

Illuminate (Provider)

이제 Illuminate Provider 클래스를 생성합니다.

1. 1.1

```
namespace App\Providers;

use Illuminate\Support\ServiceProvider;
use App\Services\GreetingServiceInterface;
use App\Services\GreetingService;

class AppServiceProvider extends ServiceProvider
{
    public function register()
    {
        $this->app->bind(GreetingServiceInterface::class, GreetingService::class);
    }

    public function boot()
    {
        //
    }
}
```

1. 1.2

1. Illuminate Provider 클래스를 생성합니다.

```
namespace App\Providers;
```

- App\Providers namespace에 GreetingServiceInterface를 등록합니다. GreetingServiceInterface를 GreetingService로 바인딩합니다.

2. Illuminate Provider 클래스를 생성합니다.

```
use Illuminate\Support\ServiceProvider;
use App\Services\GreetingServiceInterface;
```

```
use App\Services\GreetingService;
```

- `Illuminate\Support\ServiceProvider`: `Illuminate\Support\ServiceProvider` 是 `Illuminate\Contracts\Support\ServiceProvider` 的接口实现。它定义了 `register` 和 `boot` 两个方法。
- `App\Services\GreetingServiceInterface` 和 `App\Services\GreetingService`: `GreetingServiceInterface` 是 `GreetingService` 的接口，定义了 `greet` 方法。

3. **AppServiceProvider** 类

```
class AppServiceProvider extends ServiceProvider
```

- `AppServiceProvider` 是 `Illuminate\Support\ServiceProvider` 的子类。它实现了 `register` 和 `boot` 方法。

4. **register** 方法

```
public function register()
{
    $this->app->bind(GreetingServiceInterface::class, GreetingService::class);
}
```

- `register` 方法用于注册服务。它调用了 `$this->app->bind` 方法，将 `GreetingServiceInterface` 绑定到 `GreetingService`。
- `$this->app->bind(GreetingServiceInterface::class, GreetingService::class);`:
 - `GreetingServiceInterface` 是接口类。
 - `GreetingService` 是实现接口的类。
 - `GreetingServiceInterface::class` 是接口的类名。
 - `GreetingService::class` 是实现类的类名。

5. **boot** 方法

```
public function boot()
{
    //
}
```

- `boot` 方法用于在应用启动时执行一些操作。它调用了 `$this->app->boot` 方法。
- `$this->app->boot` 方法调用了 `Illuminate\Contracts\Support\ServiceProvider` 的 `boot` 方法。



`AppServiceProvider` 类实现了 `GreetingServiceInterface` 接口。它调用了 `GreetingService` 类，并调用了 `GreetingServiceInterface` 接口的 `greet` 方法。

Q1: 为什么需要 `register` 和 `boot` 方法？

Q2: 为什么需要 `GreetingServiceInterface` 接口？

Q3: 为什么需要 `boot` 方法？

You wanna more detailed information?

Revision #1

Created 4 August 2024 00:41:12 by [name]

Updated 4 August 2024 00:41:28 by