

## 80. 中间件.下

学习要点：

1. 路由中间件
2. 控制器中间件

本节课我们来学习一下要使用一下路由中间件，以及控制器中间件。

### 一. 路由中间件

1. 创建一个给路由使用的中间件，判断路由的 ID 值实现相应的验证；

```
class Auth
{
    public function handle($request, \Closure $next)
    {
        if ($request->param('id') == 10) {
            echo '是管理员！';
        }
        return $next($request);
    }
}
```

1. 路由方法提供了一个 `middleware()` 方法，让指定的路由采用指定的中间件；

```
Route::rule('ar/:id', 'Address/read')
    ->middleware(\app\middleware\Auth::class);

    ->middleware([Auth::class, Check::class]); //支持多个中间件，这里 use 了
```

2. 也可以在 `config/middleware.php` 配置文件加中，配置别名支持；

```
'alias'    => [
    'Auth'   => app\middleware\Auth::class,
    'Check'  => app\middleware\Check::class,
],

->middleware(['Auth', 'Check']);
```

3. 可以给中间件传递额外参数，通过中间件入口方法的第三个参数接收；

```
->middleware(Auth::class, 'ok');

public function handle($request, \Closure $next, $param, $param = '')
```

4. 中间件也支持分组路由，闭包路由等；

```
Route::group('ar', function () {
    Route::rule('/:id', 'Address/read')
})->middleware(Auth::class);
```

```
Route::rule('ar/:id', 'Address/read')
    ->middleware(function ($request, \Closure $next) {
        if ($request->param('id') == 10) {
            echo '是管理员!';
        }
        return $next($request);
    });
```

### 三. 控制器中间件

1. 可以让中间件在控制器里注册，让这个控制器执行的时候执行中间件；

```
protected $middleware = ['Check'];
```

2. 默认情况下，控制器中间件对所有操作方法有效，支持做限制；

```
protected $middleware = [
    'Auth' => ['only' => ['index']],
    'Check' => ['except' => ['read']],
];
```

3. 中间件给控制器传递参数，通过 Request 对象实现；

```
$request->name = 'Mr.Lee';
```