

## 06. 响应设置和重定向

学习要点：

1. 响应设置
2. 重定向

本节课我们来开始进入学习响应设置，和跳转重定向。

### 一. 响应设置

1. 路由和控制器处理完业务都会返回一个发送到浏览器的响应：`return`;
2. 比如字符串会直接输出，而数组则会输出 `json` 格式，本身是 `Response` 对象；

```
return [1, 2, 3]; //输出 json 格式
return response([1, 2, 3]); //同上
return response()->json([1, 2, 3]); //同上
```

3. 如果使用 `response()` 输出的话，可以设置状态码和响应头信息；  
`return response('index', 201);` //可以设置 HTTP 请求状态码
4. 也可以给 HTTP 添加或修改标头，比如将 `html` 解析模式改成文本 `plain` 模式；

```
return response('<b>index</b>')
->header('Content-Type', 'text/plain'); //文本解析模式
```

5. 结合上面的响应操作，再结合 `view()` 视图功能，显示纯 `HTML` 代码页面；

```
return response()->view('task', ['id'=>10], 201)
->header('Content-Type', 'text/plain');
```

### 二. 路由重定向

1. 重定向使用助手函数 `redirect()` 的 `to()` 方法，注意需要 `return` 才能跳转；

```
return redirect()->to('/'); //跳到首页
return redirect()->to('task'); //跳转到 task
return redirect()->to('task/url'); //跳转到 task/url
```

2. 也可以使用快捷方式直接进行跳转；

```
return redirect('/'); //跳到首页
return redirect('task'); //跳转到 task
return redirect('task/url'); //跳转到 task/url
```

3. `redirect()` 助手有一个对应的 `facade` 模式对象；

```
return Redirect::to('/'); //facade 模式，但需要 use 引入
```

4. 使用 `redirect()` 的 `route()` 方法，可以跳转到指定的命名路由 `URI`；

```
return redirect()->route('task.index'); //注意和 route() 方法区别
```

5. 使用 `redirect()` 的 `back()` 方法，可以重定向到上一个页面中；

```
return redirect()->back();  
return back(); //快捷方式
```

6. 使用 `redirect()` 的 `action()` 方法，可以直接重定向到控制器方法；

```
return redirect()->action('TaskController@index'); //需注册路由  
return redirect()->action('TaskController@index', ['id'=>10]);
```

7. 使用 `redirect()` 的 `away()` 方法，跳转到外部链接；

```
return redirect()->away('http://www.baidu.com'); //不带任何编码
```