**二维码收款接口文档**

收款支付

接口开发手册V1.0

文档修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018-09-13 | 1.0.1 | 整理接口文档 | 丁敏 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

[1 文档介绍 2](#_Toc487465322)

[1.1 特别声明 2](#_Toc487465323)

[1.2 阅读对象 2](#_Toc487465324)

[1.3 产品说明 3](#_Toc487465325)

[1.4 接口签名算法详解 3](#_Toc487465326)

[1.6请求参数详解 3](#_Toc487465327)

[2 接口说明 3](#_Toc487465328)

[2.1 通讯方式说明 3](#_Toc487465329)

[2.2 接口列表 4](#_Toc487465330)

[2.3收款下单接口(POST) 4](#_Toc487465331)

[2. 3.1收款下单接口调用请求说明 4](#_Toc487465332)

[请求参数说明 4](#_Toc487465333)

[2.3.3收款下单接口返回参数说明 4](#_Toc487465334)

[2.4收款回调接口 5](#_Toc487465335)

[2.4.1接口调用请求说明 5](#_Toc487465336)

[2.4.2请求参数说明 5](#_Toc487465337)

[返回说明 5](#_Toc487465338)

[2.5交易查询接口(GET) 6](#_Toc487465339)

[2.5.1接口调用请求说明 6](#_Toc487465340)

[2.5.2请求参数说明 6](#_Toc487465341)

[2.7.3返回参数说明 6](#_Toc487465342)

[2.6商户通道余额查询接口(GET) 7](#_Toc487465343)

[2.6.1接口调用请求说明 7](#_Toc487465344)

[2.6.2请求参数说明 7](#_Toc487465345)

[2.6.3返回参数说明 7](#_Toc487465346)

[1 附录 8](#_Toc487465347)

[1.1 系统返回码 8](#_Toc487465348)

### 文档介绍

#### 特别声明

未得到云付支付的书面许可，不得为任何目的、以任何形式或手段（包括但不限于机械的或电子的）复制或传播本文档的任何部分。对于本文档涉及的技术和产品，商标、版权或其它知识产权。除非得到云付支付的书面许可协议，本文档不授予这些专利、商标、版权或其它知识产权的许可。

本文档因产品功能示例和描述的需要，所使用的任何人名、企业名和数据都是虚构的，并仅限于云付支付支付内部测试使用，不等于云付支付有对任何第三方的承诺和宣传。

#### 阅读对象

贵公司的技术部门的开发、维护及管理人员。

他们应具备以下基本知识：

(1).了解网站设置和网页制作方法；

(2).了解 HTML 语言以及 ASP、.NET、 PHP 或 JAVA 等开发语言；

(3).了解信息安全的基本概念**。**

#### 产品说明

本开发手册对云付支付-收款系统注册 API 进行详细的描述，通过该指南可以对本系统有全面的了解,使商户技术人员尽快掌握本系统的接口,并能够在本系统上进行开发。

#### 接口签名算法详解

PHP版本签名算法

<\?php

/\*\*

\* 签名算法

\* @param unknown $key\_id S\_KEY（商户KEY）

\* @param unknown $array 例子：$array = array('amount'=>'1.00','out\_trade\_no'=>'2018123645787452');

\* @return string

\*/

function sign ($key\_id, $array)

{

$data = md5(number\_format($array['amount'],2) . $array['out\_trade\_no']);

$key[] ="";

$box[] ="";

$pwd\_length = strlen($key\_id);

$data\_length = strlen($data);

for ($i = 0; $i < 256; $i++)

{

$key[$i] = ord($key\_id[$i % $pwd\_length]);

$box[$i] = $i;

}

for ($j = $i = 0; $i < 256; $i++)

{

$j = ($j + $box[$i] + $key[$i]) % 256;

$tmp = $box[$i];

$box[$i] = $box[$j];

$box[$j] = $tmp;

}

for ($a = $j = $i = 0; $i < $data\_length; $i++)

{

$a = ($a + 1) % 256;

$j = ($j + $box[$a]) % 256;

$tmp = $box[$a];

$box[$a] = $box[$j];

$box[$j] = $tmp;

$k = $box[(($box[$a] + $box[$j]) % 256)];

$cipher .= chr(ord($data[$i]) ^ $k);

}

return md5($cipher);

}

?>

JAVA版本：

public class Sign {

public static String decry\_RC4(byte[] data, String key) {

if (data == null || key == null) {

return null;

}

return asString(RC4Base(data, key));

}

public static String decry\_RC4(String data, String key) {

if (data == null || key == null) {

return null;

}

return new String(RC4Base(HexString2Bytes(data), key));

}

public static byte[] encry\_RC4\_byte(String data, String key) {

if (data == null || key == null) {

return null;

