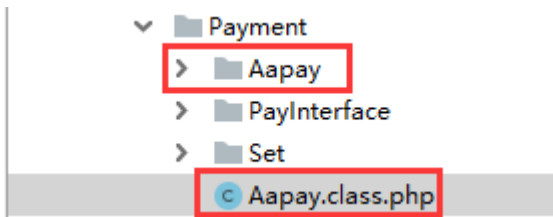


## 支付部分

### 第一步

在libs/Niaoyun/Payment 目录下新建要接入的支付方式php文件, 比如要接入Aapay支付则新建Aapay.class.php,若存在需要引入的SDK文件则也在该目录下新建文件夹存放, 如下Aapay支付



### 第二步

创建的支付方式类需要继承libs/Niaoyun/Payment/PayInterface/Ipay.class.php接口类并实现里面的pay() [] getConfigAdmin() 以及 getConfig() 接口 (Ipay.class.php接口默认存在)

Ipay.class.php:

```
<?php
namespace Niaoyun\Payment\PayInterface;

interface Ipay
{
    public function pay($parameters);

    /**
     * @return array
     */
    public function getConfigAdmin();

    /**
     * 这个方法给其他地方调用, 比如回调时候需要用到支付参数检验签名等。
     * @return array
     */
    public function getConfig();
}
```

Aapay.class.php

```

class Aapay implements Ipay
{

    /**
     * 支付方式 可选: Wechat, Alipay, Unionpay
     * @var string
     */
    private $payment_method;

    /**
     * 支付参数配置
     * @var string
     */
    private $config
    |   = [...];

    private $configAdmin
    |   = [...];

    /**
     * @return array
     */
    public function getConfigAdmin() {...}

    /**
     * @return string
     */
    public function getConfig() {...}

    public function pay($parameters) {...}
}

```

其中 `getConfigAdmin` 用于获取接口配置项，并展示与【平台后台】-【系统】-【全局设置】-【充值】中，便于管理员修改接口信息

支付参数配置以实际接口为准，示例如下图：

```

/**
 * 支付参数配置
 * @var string
 */
private $config
= [
    'mid' => '',
    'appid' => '',
    'api_key' => '',
    'url' => 'https://portal.aapay.hk/index.php/index/pay/process',
];

private $configAdmin
= [
    'mid' => [ //与$config中的键名对应
        'name' => 'mid参数', //后台参数展示的名称
        'show' => true, //该值决定参数是否在后台页面展示, true为展示, false为隐藏
    ],
    'appid' => [ //与$config中的键名对应
        'name' => 'appid参数', //后台参数展示的名称
        'show' => false, //该值决定参数是否在后台页面展示, true为展示, false为隐藏
    ],
    'api_key' => [ //与$config中的键名对应
        'name' => 'api_key参数', //后台参数展示的名称
        'show' => false, //该值决定参数是否在后台页面展示, true为展示, false为隐藏
    ],
];

