```
<!-- View stored in resources/views/greeting.blade.php -->
                    <html>
                     <body>
                       <h1>Hello, {{ $name }}</h1>
                      </body>
                    </html>
                    #router/webs.php
                    Route::get('/', function () {
                      return view('greeting', ['name' => 'James']);
simple
                                              if your view is stored at resources/
                                              views/admin/profile.blade.php, you
                                              may reference it like so:
                  view文件存储位置
                                              return view('admin.profile', $data);
                                                   use Illuminate\Support\Facades\View;
                                                   if (View::exists('emails.customer')) {
                  判断view文件是否存在
                     return view('greetings', ['name' => 'Victoria']);
                     return view('greeting')->with('name', 'Victoria');
                      多个数据的时候使用多个with
                      return view('movie')->with('pages',12)-
                       >with('hemmo','asda');
传输数据
                      在所有的view中共享数据
                                                                class AppServiceProvider extends
                                                                ServiceProvider
                      you should place calls to share
                      within a service provider's boot
                                                                   public function boot()
                      method. You are free to add
                      them to the AppServiceProvider
                                                                    View::share('key', 'value');
                      or generate a separate service
                      provider to house them
```

View::creator('profile', 'App\Http

\ViewCreators\ProfileCreator');

**View Creators** 

View(resources/views directory)

```
首先创建 ServiceProvider
                                                                                                                       <?php
                                                                                                                        namespace App\Providers;
                                                                                                                       use Illuminate\Support\Facades\View;
                                                                                                                        use Illuminate\Support\ServiceProvider;
                                                                                                                        class ComposerServiceProvider extends ServiceProvider
                                                                                                                           * Register bindings in the container.
                                                                                                                          * @return void
                                                                                                                         public function boot()
                            View composers are callbacks or class methods that
                                                                                                                            // Using class based composers...
                            are called when a view is rendered. If you have data
                                                                                                                            View::composer(
                            that you want to be bound to a view each time that
                                                                                                                              'profile', 'App\Http\ViewComposers\ProfileComposer'
                                                                                        view is rendered, a view composer can help you
view composer
                            organize that logic into a single location.
                                                                                                                            // Using Closure based composers...
                            实例:https://scotch.io/tutorials/sharing-data-between-
                                                                                                                            View::composer('dashboard', function ($view) {
                           views-using-laravel-view-composers
                                                                                                                          * Register the service provider.
                                                                                                                          * @return void
                                                                                                                         public function register()
                                                                                                                        Remember, if you create a new service provider to contain your view composer registrations, you will need to add the
                                                                                                                       service provider to the providers array in the config/app.php configuration file.
                                                                                         创建app/Http/ViewComposers 文件夹
```

```
Now that we have registered the composer, the ProfileComposer@compose method will
be executed each time the profile view is being rendered. So, let's define the composer
class:
<?php
 namespace App\Http\ViewComposers;
use Illuminate\View\View;
use App\Repositories\UserRepository;
class ProfileComposer
   * The user repository implementation.
  * @var UserRepository
  protected $users;
   * Create a new profile composer.
   * @param UserRepository $users
  * @return void
  public function __construct(UserRepository $users)
    // Dependencies automatically resolved by service container...
    $this->users = $users;
   * Bind data to the view.
   * @param View $view
  * @return void
  public function compose(View $view)
     $view->with('count', $this->users->count());
 Just before the view is rendered, the composer's compose method is called with the
 Illuminate\View\View instance. You may use the with method to bind data to the view.
```

```
Attaching A Composer To Multiple Views
You may attach a view composer to multiple views at once by passing an array of views as the first argument to the composermethod:

View::composer(
    ['profile', 'dashboard'],
    'App\Http\ViewComposers\MyViewComposer'
);
The composer method also accepts the * character as a wildcard, allowing you to attach a composer to all views:

View::composer('*', function ($view) {
    //
});
```