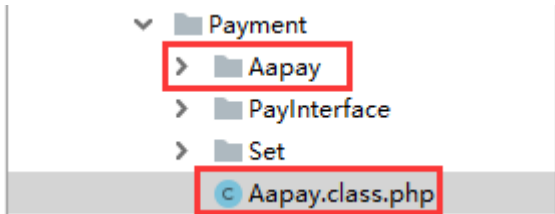


## PC端

### 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()` 和 `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
    |   = [...];

    public function __construct(){...}

    public function getPayName(){...}

    /**
     * @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

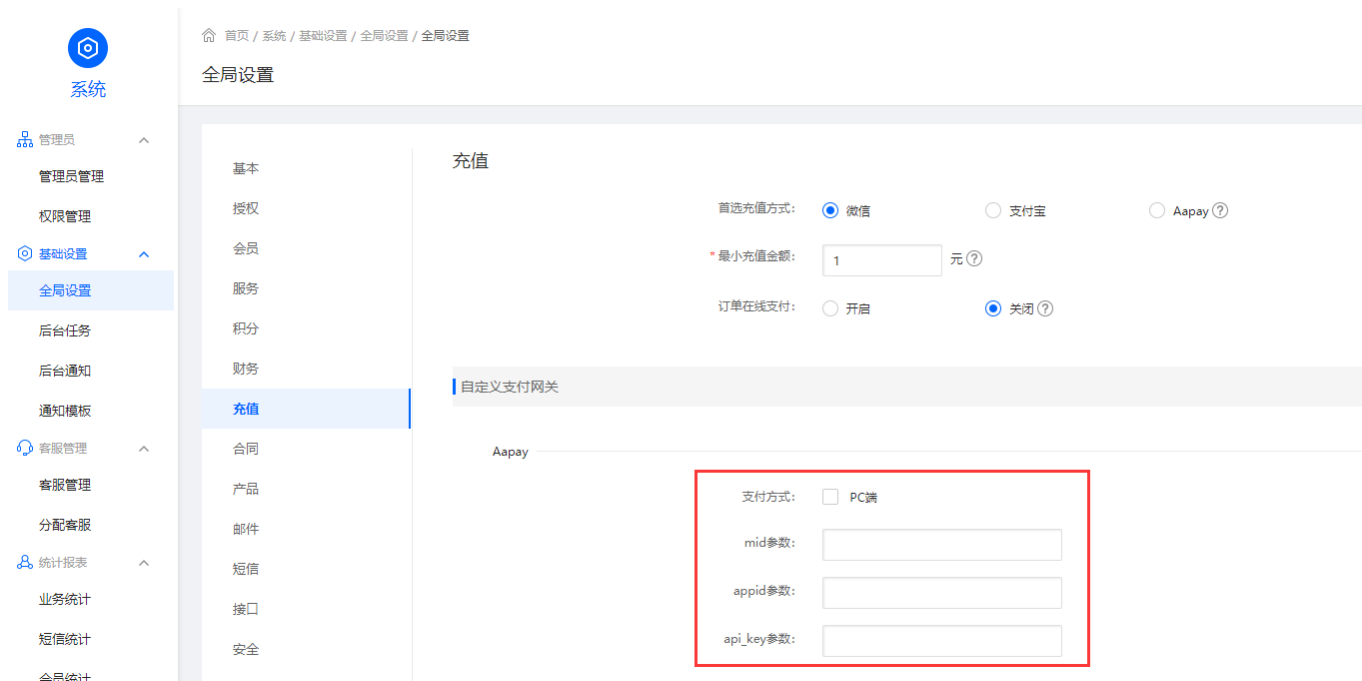
```

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】

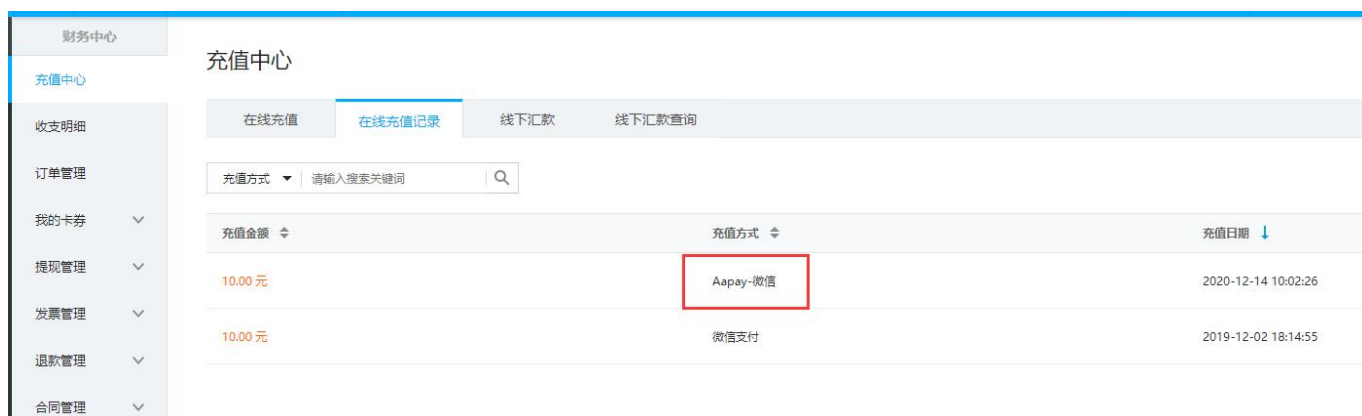


getPayName 方法为自定义支付名称，不实现该方法将用默认值。例：

