

52. 钩子和行为

学习要点：

1. 概念理解
2. 小实例

本节课我们来学习一下系统架构中钩子和行为的概念和流程的理解。

一. 概念理解

1. 首先，钩子和行为在 6.0 的版本被废弃了，用事件来取代；
2. 虽说用事件来取代，不过意思是一样的，我们还是有必要理解一下；
3. 什么是行为，就是在系统执行的流程中执行的一个动作；
4. 比如，当执行到路由时，对路由的设置进行一系列的检测，这种就叫行为；
5. 而钩子又是什么呢？可以理解为行为执行的那个位置点，触发点；
6. 系统架构里用了很多这种方式实现框架程序，我们自己手工来创建一个试试；

二. 小实例

1. 在应用目录下创建一个 behavior 文件夹，用于存放行为类，比如 Test.php

```
namespace app\behavior;
class Test
{
    public function run($params)
    {
        echo $params.', 只要触发, 我就执行!';
    }
}
```

2. 行为类创建好之后，设置一个入口方法 run(), run() 方法只要钩子被触发就执行；
3. 比如，我们将行为注册到 tags.php 中应用初始化的数组里(app_init)；

```
// 应用初始化
'app_init' => [
    'app\behavior\Test',
],
```

4. 我们也可以自定义一个钩子，然后注册到 tags.php 中，执行后触发；

```
public function bhv()
{
    //钩子
    Hook::listen('eat', '吃饭');
}
```

```
//自定义
'eat' => [
    'app\behavior\Test',
],

public function appInit($params)
{
    echo '初始化的行为被触发！';
}

public function eat($params)
{
    echo $params.'的行为被触发！';
}
```

- 5. 那么,我们可不可以让初始化对应的是初始化的行为,自定义对应自定义的行为呢;
- 6. app_init 对应的方法是 appInit(有下划线的大写), 而自定义 eat 就是 eat;

7. 系统除了 app_init 钩子, 还提供了一系列的钩子供使用;

钩子	描述	参数
app_init	应用初始化标签位	无
app_dispatch	应用调度标签位	无
app_begin	应用开始标签位	无
module_init	模块初始化标签位	无
action_begin	控制器开始标签位	当前的callback参数
view_filter	视图输出过滤标签位	当前模板渲染输出内容
app_end	应用结束标签位	当前响应对象实例
log_write	日志write方法标签位	当前写入的日志信息
log_level	日志写入标签位	包含日志类型和日志信息的数组 (V5.1.25+)
response_send	响应发送标签位	当前响应对象
response_end	输出结束标签位	当前响应对象实例