

## 53. 中间件【上】

学习要点：

### 1. 定义中间件

本节课我们来学习一下中间件的用法，定义一下中间件。

#### 一. 定义中间件

1. 中间件和钩子有点类似，它主要用于拦截和过滤 HTTP 请求，并进行相应处理；
2. 这些请求的功能可以是 URL 重定向、权限验证等等；
3. 为了进一步了解中间件的用法，我们首先定义一个基础的中间件；
4. 可以通过命令行模式，在应用目录下生成一个中间件文件和文件夹；

php think make:middleware Check

5. 具体路径为：application\http\middleware\Check.php ;  

```
namespace app\http\middleware;  
use think\Request;  
class Check  
{  
    public function handle(Request $request, \Closure $next)  
    {  
        if ($request->param('name') == 'index') {  
            return redirect('/');  
        }  
        return $next($request);  
    }  
}
```

6. 然后将这个中间件进行注册，在应用目录下创建 middleware.php 中间件配置；

```
return [  
    app\http\middleware\Check::class  
];
```

7. 中间件的入口执行方法必须是：handle()方法，第一参数请求，第二参数是闭包；
8. 业务代码判断请求的 name 如果等于 index，就拦截住，不再执行，跳转到首页；
9. 但如果请求的 name 是 lee，那需要继续往下执行才行，不能被拦死；
10. 那么就需要\$next(\$request)把这个请求去调用回调函数；
11. 中间件 handle()方法规定需要返回 response 对象，才能正常使用；
12. 而\$next(\$request)，研读源码追踪发现，它就是返回的 response 对象；
13. 为了测试拦截后，无法继续执行，可以 return response()助手函数测试；