```

获取平台配置数据

```

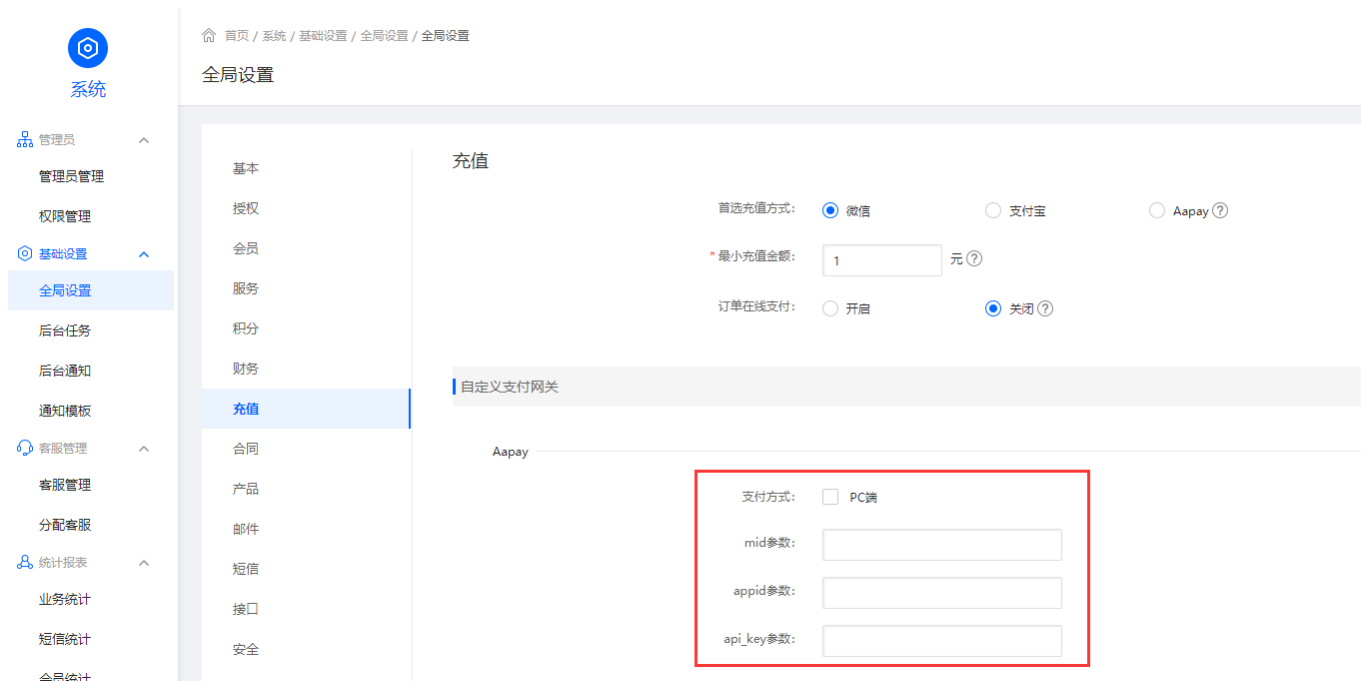
public function __construct()
{
    foreach ($this->configAdmin as $k => $v) {
        $this->config[$k] = C('recharge')[substr( string: __CLASS__, start: strrpos( haystack: __CLASS__, needle: '\\') + 1) . $k];
    }
}

/**
 * @return array
 */
public function getConfigAdmin()
{
    return $this->configAdmin;
}

```

平台参数配置位置：【后台】-【系统】-【全局设置】-【充值】

注：目前仅支持PC端



pay方法接收一个数组参数\$parameters格式如下：

字段	类型	描述
total_fee	Float	付款金额
out_trad_no	String	订单系统唯一订单号
body	String	订单名称

### 第三步

Aapay.class.php 实现的pay方法实现你自己的支付方式返回结果格式分为两种如下：

**第一种：** 支付方式是类似微信扫码支付，并且返回的地址需要使用js去生成二维码，则需要返回如下结果：

```
[
'code'=>200, //200表示成功, 201 表示失败
'code_url'=>$code_rul, //支付方式返回的是二维码数据，并需要使用js生成二维码的则用code_url字段，用于 iframe 页面（生成二维码的js平台已集成，不用另外开发）
'msg'=>'success' //成功success[]失败fail
]
```

**第二种：** 支付方式是类似支付宝的页面跳转，则需要返回如下结果：

```
[
'code'=>200, //200表示成功, 201 表示失败
'html_text'=>$html_text, //返回的是跳转链接则用html_text字段，用于新窗口打开的页面
'msg'=>'success' //成功success[]失败fail
]
```

**第三种：** 支付方式是类似微信扫码支付，并且返回的地址是已生成好的二维码地址，则需要返回如下结果：

```
[
'code'=>200, //200表示成功, 201 表示失败
```

```

'qrcode_url'=>$qrcode_url, //返回的是已经生成好的二维码地址， 则用qrcode_url字段，
用于 iframe 页面
'msg'=>'success' //成功success[]失败fail
]

