

54. 中间件【下】

学习要点：

1. 前/后置中间件
2. 路由中间件
3. 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. 前/后置中间件

1. 上节课，我们创建了一个简单的中间件，它拦截 HTTP 验证请求匹配后跳转；
2. 这种将`$next($request)`放在方法底部的方式，属于前置中间件；
3. 前置中间件就是请求阶段来进行拦截验证，比如登录判断、跳转、权限等；
4. 而后置中间件就是请求完毕之后再进行验证，比如写入日志等等；

```
public function handle(Request $request, \Closure $next)
{
    //中间件代码，前置
    return $next($request);
}
```

```
public function handle(Request $request, \Closure $next)
{
    $response = $next($request);
    //中间件代码，后置
    return $response;
}
```

二. 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 $request, \Closure $next)
    {
        if ($request->param('id') == 10)
        {
            echo '是管理员，提供后台权限并跳转操作';
        }
        return $next($request);
    }
}
```

5. 如果将 Auth 中间件注册到 `middleware.php` 中，就变成公有中间件了；
6. 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('read/:id', 'Inject/read')->middleware('Auth');
```

7. middleware()方法，除了传类名，还可以是命名空间的两种形式，均支持；

```
->middleware('app\http\middleware\Auth')
->middleware(app\http\middleware\Auth::class)
```

8. 一个路由规则，如果要注册多个中间件，可以用数组的绑定；

```
Route::rule('read/:id', 'Inject/read')
    ->middleware(['Auth', 'Check']);
```

9. 也支持分组路由，闭包路由等；

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

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

10. 中间件 handler()方法的第三参数，可以路由进行设置；

```
Route::rule('read/:id', 'Inject/read')
    ->middleware('Auth:abc');
```

```
public function handle(Request $request, \Closure $next, $name)
{
    echo $name;
}
```

11. 在定义全局中间件绑定的时候，如果想传入参数，可以设置为数组模式；

```
[app\http\middleware\Auth::class, 'hello']
'Auth',
'Auth:hello'
```

三. 控制器中间件

1. 可以让中间件在控制器里注册，控制器必须继承 Controller 基类；

```
class Inject extends Controller
{
    protected $middleware = ['Auth'];
}
```

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', 'test']],  
    'Check' => ['except' => ['bhv', 'read']],  
];
```

3. 中间件给控制器传递参数，通过 Request 对象实现；

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