

## 【一】插件类主文件

- 插件命名规范

插件类名采用大驼峰命名法，都带 Plugin类名后缀，如 DemoPlugin,ClientCarePlugin

- 插件类主文件位于插件根目录

命名格式：插件名+Plugin.class.php

- 在插件根目录创建AliFacePayPlugin.class.php文件, 主要代码如下：

```
<?php
namespace Plugin\AliFacePayPlugin;
use Plugin\Plugin;
class AliFacePayPlugin extends Plugin
{
    /**
     * 插件挂载钩子
     * @var array
     */
    public $hooks = ['pluginPaymentHtml', 'pluginPaymentJs'];
    /**
     * 支付的html页面
     * @param array $param
     */
    public function pluginPaymentHtml($param = [])
    {
        if ($this->verify()) {
            $this->assign('payment', $param);
            $this->display('View/payment');
        }
    }
    /**
     * 上面支付html页面加载的js
     */
    public function pluginPaymentJs()
    {
        if ($this->verify()) {
            $path = substr(PLUGIN_PATH . $this->code . "/View/js/payment.js",
1);
            echo "<script src='{$path}'></script>" . PHP_EOL;
        }
    }
}
```

支付的html页面和js会加载到template/User/Zkeys/PC/Page/Finance/Payment/index.html文件中

插件支付

```
{:hook('pluginPaymentHtml', $payment )}
```

插件支付

插件支付JS start

```
{:hook('pluginPaymentJs', $payment )}
```

插件支付JS end

插件调用支付接口

[http://域名/User/Payment/pay/plugin/AlifacePay?total\\_fee=金额](http://域名/User/Payment/pay/plugin/AlifacePay?total_fee=金额)

请求的是app/Plugin/AlifacePay/Controller/PaymentController.class.php::pay()

在pay方法调用createOrder(\$userID, \$payType, \$out\_trade\_no, \$total\_fee, \$extraParams = [])方法创建充值记录，参数分别是用户ID、充值类型、订单号、充值金额、订单其他信息。

pay()方法返回数据，应该包含参数out\_trade\_no如

```
{"code": "10000", "msg": "Success", "out_trade_no": "740682633736892091", "qr_code": "https://qr.alipay.com/bd0xxxcisxxx"}
```

然后通过轮询/user/payment/ajaxCheck?orderNo=740682633736892091查询是否支付完成

需要创建一个类接收第三方平台（如支付宝）的回调，  
如app/Plugin/AlifacePay/Controller/NotifyController.class.php,则回调地址为

[http\(s\)://域名/ApiNotify/notify/alipayReturn/plugin/AlifacePay](http(s)://域名/ApiNotify/notify/alipayReturn/plugin/AlifacePay)会调用

\Plugin\AlifacePay\Controller\NotifyController::alipayReturn()

伪代码：

```
M()->startTrans();
    if (!checkOrderStatus($out_trade_no)) {
        $orderHandleResult = orderHandle($parameter); //进行订单处理，并传送从
        支付宝返回的参数;
        if ($orderHandleResult) {
            M()->commit();
            // 支付订单
            $orderModel = new OrdersModel();
            $orderModel->callbackOrderPay($out_trade_no);
        } else {
            M()->rollback();
        }
    }
}
```