






虚拟货币支付 PHP SDK

基于虚拟货币支付平台V2接口开发的PHP SDK，支持多种虚拟货币支付方式。

目录

- [功能特性](#)
- [环境要求](#)
- [安装配置](#)
- [快速开始](#)
- [文件说明](#)
- [接口说明](#)
- [签名算法](#)
- [常见问题](#)
- [注意事项](#)

功能特性

-  支持SHA256WithRSA签名算法
-  支持统一下单接口
-  支持订单查询接口
-  支持异步通知处理
-  支持同步跳转处理
-  完善的签名验证机制
-  详细的错误处理
-  支持多种虚拟货币支付方式

环境要求

- PHP >= 7.0
- OpenSSL 扩展
- cURL 扩展
- JSON 扩展

安装配置

1. 获取密钥

1. 登录商户后台
2. 进入"个人资料 -> API信息"页面
3. 点击"生成商户RSA密钥对"
4. 保存好以下内容：
 - 商户私钥：用于签名请求
 - 平台公钥：用于验证回调签名

2. 配置参数

编辑 `config.php` 文件，填写以下信息：

```
php

return [
    // 商户ID
    'pid' => '您的商户ID',

    // 商户私钥（完整复制，包括BEGIN和END标记）
    'private_key' => <<<EOT
-----BEGIN PRIVATE KEY-----
您的商户私钥内容
-----END PRIVATE KEY-----
EOT,

    // 平台公钥（完整复制，包括BEGIN和END标记）
    'public_key' => <<<EOT
-----BEGIN PUBLIC KEY-----
平台公钥内容
-----END PUBLIC KEY-----
EOT,

    // 回调地址（必须外网可访问）
    'notify_url' => 'https://yourdomain.com/notify.php',
    'return_url' => 'https://yourdomain.com/return.php',
];
```

3. 部署文件

将整个目录上传到您的Web服务器，确保以下文件可访问：

虚拟币支付DEMO/

— config.php	# 配置文件
— RsaHelper.php	# RSA工具类
— create_order.php	# 创建订单
— query_order.php	# 查询订单
— notify.php	# 异步通知
— return.php	# 同步跳转
— README.md	# 说明文档

快速开始

创建订单

```
<?php
require_once 'config.php';
require_once 'RsaHelper.php';

$config = require 'config.php';

// 订单参数
$orderData = [
    'method' => 'web',           // 接口类型
    'device' => 'pc',           // 设备类型
    'type' => 'USDT_TRC20',      // 支付方式
    'out_trade_no' => 'ORDER' . time(), // 订单号
    'name' => '商品名称',        // 商品名
    'money' => '100.00',         // 金额
];

// 构建请求参数
$params = [
    'pid' => $config['pid'],
    'method' => $orderData['method'],
    'type' => $orderData['type'],
    'out_trade_no' => $orderData['out_trade_no'],
    'notify_url' => $config['notify_url'],
    'return_url' => $config['return_url'],
    'name' => $orderData['name'],
    'money' => $orderData['money'],
    'clientip' => '127.0.0.1',
```

```
php    'timestamp' => RsaHelper::getTimestamp(),
    ];

    // 生成签名
    $params['sign'] = RsaHelper::generateSign($params, $config['private_key']);
    $params['sign_type'] = 'RSA';

    // 发送请求...
```

查询订单

```
php

<?php
require_once 'config.php';
require_once 'RsaHelper.php';

$config = require 'config.php';

// 查询参数
$params = [
    'pid' => $config['pid'],
    'out_trade_no' => 'ORDER123456',      // 商户订单号
    'timestamp' => RsaHelper::getTimestamp(),
];

// 生成签名
$params['sign'] = RsaHelper::generateSign($params, $config['private_key']);
$params['sign_type'] = 'RSA';

// 发送请求...
```

文件说明

文件	说明
config.php	配置文件，包含商户ID、私钥、公钥等配置
RsaHelper.php	RSA签名验签工具类，处理签名相关逻辑
create_order.php	创建订单示例，演示统一下单接口调用
query_order.php	查询订单示例，演示订单查询接口调用

文件	说明
<code>notify.php</code>	异步通知处理，接收并处理支付平台的回调通知
<code>return.php</code>	同步跳转处理，展示支付结果页面

接口说明

1. 统一下单接口

请求地址：`https://pay.uesa.cc/api/pay/create`

请求方式：POST

必传参数：

参数名	类型	说明
pid	Int	商户ID
method	String	接口类型：web/jump/jsapi/app等
type	String	支付方式
out_trade_no	String	商户订单号
notify_url	String	异步通知地址
return_url	String	同步跳转地址
name	String	商品名称
money	String	金额（元，最多2位小数）
clientip	String	用户IP地址
timestamp	String	时间戳（10位秒级）
sign	String	签名
sign_type	String	签名类型（固定为RSA）