```

52
53 public function getPayName()
54 {
55     return 'AaPay-微信';
56 }
57

```



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_trade_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: strrpos( 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:

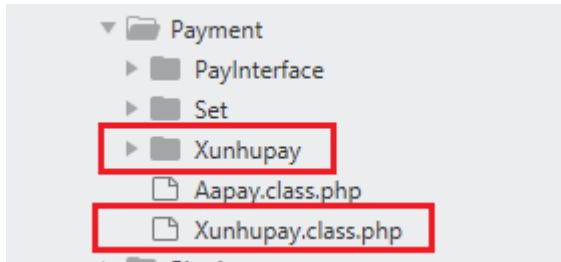


手机版

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Xunhupay支付则新建Xunhupay.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Xunhupay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 wapPay() 和 getConfigAdmin() 以及 getConfig() 接口 (Ipay.class.php接口默认存在)

getPayName() 接口可以实现, 实现则支付时候显示为自定义名称, 否则显示为类名

Ipay.class.php:

```

Ipay.class.php
<?php
namespace Niaoyun\Payment\PayInterface;

interface Ipay
{
    /** 支付 */
    public function pay($parameters); //PC调用的自定义方式

    /** */
    public function getConfigAdmin(); //获取后台设置的参数

    /** */
    public function getConfig(); //提供给其他地方调用设置参数, 如支付对调签名认证

    /** */
    public function getPayName(); //获取自定义支付名称

    /** */
    public function wapPay($parameters); //移动端的自定义支付方式
}
```

Xunhupay.class.php



```

<?php
class Xunhupay implements Ipay
{
    private $config =

    private $configAdmin =

    public function __construct()
    {
    }

    public function getConfig()
    {
        return $this->config;
    }

    public function getConfigAdmin()
    {
        return $this->configAdmin;
    }

    public function getPayName()
    {
        return '支付宝';
    }

    //实现PC支付
    public function pay($parameters)
    {
        //这里实现您的支付方式
    }

    //实现移动端支付
    public function wapPay($parameters)
    {
        //这里实现您的支付方式
    }
}

```

未实现getPayName()方法，则默认为类名

返回

充值中心

充值记录

充值方式



Xunhupay

充值最小金额 1 元，充值金额必须为整数

充值金额

请输入您的充值金额

开始充值

未实现getPayName()方法

实现getPayName()方法，则为自定义名称

返回充值中心充值记录

充值方式

支付宝

充值最小金额 1 元，充值金额必须为整数

充值金额

1

开始充值

其中支付参数配置如下，`configAdmin` 中的键必须在 `config` 中存在，如不需要在后台配置  
的参数，`configAdmin` 中可以省略，如`apiurl`在`config`中存在，在`configAdmin`中不存在，  
`getConfigAdmin` 用于获取接口配置项，并在【平台后台】-【系统】-【全局设置】-【充值】中展示，  
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：

ipay.class.php

Xunhupay.class.php