```

Pass[]两者返回的字段名不一样

### 返回给前端的内容

用于二维码支付，根据type类型判断是否需要js生成二维码

```

{
    "result":true,
    "total_fee":"10.00", //价格
    "type":"2", //0:代表qrcodeUrl字段需要js生成二维码; 1: 新窗口打
开[]2[]qrcodeUrl字段为已经生成好的二维码
    "qrcodeUrl":"http://qrcodeUrl.test.com/xxxx", //二维码地址
    "orderNo":"123456789" //订单编号
}

```

### 回调部分

#### 第一步

在 app\ApiNotify\Controller\ 下创建回调php文件，比如[]Aapay[]则创建AapayNotifyController.class.php，继承并实现app\ApiNotify\controller\PaymentNotifyController.class.php 中的 checkSign 和 getPayStatus 这两个方法

```

class AapayNotifyController extends PaymentNotifyController
{
    private $payConfig = [];

    function _initialize()
    {
        parent::_initialize();
        $className = substr( string: __CLASS__, start: strpos( haystack: __CLASS__, needle: '\\' ) + 1, length: -16);
        $class      = new \ReflectionClass( argument: "\Niaoyun\Payment\\".$className );
        /** @var \Niaoyun\Payment\PayInterface\Ipay $obj */
        $obj        = $class->newInstance();
        //获取Aapay支付配置参数并赋值给payConfig
        $this->payConfig = $obj->getConfig();
    }

    /**
     * 验证签名
     */
    public function checkSign($parameters) {...}

    /**
     * 回调状态
     */
    public function getPayStatus($parameters) {...}
}

```

```

/*
 * 验证签名
 */
public function checkSign($parameters)
{
    //验证签名，具体逻辑请按实际情况编写
    if ($parameters == md5(time()))
    {
        return true;
    } else {
        return false;
    }
}

```

```

/*
 * 回调状态
 */
public function getPayStatus($parameters)
{
    if ($parameters['result_code'] == 'SUCCESS') {
        //以下参数用于平台调用
        $this->orderNo    = $parameters['mh_oid']; //订单号
        $this->trade_no   = ''; //支付回调交易号
        $this->total_fee  = $parameters['origin_amount']; //订单金额
        $this->payAccount = $parameters['client_id']; //买家唯一标识
        Log::write( message: '回调状态: true', level: 'payment');
        return true;
    } else {
        return false;
    }
}

```

## 第二步

支付回调成功地址：<http://您的平台域名/ApiNotify/AapayNotify/notify>

备注：支付方式的图标放在 `public/template/user/Zkeys/PC/static/css/ucModule/finance/img/` 下面，图标名字为支付方式名和创建的对应php文件类名一样

附思维导图流程截图：

# 统一支付接入

## 支付部分

step1

libs/Niaoyun/Payment目录下新建要接入的支付方式php文件

例如: Aapay支付则新建Aapay.class.php

Aapay.class.php文件

若存在需要引入的SDK文件则也在该目录下新建文件来存放

例如 Aapay支付则新建 Aapay文件夹

```
use Niaoyun\Payment\PayInterface\pay;
```

实现pay方法

pay方法接收的参数\$parameters

- total\_fee(付款金额)
- out\_trade\_no(订单系统唯一订单号)
- body(订单名称)

pay方法进行支付实现后返回结果

```
若支付接口是二维码支付  
[  
    'code'=>200, //200表示成功,201 表示失败  
    'code_url'=>$code_url, //支付方式返回的是二维码数据则用code_url字段  
    'msg'=>'success' //成功success, 失败fail  
]  
若支付接口是页面跳转  
[  
    'code'=>200, //200表示成功,201 表示失败  
    'html_text'=>$html_text, //返回的是跳转链接则用html_text字段  
    'msg'=>'success' //成功success, 失败fail  
]
```

## 回调部分

在App\ApiNotify\Controller下创建回调php文件

比如Aapay则创建AapayNotifyController.class.php

支付回调成功地址http://xxx.com/ApiNotify/AapayNotify/notify

继承并app\ApiNotify\Controller\PaymentNotifyController.class.php

实现checkSign和getPayStatus这两个方法

## 支付方式图标

支付方式的图标放在public/template/User/Zkeys/PC/static/css/ucModule/finance/img/下面

图标名字为支付方式名和创建的对应该php文件类名一样