

## 39. 路由的分组和注解

学习要点：

1. 路由分组
2. 注解路由

本节课我们来学习一下路由的分组功能和路由的注解。

### 一. 路由分组

1. 路由分组，即将相同前缀的路由合并分组，这样可以简化路由定义，提高匹配效率；
2. 在定义分组路由前，我们专门做一个类，来实际演练这个效果；

```
class Collect extends Controller
{
    public function index()
    {
        return 'index';
    }

    public function read($id)
    {
        return 'read id:'.$id;
    }

    public function who($name)
    {
        return 'your name:'.$name;
    }
}
```

3. 使用 `group()` 方法，来进行分组路由的注册；

```
Route::group('col', [
    ':id'      => 'Collect/read',
    ':name'    => 'Collect/who'
])->ext('html')->pattern(['id'=>'d+$', 'name'=>'w+$']);
```

4. 使用 `group()` 方法，并采用闭包的形式进行注册；

```
Route::group('col', function () {
    Route::get(':id', 'Collect/read');
    Route::get(':name', 'Collect/who');
})->ext('html')->pattern(['id'=>'d+$', 'name'=>'w+$']);
```

5. 使用 `prefix()` 方法，简化路径的地址；

```
Route::group('col', function () {
    Route::get(':id', 'read');
    Route::get(':name', 'who');
}) ->prefix('Collect/')
    ->ext('html')
    ->pattern(['id'=>'d+$', 'name'=>'w+$']);
```

6. 使用 `append()` 方法，可以额外传入参数，用 `request` 获取；

```
Route::group()...->append(['flag'=>1]);
```

7. 路由规则(主要是分组和域名路由)定义的文件，加载时会解析消耗较多的资源；  
 8. 尤其是规则特别庞大的时候，延迟解析开启让你只有在匹配的时候才会注册解析；  
 9. 我们在 `app.php` 中开启延迟解析，多复制几组规则，然后通过 `trace` 来查看内存；

```
'url_lazy_route' => true,
```

## 二. 注解路由

1. 路由系统还提供了一个可以在注解(注释)中直接创建路由的方式，但默认关闭；  
 2. 我们在 `app.php` 中，开启路由注解功能；

```
'route_annotation' => true,
```

3. 然后在控制器设置注解代码即可，可以使用 `PHPODOC` 生成一段，然后添加路由规则；

```
/**
 * @param $id
 * @return string
 * @route('col/:id');
 */
```

4. 第二参数，可以设置请求类型，而需要设置更多的规则，可以换行设置；

```
/**
 * @param $id
 * @return string
 * @route('col/:id', 'get')
 * ->ext('html')
 * ->pattern(['id'=>'d+'])
 *
 */
```

5. 有几个注意点：语句结尾不需要分号，路由规则结束后，需要有一个空行；  
 6. 支持资源路由，下节课会讲到具体资源路由；

```
/**
 * @route('col')
 */
class Collect extends Controller
```