}

byte b\_data[] = data.getBytes();

return RC4Base(b\_data, key);

}

public static String encry\_RC4\_string(String data, String key) {

if (data == null || key == null) {

return null;

}

return toHexString(asString(encry\_RC4\_byte(data, key)));

}

private static String asString(byte[] buf) {

StringBuffer strbuf = new StringBuffer(buf.length);

for (int i = 0; i < buf.length; i++) {

strbuf.append((char) buf[i]);

}

return strbuf.toString();

}

private static byte[] initKey(String aKey) {

byte[] b\_key = aKey.getBytes();

byte state[] = new byte[256];

for (int i = 0; i < 256; i++) {

state[i] = (byte) i;

}

int index1 = 0;

int index2 = 0;

if (b\_key == null || b\_key.length == 0) {

return null;

}

for (int i = 0; i < 256; i++) {

index2 = ((b\_key[index1] & 0xff) + (state[i] & 0xff) + index2) & 0xff;

byte tmp = state[i];

state[i] = state[index2];

state[index2] = tmp;

index1 = (index1 + 1) % b\_key.length;

}

return state;

}

private static String toHexString(String s) {

String str = "";

for (int i = 0; i < s.length(); i++) {

int ch = (int) s.charAt(i);

String s4 = Integer.toHexString(ch & 0xFF);

if (s4.length() == 1) {

s4 = '0' + s4;

}

str = str + s4;

}

return str;// 0x表示十六进制

}

private static byte[] HexString2Bytes(String src) {

int size = src.length();

byte[] ret = new byte[size / 2];

byte[] tmp = src.getBytes();

for (int i = 0; i < size / 2; i++) {

ret[i] = uniteBytes(tmp[i \* 2], tmp[i \* 2 + 1]);

}

return ret;

}

private static byte uniteBytes(byte src0, byte src1) {

char \_b0 = (char) Byte.decode("0x" + new String(new byte[]{src0})).byteValue();

\_b0 = (char) (\_b0 << 4);

char \_b1 = (char) Byte.decode("0x" + new String(new byte[]{src1})).byteValue();

byte ret = (byte) (\_b0 ^ \_b1);

return ret;

}

private static byte[] RC4Base(byte[] input, String mKkey) {

int x = 0;

int y = 0;

byte key[] = initKey(mKkey);

int xorIndex;

byte[] result = new byte[input.length];

for (int i = 0; i < input.length; i++) {

x = (x + 1) & 0xff;

y = ((key[x] & 0xff) + y) & 0xff;

byte tmp = key[x];

key[x] = key[y];

key[y] = tmp;

xorIndex = ((key[x] & 0xff) + (key[y] & 0xff)) & 0xff;

result[i] = (byte) (input[i] ^ key[xorIndex]);

}

return result;

}

public static void main(String[] args) {

}

}

#### 1.6请求参数详解

### 接口说明

#### 通讯方式说明

在未特别注明的情况下,所有的接口均可采用 HTTP 的POST 方式发起请求。

#### 接口列表

|  |  |
| --- | --- |
| **序号** | **接口名称** |
| 2.3 | 收款下单接口 |

#### 2.3收款下单接口(POST)

#### 2. 3.1收款下单接口调用请求说明

http://pay.rdnux.cn/gateway/index/checkpoint.do

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **参数长度** | **是否签名** | **是否必填** | **参数说明** |
| account\_id | 商户ID、在平台首页右边获取商户ID | STRING(11) | 是 | 是 | 商户ID |
| content\_type | 请求过程中返回的网页类型，text或json  机 | STRING(10) | 是 | 是 | 请求过程中返回的网页类型，text或json |
| thoroughfare | 初始化支付通道，目前通道：wechat\_auto（公开版微信）、alipay\_auto（公开版支付宝）、service\_auto（服务版微信/支付宝） | STRING(10) | 是 | 是 | wechat\_auto |
| type | 支付类型，该参数在服务版下有效（service\_auto），其他可为空参数，微信：1，支付宝：2 | STRING(20) | 是 | 是 | 1 |
| out\_trade\_no | 订单信息，在发起订单时附加的信息，如用户名，充值订单号等字段参数 | STRING | 是 | 是 | 2018062668945 |
| robin | 轮训，2：开启轮训，1：进入单通道模式 | STRING | 是 | 是 | 2 |
| keyId | 设备KEY，在公开版列表里面Important参数下的DEVICE Key一项，如果该请求为轮训模式，则本参数无效，本参数为单通道模式 | STRING(20) | 是 | 是 | 785D239777C4DE7739 |
| amount | 支付金额，在发起时用户填写的支付金额 | STRING |  | 是 | 1.00 |
| callback\_url | 异步通知地址，在支付完成时，本平台服务器系统会自动向该地址发起一条支付成功的回调请求 |  |  |  | http://pay.rdnux.cn/callback\_url/pay.do |
| success\_url | 支付成功后网页自动跳转地址，仅在网页类型为text下有效，json会将该参数返回 |  |  |  | http://pay.rdnux.cn/index/doc/getQrcode.do |
| error\_url | 支付失败时，或支付超时后网页自动跳转地址，仅在网页类型为text下有效，json会将该参数返回 |  |  |  | http://pay.rdnux.cn/index/doc/getQrcode.do |
| sign | 签名算法，在支付时进行签名算法，详见[《云付支付签名算法》](http://pay.iswoole.com/index/doc/sign.do) |  |  |  | d92eff67b3be05f5e61502e96278d01b |