```

<?php
//
namespace NiaoYun\Payment;

use NiaoYun\Payment\PayInterface\Ipay;
use Think\Log;

class Xunhupay implements Ipay
{
    /**
     * 支付参数配置
     * @var array
     */
    private $config
    = [
        'mchid' => '', //平台分配商户号
        'key'   => '', //商户KEY
        'apiurl' => 'https://admin.xunhuweb.com/pay/payment', //支付API地址
    ];

    private $configAdmin
    = [
        'mchid' => [ //与config中键名一样
            'name' => '商户ID', //后台参数展示的名称
            'show' => true, //参数的值在后台是否展示, trur展示, false隐藏
        ],
        'key' => [ //与config中键名一样
            'name' => '商户KEY', //后台参数展示的名称
            'show' => false, //参数的值在后台是否展示, trur展示, false隐藏
        ],
    ];

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

```

Xunhupay

支付方式: ☒ PC端 ☒ 手机端

商户ID:

商户KEY:

因为show为false,这里隐藏了

平台参数配置位置：【后台】-【系统】-【全局设置】-【充值】

首页 / 系统 / 基础设置 / 全局设置 / 全局设置

全局设置

基本

授权

会员

服务

积分

财务

充值

合同

产品

邮件

短信

接口

安全

充值

首选充值方式:

☐ 微信
☒ 支付宝
☐ Xunhupay ?

\*最小充值金额:

元 ?

订单在线支付:

☒ 开启
☐ 关闭 ?

自定义支付网关

Xunhupay

支付方式: ☒ PC端 ☒ 手机端

商户ID:

商户KEY:

wapPay 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 |

| 字段          | 类型     | 描述        |
|-------------|--------|-----------|
| out_trad_no | String | 订单系统唯一订单号 |
| body        | String | 订单名称      |

### 第三步

wapPay方法实现你自己的支付方式，返回结果格式分为两种如下（大多数为第一种）

第一种(也是主流的方式)：

支付方式是类似微信H5支付，进行URL重定向,则需要返回如下结果：

```
[
  'code'=>200,    //200表示成功, 400 表示失败
  'type'=>'redirect',
  'url'=>$url,    //返回的是即将跳转的URL地址
  'msg'=>'success' //成功success[]失败fail
]
```

伪代码示例如下：

```
public function wapPay($parameters)
{
    //这里按照文档生成url
    $pay_url = $this->generalPayUrl($parameters);
    //生成url, 按照这种数组方式返回去
    return ['code' => 200, 'type' => 'redirect', 'url' => $pay_url, 'msg' => 'success'];
}
```

第二种：

支付方式返回的是一个html页面：

```
[
  'code'=>200,    //200表示成功, 400 表示失败
  'type'=> 'html', //固定类型, 表示返回的是html页面
  'html'=>$html, //支付方式返回的是一个新的html自动跳转页面
  'msg'=>'success' //成功success[]失败fail
]
```

伪代码示例如下：

```

public function wapPay($parameters)
{
    //根据文档，按照特定的程序生成的html代码如下(html有删减，仅做示例)
    $sHtml = <<<eof
<html>
<head></head>
<body>
<form id="alipaysubmit" name="alipaysubmit" action="https://mapi.alipay.com/gateway.do?_input_charset=utf-8" method="get">
<input type="hidden" name="_input_charset" value="utf-8" />
<input type="hidden" name="app_pay" value="Y" />
<input type="hidden" name="total_fee" value="1" />
<input type="hidden" name="sign" value="ZNL9TxN2+g5HUPPfWkc/ilifpXA3seabTAmqGeZTdntCf09H08hovx7wiQ==" />
<input type="hidden" name="sign_type" value="RSA" />
<input type="submit" value="确认" style="display:none;" />
</form>
<script>document.forms['alipaysubmit'].submit();</script>
</body>
</html>
eof;
    return ['code' => 200, 'type' => 'html', 'html' => $sHtml, 'msg' => 'success'];
}

```

回调部分

## 第一步

在 app\ApiNotify\Controller\ 下创建回调php文件，比如Xunhupay则创建XunhupayNotifyController.class.php，继承并实现 app\ApiNotify\controller\PaymentNotifyController.class.php 中的 checkSign 和 getPayStatus 这两个方法。这2个方法参数获取的是 I('request.')，如果获取不到，请用其他方式，如\$raw = file\_get\_contents('php://input'); (不同平台回调方式不一样)

```

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/{自定义支付类名} + Notify/notify>

如: <http://www.example.com/ApiNotify/XunhupayNotify/notify>

备注:

支付方式的图标放在 `templateUserZkeysMobileStaticcssucMobilefinanceimg` 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, 如Xunhupay.png 建议大小是40px \* 40px