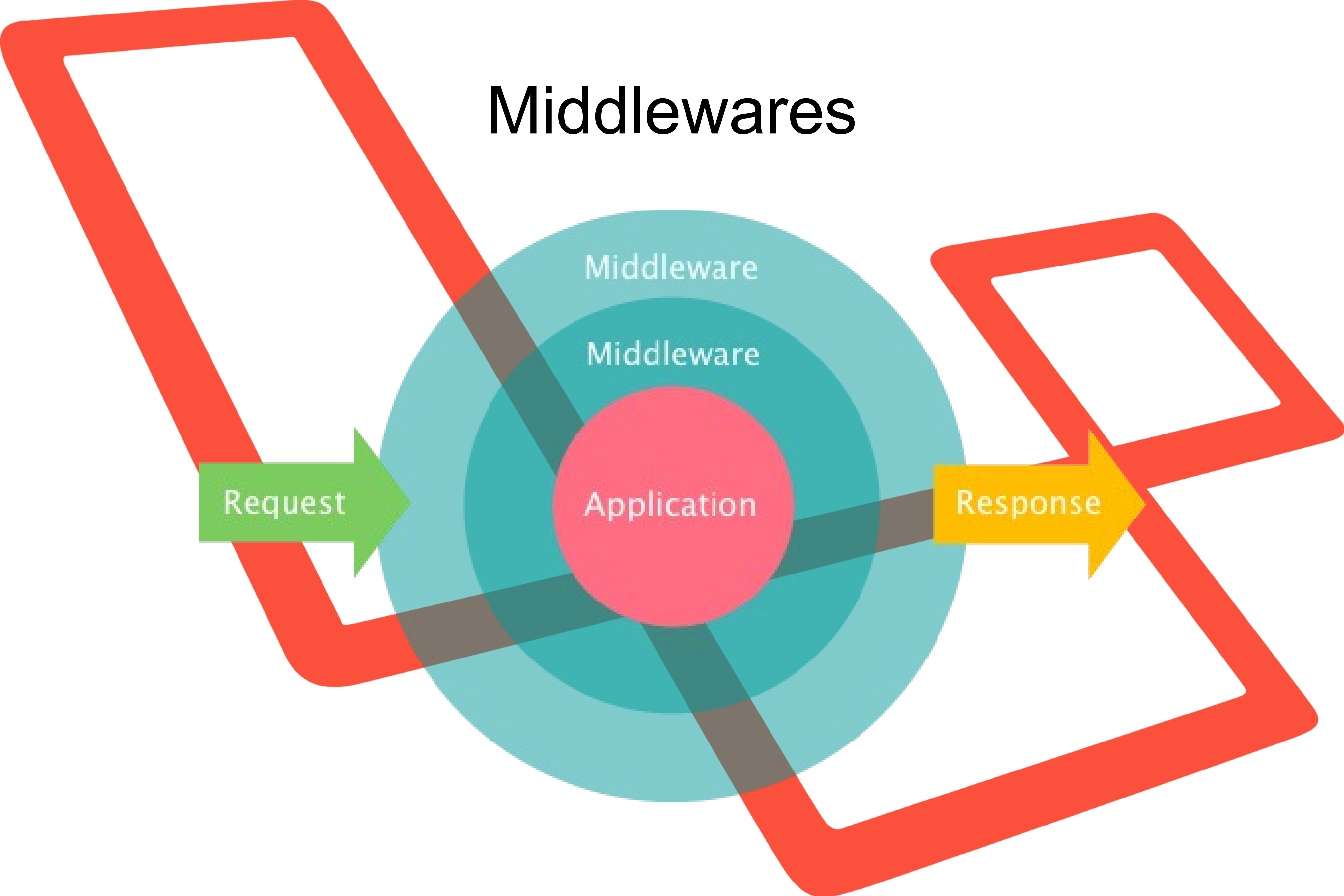
- It's only a class to providing “static proxies” to class in the Service Container.
- Almost all of the Laravel's features provide a Facade. Example: Cache, Event etc.



```php
<?php

namespace Illuminate\Support\Facades;

/**
 * @see \Illuminate\Cache\CacheManager
 * @see \Illuminate\Cache\Repository
 */
class Cache extends Facade
{
    /**
     * Get the registered name of the component.
     *
     * @return string
     */
    protected static function getFacadeAccessor()
    {
        return 'cache';
    }
}
```

Laravel built-in Facades: https://laravel.com/docs/5.4/facades#facade-class-reference

Middlewares



Talk is cheap.

Show me the code.

Linus Torvalds