

接口规则

协议规则 调用API 必须遵循以下规则：

	规则
传输方式	为保证数据传输的安全性，必须采用HTTPS 传输
POST 请求	Content-Type 设置为：application/x-www-form-urlencoded
字符编码	统一采用UTF-8 编码
签名算法	HMAC-SHA256，具体签名方法参考4.3.2 签名算法
加密算法	对称加密，使用PKCS5Padding 方式填充的3DES 加密

一、参数规定

1. 服务器地址

正式环境地址： 需要技术提供

测试环境地址： 需要技术提供

2. 时间戳

标准北京时间，时区为东八区，自 1970 年 1 月 1 日 0 点 0 分 0 秒以来的秒数。值为毫秒级，需要转换成秒(10 位数字)。

3. 商户订单号

商户支付订单号由商户自定义生成，允许使用 字母、数字、下划线 等，请勿使用汉字或全角等特殊字符。商户订单号由商户保持唯一性（ 建议根据当前系统时间加随机序列来生成订单号 ）。超时情况下重新发起一笔打款需要使用原订单号，避免重复支付。已支付的订单不能重新发起打款。

4. header 参数

header 里 必须 传入商户号 mch-id (mch-id 由平台提供)

字段名	类型	描述
mch-id	string	商户号 (必须)

5. body 参数

注意：不管是 post 请求还是 get 请求，都固定传以下参数，api 列表里的接口参数加密以后的

数据对应body 里的data。

字段名	类型	描述
data	string	使用Base64 编码加密以后的数据
nonce_str	string	生成的随机数，主要保证签名不可预测
timestamp	int	10 位数字的时间戳
sign	string	通过签名算法计算得出的签名值
sign_type	string	签名类型，默认为 HMAC-SHA256，固定填写 sha256

二、安全规范

1、加密

加密方式

加密使用PKCS5Padding 方式填充的3DES 算法，加密模式为 CBC，最后使用Base64 编码。

加密算法（以PHP为例）

```
<?php
namespace App\Common\Utils;
class Des
{
    /**
     * @var 秘钥
     */
    protected $key;

    /**
     * @var 偏移量
     */
    protected $iv;

    public function construct($key, $iv = null) {
        $this->key = $key;
        $this->iv = empty($iv) ? substr($key, 0, 8) : $iv;
    }

    /**
     *加密
     *
     * @param $string
     * @return string
     */
    public function encrypt($string) {
        return base64_encode(openssl_encrypt($string, 'des-ede3-cbc', $this->key, OPENSSL_RAW_DATA, $this->iv));
    }

    /**
     *解密
     *
     * @param $data
     * @return string
     */
    public function decrypt($data) {
        return openssl_decrypt(base64_decode($data), 'des-ede3-cbc', $this->key, OPENSSL_RAW_DATA, $this->iv);
    }
}
```

示例demo

这里加密的 appkey 为 **cDzaGNjTG2Nh3WJj9PKlfXOO**（开放平台获取）

加密代码调用：

```
$appkey = 'cDzaGNjTG2Nh3WJj9PKlfXOO';
$des = new Des($appkey);
$params = [
    'trade_no' => '20190527141700040203014817',
    'mch_id' => 'Y0vmARJOXzNH',
    'chan_id' => '11001100', 'name'
=> '盖伦',
    'card_no' => '6227000011112222333',
    'phone' => '13888888888',
    'pay_amount' => 10000,
    'pay_remark' => '打款备注'
];
$encrypted = $des->encrypt(json_encode($params));
dump($encrypted);
```

加密的打印结果如下：

```
8XgnaloAPNpCu0QW7lhRIYwHE+aA7OftalzQlZljIBl8oFYuCwH1Drpb/5VTELnQZ92Z+bMPQto/MOlRIC+0rE
vp4wRDlYjPtAoZtgblkyBJA2fcJyfjOz3eu4jnFhky44qkDn7sbrKe9aSCnqBvr66PYBxL+R4RaymzJgoa7tK6
XsYEjAYWff2wWBQMqOnl3mHYd2jkdA/qYe9Rrl6rawoALp7rptk0mNF9SBpkZa+DsuriFidopXXuWvDgkvqnwz
RL+nRlv5SwiKIZeeZIRHokareU9c/gxOhIovWViMM=
```

