

## 08. 资源嵌套. 浅嵌套. 自定义

学习要点：

1. 嵌套资源
2. 资源自定义

本节课我们来开始学习控制器的资源嵌套功能、浅嵌套以及资源自定义。

### 一. 嵌套资源

1. 声明：资源控制器是某个特定场景下的产物，完全理解需要 PHP 项目基础；
2. 比如开发过博客系统，留言帖子系统之类，具有类似思维，否则你懂的...；
3. 只是学习了 PHP 基础，就立刻学习框架的同学，可以过一遍即可(不影响后续)...；
4. 在一篇博文(Blog)下有多条评论(Comment)，编辑某条博文下的一条评论；
5. 以上需求，可以通过嵌套资源路由来实现这个功能；

```
php artisan make:controller CommentController --resource  
//嵌套资源路由  
Route::resource('blogs.comments', 'CommentController');
```

HTTP 类型	路由 URI	控制器方法	路由命名
GET	blogs/{blog}/comments	index()	blogs.comments.index
GET	blogs/{blog}/comments/create	create()	blogs.comments.create
POST	blogs/{blog}/comments	store()	blogs.comments.store
GET	blogs/{blog}/comments/{comment}	show()	blogs.comments.show
GET	blogs/{blog}/comments/{comment}/edit	edit()	blogs.comments.edit
PUT/PATCH	blogs/{blog}/comments/{comment}	update()	blogs.comments.update
DELETE	blogs/{blog}/comments/{comment}	destroy()	blogs.comments.destroy

6. 以上需求，可以通过嵌套资源路由来实现这个功能，编辑方法以及传参如下：

```
public function edit($blog_id, $comment_id)  
{  
    return '编辑博文下的评论，博文 id: '.$blog_id.', 评论 id: '.$comment_id;  
}
```

7. 而实际上，每个 id 都是独立唯一的，并不需要父 id 和子 id 同时存在；
8. 为了优化资源嵌套，通过路由方法->shallow()实现浅层嵌套方法；

```
//浅层嵌套  
Route::resource('blogs.comments', 'CommentController')->shallow();
```

9. 实现后的路由，在传递参数方法也比较精准，具体如下：

HTTP 类型	路由 URI	控制器方法	路由命名
GET	blogs/{blog}/comments	index()	blogs.comments.index
GET	blogs/{blog}/comments/create	create()	blogs.comments.create
POST	blogs/{blog}/comments	store()	blogs.comments.store
GET	comments/{comment}	show()	blogs.comments.show
GET	comments/{comment}/edit	edit()	blogs.comments.edit
PUT/PATCH	comments/{comment}	update()	blogs.comments.update
DELETE	comments/{comment}	destroy()	blogs.comments.destroy

```
public function edit($id)
{
    return '评论 id.'.$id;
}
```

10. 如果觉得资源路由命名过长，可以自己自定义，有两种方式：

```
->name('index', 'b.c.i');
->names([
    'index' => 'b.c.i'
]);
```

11. 如果觉得资源路由的参数不符合你的心意，也可以改变：

```
->parameter('blogs', 'id');
->parameters([
    'blogs'    => 'blog_id',
    'comments' => 'comment_id'
]);
```