2. 订单查询接口

请求地址：`https://pay.uesa.cc/api/pay/query`

请求方式：POST

必传参数：

参数名	类型	说明
pid	Int	商户ID
trade_no	String	平台订单号（与out_trade_no二选一）
out_trade_no	String	商户订单号（与trade_no二选一）
timestamp	String	时间戳（10位秒级）
sign	String	签名
sign_type	String	签名类型（固定为RSA）

3. 支付结果通知

通知方式：GET

通知参数：

参数名	类型	说明
pid	Int	商户ID
trade_no	String	平台订单号
out_trade_no	String	商户订单号
trade_status	String	交易状态（TRADE_SUCCESS）
money	String	订单金额
type	String	支付方式
timestamp	String	时间戳
sign	String	签名
sign_type	String	签名类型

返回说明： 处理成功后需返回字符串 `success`

签名算法

签名步骤

1. 获取所有非空请求参数
2. 剔除 `sign` 和 `sign_type` 字段
3. 按键名ASCII码升序排序
4. 组合成"参数=参数值"格式，用 `&` 连接
5. 使用商户私钥进行SHA256WithRSA签名
6. 对签名结果进行Base64编码

验签步骤

1. 按签名步骤1-4获取待签名字符串
2. 对签名进行Base64解码
3. 使用平台公钥进行RSA验签

签名示例

```
php
// 参数排序和拼接
$params = [
    'money' => '100.00',
    'name' => '商品',
    'out_trade_no' => 'ORDER123',
    'pid' => '1001',
    'timestamp' => '1721206072',
];

// 排序后: money=100.00&name=商品
// &out_trade_no=ORDER123&pid=1001&timestamp=1721206072
$signString = RsaHelper::getSignString($params);

// 生成签名
$sign = RsaHelper::generateSign($params, $privateKey);
```

支付方式

调用值	描述
TRX	TRX
USDT_TRC20	USDT-TRC20
EVM_ETH_ETH	ETH
EVM_ETH_USDT_ERC20	USDT-ERC20
EVM_ETH_USDC_ERC20	USDC-ERC20
EVM_Polygon_USDT_ERC20	USDT-Polygon
EVM_Polygon_USDC_ERC20	USDC-Polygon

? 常见问题

1. 签名失败

原因：

- 私钥格式不正确
- 参数未正确排序
- 包含空值参数
- 编码不一致

解决方案：

- 确保私钥包含完整的BEGIN和END标记
- 使用 `RsaHelper::getSignString()` 方法自动处理排序
- 过滤掉空值参数

2. 验签失败

原因：

- 公钥格式不正确
- 签名已被篡改
- 参数顺序不一致

解决方案：

- 确保使用正确的平台公钥
- 使用 `RsaHelper.verifySign()` 方法进行验签
- 保持参数原始顺序

3. 回调通知未收到

原因：

- notify_url地址不可访问
- 防火墙拦截
- 服务器配置问题

解决方案：

- 确保回调地址外网可访问
- 检查服务器防火墙设置
- 查看服务器访问日志

4. 订单重复处理

原因：

- 支付平台可能多次发送同一通知

解决方案：

- 在业务逻辑中检查订单状态
- 已处理的订单直接返回 `success`

注意事项

安全相关

1. 私钥保护

- 商户私钥必须妥善保管，不要泄露
- 不要将私钥提交到公开的代码仓库
- 定期更换密钥对

2. 签名验证

- 所有回调通知必须验签
- 验证商户ID是否匹配
- 验证订单金额是否一致
- 验证时间戳防止重放攻击

3. 回调处理

- 必须检查订单状态，避免重复处理
- 使用数据库事务保证数据一致性
- 记录详细的日志便于排查问题

业务相关

1. 订单号

- 确保商户订单号唯一
- 建议使用时间戳+随机数生成

2. 金额处理

- 金额格式为字符串，最多2位小数
- 使用 `bccomp()` 比较金额，避免浮点数精度问题

3. 时间戳

- 必须使用10位整数（秒级）
- 建议验证时间戳有效期（如 ± 5 分钟）

4. 网络请求

- 设置合理的超时时间
- 处理网络异常情况
- 建议使用HTTPS

技术支持

如有问题，请联系支付平台客服获取技术支持。

许可证

本SDK仅供商户对接使用，请勿用于其他用途。

最后更新： 2024年