解密代码调用：

```
$appkey = 'cDzaGNjTG2Nh3WJj9PKlfXOO';
$des = new Des($appkey);
$encrypted
'8XgnaloAPNpCu0QW7lhRIYwHE+aA7OftalzQlZljIBl8oFYuCwH1Drpb/5VTELnQZ92Z+bMPQto/MOlRIC+0rE
Evp4wRDlYjPtAoZtgblkyBJA2fcJyfjOz3eu4jnFhky44qkDn7sbrKe9aSCnqBvr66PYBxL+R4RaymzJgoa7tK
6XsYEjAYWff2wWBQMqOnl3mHYd2jkdA/qYe9Rrl6rawoALp7rptk0mNF9SBpkZa+DsuriFidopXXuWvDgkvqnw
zRL+nRlv5SwiKIZeeZIRHokareU9c/gxOhIovWViMM=';
$decrypted = $des->decrypt($encrypted);
dump($decrypted);
dump(json_decode($decrypted, true));
```

其他语言代码请参考加密规则自行实现，如需协助调试请联系对应的技术。

2、签名

2.1 签名方式

第一步，将所有要发送的数据，使用URL 键值对的格式

（data=value1&nonce_str=value2×tamp=value3&key=appsecret）拼接成待签名的字符

串string

第二步，计算string 字符串的hmac，得到的哈希值即为签名

2.2 签名算法（以PHP 为例）

```
<?php

namespace App\Common\Util;

class Sign
{
    /**
     * @var 用于签名的密钥
     */
    protected $appSecret;

    public function __construct($appSecret)
    {
        $this->appSecret = $appSecret;
    }

    /**
     * 获取签名
     *
     * @param string $data
     * @param string $nonce_str
     * @param int $timestamp
     * @return string
     */
    public function getSign(string $data, string $nonce_str, int $timestamp)
    {
        $string = "data=$data&nonce_str=$nonce_str&timestamp=$timestamp&key=$this->appSecret";
        return hash_hmac('sha256', $string, $this->appSecret);
    }
}
```

示例demo

```
{
  "trade_no": "20190527141700040203014817",
  "mch_id": "Y0vmARJOXzNH",
  "chan_id": "11001100",
  "name": "\u76d6\u4f26",
  "card_no": "622700001111222333",
  "phone": "13888888888",
  "pay_amount": 10000,
  "pay_remark": "\u6253\u6b3e\u5907\u6ce8"
}

array:8 [
  "trade_no" => "20190527141700040203014817"
  "mch_id" => "Y0vmARJOXzNH"
  "chan_id" => "11001100"
  "name" => "盖伦"
  "card_no" => "622700001111222333"
  "phone" => "13888888888"
  "pay_amount" => 10000
  "pay_remark" => "打款备注"
]
```

这里用于签名的 appsecret 为 **KkdnLVESDV15baENs9rkk11b**，（开放平台获取）

假设发送的数据如下：

```
data
'8XgnaloAPNpCu0QW7lhRIYwHE+aaA7OftalzQlZljiB18oFYuCwH1Drpb/5VTELnQZ92Z+bMPQto/M0lRIC+0rEvp4wRDlYjPtAoZtgb1kyBJA2fcJyFjOz3eu4jnFhky44qkDn7sbrKe9aSCnqBvr66PYBxL+R4RaymzJgoa7tK6XsYEjAYWff2wWBQMqoN13mHYd2jkdA/qYe9Rrl6rawoALp7rptk0mNF9SBpkZa+DsuriFidopXXuWvDgkvqnwzRL+nRlv5SwiKIZeeZIRHokareU9c/gxOhIovWViMM='
nonce_str      : 'QiWW6nAk'
timestamp      : 1546272000
```

