40.Blade 模板入门

学习要点：

1.Blade 简介

2.模板基础功能

本节课我们来开始学习 Blade 模板的功能以及模板的一些基础功能。

一. Blade 简介

1. Blade 是 Laravel 内置的模板引擎，之前课程中已经知道如何创建；
2. 我们用 Task 控制器的 user()方法来测试，使用 view()方法来引入模板；

```
public function user()
{
    return view('user');
}
```

3. 创建 user.blade.php 模板文件，模板支持原生 PHP 开发；

```
<?php echo 'Blade'.(1 + 1)?>
```

4. 和其它模板引擎一样，模板文件被执行后会缓存，而编辑修改后会自动重新缓存；
5. 我们将模板文件存放在 resources/views 目录里，后缀为：.blade.php；

二. 模板基础功能

1. 之前的课程中，简单使用过，我们再复习一下，在控制器可以给模板传递变量；

```
//参数 2 数组，声明模板变量
return view('user', [
    'name'  =>  'Mr.Lee'                        //{{$name}} 模板变量
]);

//facade 方法
return View::make('user', [
    'name'  =>  'Mr.Lee'
]);
```

2. 如果模板页面内容极其简单，也可以直接通过路由加载模板，绕过控制器；

```
Route::get('/task/user', function () {
    return view('user', [
        'name'  =>  'Mr.Lee'
    ]);
});
```

3. 如果模板根目录下建立子目录，调用方法用点符号作为路径格式；

```
//resources/views/admin/index.blade.php
return view('admin.index');
```

4. 有时可能有判断模板文件是否存在的需求，可以使用 exists()方法；

```
//判断模板文件是否存在
return view()->exists('admin.index');
//facade 方法
return View::exists('admin.index');
```

5. 也可以使用 first()方法，加载存在于数组中的第一个模板；

```
//加载存在于数组中的第一个模板
return view()->first(['abc', 'user', 'def'], [
    'name'  =>  'Mr.Lee'
]);
//facade 方法
return View::first(['abc', 'user', 'def']...
```