

**Decode Web**[FOLLOW](#)

Senior Laravel developer with 5+ years of experience

Laravel service providers explained in depth

Published Jan 21, 2020



[Laravel](#) is completely a mystery because even if you are an experienced developer you will toil hard to learn its core functionality. One of the functionalities is, its **Service Providers**, on which I am going to give a detailed overview. These are really simpler than you have imagined, I hope so.. Therefore lets bootstrap..

What are service providers in laravel?

Service providers in laravel application is the central place where application is bootstrapped. That is, laravel's core services and our application's services, classes and their dependencies are injected in service container through providers.

More simpler definition may be given as, providers are funnel/nozzle through which we pour fuel called classes into a fuel tank called service container of an engine called Laravel.

For example, open config/app.php, and find an array with name "providers"

**Enjoy this post?**

2

1

```
'providers' => [

    /*
     * Laravel Framework Service Providers...
     */
    Illuminate\Auth\AuthServiceProvider::class,
    Illuminate\Broadcasting\BroadcastServiceProvider::class,
    Illuminate\Bus\BusServiceProvider::class,
    Illuminate\Cache\CacheServiceProvider::class,
    Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
    Illuminate\Cookie\CookieServiceProvider::class,
    .
    .
    .
],
```

As you can see, these are some of the service providers come along with laravel itself or you can say they are core services which are injected into service container.

Service provider prevents multiple instances of classes

Yes, Imagine you have created a class which needs multiple dependencies & basically you write it like this:

```
$foo = new Foo(new Bar(config('some_key')), new Baz(new Moo(), new Boo())
```

It is 100% correct way to write like this but not a smart one because each time when you use class Foo you will have to write complete syntax by initializing each internal classes too. As a developer/programmer we are deemed to be smart one technically so why follow an ancient way ? Therefore this situation is prevented if we use service provider.

How to create service provider in laravel ?

You might know, Laravel provides an artisan command to create service provider.

```
| php artisan make:provider MyServiceProvider |
```



Enjoy this post?

♡ 2

💬 1

This command will create a service provider at **App/Providers/** directory with name as MyServiceProvider. As a laravel convention we append ServiceProvider with class name whenever a new provider class is created so that we can just get to know that this particular file is of service provider type easily.

Each provider class extends base ServiceProvider class.

Example

```
<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

class MyServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     *
     * @return void
     */
    public function register()
    {
        //
    }

    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        //
    }
}
```

Now you can see, above class has only two major functions **register()** and **boot()**. Let us see each of them individually.



Enjoy this post?

♡ 2

💬 1

It is very important to know the fact that `register()` method allows us to define bindings to our service container. For instance, see code below.

```
public function register()
{
    $this->app->singleton(my_class, function ($app) {
        return new MyClass($app);
    });
}
```

`$this->app` : This is a global variable available in laravel through which we can access internal settings of the application.

`singleton` : It is a trait, taken from Ruby module. When used this trait, we tell our application that whatever class it is given as parameter, it should have only one instance through out the application. Here MyClass will be resolved once and will have only one instance which can be accessed via my_class. Thus, it's unnecessary to instantiate multiple copies of this class.

The boot method

Next important method is `boot()`. In `boot()`, we can access all the services which are registered in `register()` method, and most importantly, view composer is *registered* in boot method only.



Enjoy this post?

♡ 2

💬 1

```
<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;
Illuminate\Support\Facades\View;

class MyServiceProvider extends ServiceProvider
{
    .
    .
    .
    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        View::composer(
            'dummy', 'App\Http\ViewComposers\MyComposer'
        );
    }
}
```

Or we can explicitly bind a model.

```
public function boot()
{
    parent::boot();

    Route::model('user', App\User::class);
}
```

How to register a service provider and why ?

Now that we have created a provider, we now require it to be registered, why?



Enjoy this post?

♥ 2

💬 1

Head directly to config/app.php and find providers array, as we already seen at the start of this post, add this line

```
'providers' => [  
  
    /*  
     * Application Service Providers...  
     */  
    App/Providers/MyServiceProvider::class  
]
```

Thus every time laravel boots **MyServiceProvider** will load automatically but after laravel 5.7 we are no more required add service in **app.php** because **composer** automatically loads services now but this is about packages and we are just making an internal service provider here hence we need to add explicitly.

Conclusion

In this article, I discussed what are service providers in laravel and its architecture, methods: boot() and register() then finally how to register them in app.php. I hope this article helped you along the journey to learn Laravel, do share your view, comments below.

Thank you !

Laravel

PHP

Object orientated programming

Laravel service provider

Backend development

Enjoy this post? Give **Decode Web** a like if it's helpful.

♡ 2

💬 1

🔗 SHARE

Decode



Enjoy this post?

♡ 2

💬 1

Decode Web

Senior Laravel developer with 5+ years of experience

I am a professional software developer and tech blogger. Writes beautiful code and blogs on technologies and programming.

[FOLLOW](#)

 1 Reply



Leave a reply



Jacques Amar 2 years ago



There is NOTHING in this article that explains what are service providers. Just showing usage and a vague re-stating from the documentation is useless. The title mentions "explained" and no explanation is given.

 Reply

Find a Pair Programming Partner on Codementor

Want to improve your programming skills? Choose from 10,000+ mentors to pair program with.

[GET STARTED](#)

Enjoy this post?

 2

 1