#### 请求参数说明

#### 2.3.3收款下单接口返回参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数说明** |
| code | 返回码 | 请参照附录：1.1返回码列表 |
| msg | 签名信息 | 返回信息的详细说明 |
| data | 集合 | 集合数据 |
| data - order\_id | 订单ID | 请求支付返回的订单ID |

#### 2.4收款回调接口

#### 2.4.1接口调用请求说明

**回调接收地址，是调用收款接口时传入的地址，在些地址上加上其它请求参数回调。**

#### 2.4.2请求参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **参数说明** |
| account\_name | 商户登录名（会员名），识别是哪个会员名回调过来的请求 | laoniu |
| pay\_time | 支付成功的时间戳（10位） | 1529775437 |
| status | 支付状态，支付状态只有成功一个状态（success） | success |
| amount | |  | | --- | | 支付金额 | | 1.00 |
| out\_trade\_no | 订单信息，在发起订单时附加的信息，如用户名，充值订单号等字段参数 | 2018062312410711888 |
| trade\_no | 交易流水号，由系统生成的交易流水号 | 2018062312410729584 |
| fees | 手续费，本次回调过程产生的手续费用（已经在平台账户中扣除） | 0.04 |
| sign | 签名算法，在支付时进行签名算法，详见《云付支付签名算法》 | d92eff67b3be05f5e61502e96278d01b |
| callback\_time | |  | | --- | | 回调时间，在回调时产生的时间戳（10位） | | 1529775437 |
| type | 当前订单支付类型，1为微信，2为支付宝 | |  | | --- | | 1 | |  | |
| account\_key | |  | | --- | | 商户KEY（S\_KEY） | | 4C61C86ABEBC6243 |

#### 返回说明

收到交易结果后，应返回内容“success”（表示通知成功）。商户在 30 分钟内未收到平台的支付结果通知时，建议应该主动向 平台发起交易信息查询交易（该步骤可选），查询该交易是否成功，可 查询一次或多次。参考“交易结果查询”接口

#### 2.5交易查询接口(GET)

#### 2.5.1接口调用请求说明

**微 信订单接口URL（公开版v1.0）：**http://pay.rdnux.cn/gateway/pay/automaticWechat.do

**支付宝订单接口URL（公开版v1.0）：**http://pay.rdnux.cn/gateway/pay/automaticAlipay.do

**服务版订单接口URL（服务版v1.0）：**http://pay.rdnux.cn/gateway/pay/service.do

#### 2.5.2请求参数说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数含义** | **参数长度** | **是否签名** | **是否必填** | **参数说明** |
| id | 订单ID、在创建订单时会返回订单ID号、注意，只有json网页类型下才会有返回值 | STRING(10) | 是 | 是 | 10000 |
| content\_type | 请求过程中返回的网页类型，text或json | STRING(11) | 是 | 是 | json |

#### 2.7.3返回参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | | **参数含义** | **参数说明** |
| code | | 返回码 | 请参照附录：1.1返回码列表 |
| msg | | 签名信息 | 返回信息的详细说明 |
| data | | 返回订单信息集合 |  |
| data. id | | 通道方订单号 | 订单ID |
| data. wechat\_name | | 收款设备ID | 收款设备ID |
| data. creation\_time | 创建时间 | | 10位时间戳 |
| data. status | | 订单状态 | 订单状态 1 等待下发支付二维码 2未支付 3订单超时 4已支付 |
| data. amount | | 订单金额 | 1.00 |
| data. success\_url | | 成功后返回的链接 | 成功后返回的链接 |
| data. error\_url | | 失败返回的链接 | 失败返回的链接 |
| data. out\_trade\_no | | 订单号 | 交易订单号，用户名，备注信息 |
| data. qrcode | | 支付二维码 | 支付二维码 |

#### 附录

#### 系统返回码

|  |  |
| --- | --- |
| **返回码** | **说明** |
| 200 | 成功 |
| -1 | 支付金额不正确 |
| -3 | 签名失败 |