第一步，拼接字符串string，注意参数的顺序固定为

data=value1&nonce_str=value2×tamp=value3&key=appsecret

```
$string
'data=8XgnaloAPNpCu0QW7lhRIYwHE+aaA7OftalzQlZljiB18oFYuCwH1Drpb/5VTELnQZ92Z+bMPQto/M0lRIC+0rEvp4wRDlYjPtAoZtgb1kyBJA2fcJyFjOz3eu4jnFhky44qkDn7sbrKe9aSCnqBvr66PYBxL+R4RaymzJgoa7tK6XsYEjAYWff2wWBQMqoN13mHYd2jkdA/qYe9Rrl6rawoALp7rptk0mNF9SBpkZa+DsuriFidopXXuWvDgkvqnwzRL+nRlv5SwiKIZeeZIRHokareU9c/gxOhIovWViMM=&nonce_str=QiWW6nAk&timestamp=1546272000&key=KkdnLVESDV15baENs9rkk11b'
```

第二步，计算哈希值得到签名

0c2a5d6ff4b9f575f04c345f64886648309e919a0599be426213e42a1962a266

3、接口响应

接口返回值统一为json 格式，返回值包含内容如下：

字段名	类型	描述
code	int	业务响应的code 码（非HTTP 状态码）
data	string	业务数据，具体的字段内容参考各个API 接口的返回值
msg	string	返回值的状态描述

三、接口列表

1、请求打款(/v1/payment)

1.1 请求参数

请求头

字段名	类型	描述
mch-id	string	商户号（必须）
Content-Type	string	application/x-www-form-urlencoded; charset=UTF-8

请求体

字段名	类型	描述
data	string	加密数据
nonce_str	string	随机字符串
timestamp	string	时间戳10 位
key	String	平台提供key

sign	string	签名数据
------	--------	------

data 加密前参数内容详解:

名称	类型	必须	描述	长度
mch_id	String	是	商户ID，平台提供	50
trade_no	String	否	交易订单号，必须唯一	50
chan_id	String	是	通道ID，平台提供	20
name	String	否	收款人姓名	50
card_no	String	否	收款人银行账号	100
id_card	String	否	收款人身份证号码	18
phone	String	是	收款人手机号	11
pay_amount	int	是	支付金额，单位分	20

1.2 接口响应

名称	示例值	描述
code	200	状态
msg	ok	详细信息
data	详细数据	返回数据

示例:

```
{
  "code":200,
  "msg":"ok",
  "data":{
    "trade_no":"202005221212123460",
    "pay_no":"20200522140630026803036356",
    "pay_amount":"1.00"
  }
}
```

1.3 打款结果异步通知

商户需要实现该接口，平台调用该接口,完成出款状态通知。商户接到通知结果后需要返回 `success` 标识，表示接受成功。回调频率 通知的间隔频率一般是：4m,10m,10m,1h,2h,6h,15h。

示例：

```
{
  'mch_id':'USifqvl1f6PZ',
  'chan_id':'1100100',
  'trade_no':'202005221212123463', //订单号
  'pay_type':'银行卡',
  'pay_no':'20200523110847022403039706', //打款流水号
  'pay_amount':'1', //打款金额，单位分
  'id_card':'132434198909872234', //收款人身份证号
  'name':'张三', //收款人姓名
  'card_no':'6214830128387777', //收款人银行卡号
  'phone':'13426278765', //收款人手机号
  "state_id":10, //打款结果 ID
  "state":"打款中", //打款结果描述
  "created_at":"2020-05-22 14:06:37",
  "finished_at":"2020-05-22 14:06:37"
}
```

1.4 打款结果状态码

状态码	描述
-----	----

8	打款成功
9	打款失败
10	打款中
11	证件号码格式错误
12	未发起签约
13	收款人姓名与已签约数据不符
14	手机号格式错误
15	收款账户格式错误
18	打款金额不能为 0
21	超过日打款金额
22	超过月打款金额
23	超过季打款金额
24	超过年打款金额
41	商户余额不足

2、打款查询接口(/v1/payment-query)

2.1 请求参数

请求头

字段名	类型	描述
mch-id	string	商户号（ 必须， 平台提供 ）
Content-Type	string	application/x-www-form-urlencoded; charset=UTF-8

请求体

字段名	类型	描述
data	string	加密数据

nonce_str	string	随机字符串
timestamp	string	时间戳10 位
key	String	平台提供key
sign	string	签名数据

data 加密前参数内容详解:

名称	类型	必须	描述	长度
mch_id	String	是	商户号	50
trade_no	string	是	交易订单号（和pay_no 二选一）	50
pay_no	string	是	流水号（和trade_no 二选一）	50
is_encrypt	int	是	是否返回密文(0 否, 1 是)	

