

05. 回退. 当前路由. 单行为

学习要点：

1. 单行为控制器
2. 路由回退
3. 当前路由

本节课我们来开始学习路由和控制器的一些方法：回退、当前路由、单行为；

一. 单行为控制器

1. 之前的课程，我们简单的创建和定义了控制器，并继承了控制器基类；
2. 为何要继承基类？因为继承基类后，可以使用基类的方法，比如中间件等；
3. 继承基类后除了支持中间件快捷使用，还支持验证、列队等快捷方法；

```
public function __construct()
{
    $this->middleware('中间件');
}
```

4. 如果你想要定义一个只执行一个方法的控制器，可以使用单行为控制器；
5. 单行为控制器使用__invoke()方法，可以使用命令行创建；

```
php artisan make:controller OneController --invokable
```

```
//手工创建
class OneController extends Controller
{
    public function __invoke()
    {
        return '单行为控制器';
    }
}
```

6. 单行为控制器，路由定义就不需要指定特定的方法，指定控制器即可；
 7. 单行为控制器只是语义上的单行为，并没有限制创建更多方法访问；
- ```
Route::get('one', 'OneController');
```

### 二. 路由回退

1. 如果我们跳转到了一个不存在路由时，会产生 404 错误，体验不佳；
2. 可以使用回退路由，让不存在的路由自动跳转到你指定的页面去；
3. 注意：由于执行顺序问题，必须把回退路由放在所有路由的最底部；

```
Route::fallback(function () {
 return redirect('/');
});
```

4. 当然，你也可以制作一个自己的 404 页面，用回退路由加载这个页面；

```
Route::fallback(function () {
 return view('404');
});
```

### 三. 当前路由

1. 我们可以通过使用::current()系列方法，来获取当前路由的访问信息；

```
Route::get('index', function () {
 //当前路由信息
 dump(Route::current());
 //返回当前路由的名称
 return Route::currentRouteName();
 //返回当前路由指向的方法
 return Route::currentRouteAction();
})->name('localhost.index');
```