2.2 接口响应

示例如下:

```
{
  "code":200,
  "msg":"ok",
  "data":{
    "mch_id":"USifqvl1f6PZ",
    "chan_id":"10110110",
    "trade_no":"202005221212123460",
    "pay_type":"银行卡",
    "pay_no":"20200522140630026803036356",
    "pay_amount":1, //付款金额
    "id_card":"131432193002073414",
    "name":"张三",
    "card_no":"6214830110002334",
    "phone":"13226209317",
    "state_id":10, //打款结果ID
    "state":"打款中", //打款结果描述
    "created_at":"2020-05-22 14:06:37",
    "finished_at":"2020-05-22 14:06:37"
  }
}
```

3、账户余额查询(/v1/merchant-account)

3.1 请求参数

请求头

字段名	类型	描述
mch-id	string	商户号（ 必须， 平台提供 ）
Content-Type	string	application/x-www-form-urlencoded; charset=UTF-8

请求体

字段名	类型	描述
data	string	加密数据
nonce_str	string	随机字符串
timestamp	string	时间戳10 位
key	String	平台提供key
sign	string	签名数据

data 加密前参数内容详解:

名称	类型	必须	描述	长度
mch_id	String	是	商户号， 平台提供	50

3.2 接口响应

{

```
"code":200,
"msg":"ok",
"data":[
  {
    "chan_id":"10110110", //通道 ID
    "bank_amount":50000, //单位分
  }
]
```

4、发起签约(/v1/signup)

4.1 请求参数

请求头

字段名	类型	描述
mch-id	string	商户号（ 必须， 平台提供 ）
Content-Type	string	application/x-www-form-urlencoded; charset=UTF-8

请求体

字段名	类型	描述
data	string	加密数据
nonce_str	string	随机字符串
timestamp	string	时间戳10 位
key	String	平台提供key
sign	string	签名数据

data 加密前参数内容详解:

名称	类型	必须	描述	长度
mch_id	String	是	商户号，平台提供	50
chan_id	string	是	通道ID，平台提供	50
proto_id	int	是	协议ID，平台提供	10
name	string	是	用户名	50
Id_card	string	是	身份证号	18
phone	string	是	手机号	11

4.2 接口响应

```
{
  "code":200,
  "msg":"ok",
  "data":null
}
```

5、签约结果查询(/v1/sign-query)

5.1 请求参数

请求头

字段名	类型	描述
mch-id	string	商户号（ 必须，平台提供 ）
Content-Type	string	application/x-www-form-urlencoded; charset=UTF-8

请求体

字段名	类型	描述
data	string	加密数据
nonce_str	string	随机字符串
timestamp	string	时间戳10 位
key	String	平台提供key
sign	string	签名数据

data 加密前参数内容详解:

名称	类型	必须	描述	长度
mch_id	String	是	商户号，平台提供	50
chan_id	string	是	通道ID，平台提供	50
Id_card	string	是	身份证号	18

5.2 接口响应

```
{
  "code":200,
  "msg":"ok",
  "data":{
    "mch_id":"USifqvl1f6PZ",
    "chan_id":"10110110",
    "id_card":"130435199002073214",
    "agree_id":"20200605105642058306018287", //协议号，通过此数值进行协议下载
    "is_check":true,
    "name":"刘立鑫",
    "is_sign":true //签约结果 true or false
  }
}
```

6、商户余额查询(/v1/merchant-account)

6.1 请求参数

请求头

字段名	类型	描述
mch-id	string	商户号（ 必须，平台提供 ）
Content-Type	string	application/x-www-form-urlencoded; charset=UTF-8

请求体

字段名	类型	描述
data	string	加密数据
nonce_str	string	随机字符串
timestamp	string	时间戳10 位
key	String	平台提供key
sign	string	签名数据

data 加密前参数内容详解:

名称	类型	必须	描述	长度
mch_id	String	是	商户号，平台提供	50

6.2 接口响应

```
{
  "code":200,
  "msg":"ok",
  "data":{
    "mch_id":"USifqvl1f6PZ",
    "chan_id":"10110110",
    "amount":110000, //账户余额 单位分
    "available_balance":1000000//可用余额 单位分
  }
}
```