**SUNEEE 支付网关接入手册**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修订人** | **内容** | **修改时间** |
| V 1.0.0 | kevin.qin(伯约)  卓尔  卓尔  众星  卓尔 | 支付网关平台技术方案(初稿)  1. 银联  2. 支付宝  3. 微信支付  4. 工商银行 | 2015-09-06  2015-09-20  2015-09-20  2015-09-20  2016-01-25 |
| V1.0.1 | 腾渊 | 5.修订生产环境地址 | 2017-03-10 |
| V1.0.1 | 道隐 | 6.审核并补充修订业务时序图 | 2017-03-10 |

目录

[1 支付网关简介 2](#_Toc429384772)

[1.1 SE支付网关简介 2](#_Toc429384773)

[1.2 SE支付网关基本条件 2](#_Toc429384774)

[2 支付接口 2](#_Toc429384775)

[2.1 支付入口 2](#_Toc429384776)

[2.1 支付代码例子 6](#_Toc429384777)

# 支付网关简介

## SE支付网关简介

在国内外的市场上比较流行的第三方支付平台很多, paypal, visa, 银联, 支付宝, 财付通， 微支付, 招商银行, 交通银行, 各种银行的信用卡支付等等.

SUNEEE支付网关（简称SE支付网关）是一个集合各种类型的第三方支付接口的中间件平台, 它可以避免商户的站点去实现多种交易方式的对接, 而只需要和SE支付网关对接。这种模式有以下优点. 1. 脱离了商户站点，减少了并发压力. 2. 作为一个独立的平台,提高了系统的易用性和维护性. 3. 从运营角度,可以单独作为一个第三方的支付平台去运作。

目前我们根据国内市场主要去实现 银联, 支付宝, 微信支付, 招商银行 4个支付接口。

## SE支付网关必要条件

凡是接入SE支付网关的商户, 必须在SUNEEE支付网关平台进行开户, 并分配商户以下两个参数:

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 参数描述 |
| 1 | se\_payment\_code | string(32) | SE支付网关的授权号,网关通过该值去识别商户信息. |
| 2 | se\_private\_key | string(32) | SE 支付网关的秘钥,商户站点需要该秘钥进行对签名字段加密. |

## SE支付网关账户条件

SE网关支付的必要条件

### SE支付网关无账户情况

**无账户情况：是指商家只需在象翌平台开通授权账户即可使用，而不必申请额外相关账户条件，简单方便。账户信息都是象翌平台账户。**

### SE支付网关有账户情况

#### 支付宝账户信息

商家在申请支付宝账户后，登录到支付宝【商家服务】签约相应的支付宝产品，其中包括【即时到账】和【手机网站支付】,得到以下相关信息：

|  |  |
| --- | --- |
| Partner | 合作身份者id |
| seller\_email | 收款支付宝账号 |
| Key | 安全检验码，以数字和字母组成的32位字符 |
| cacert.pem | 支付宝相关证书文件1 |
| alipay\_public\_key.pem(wap版需要) | 支付宝相关证书文件2 |
| rsa\_private\_key.pem(wap版需要) | 支付宝相关证书文件3 |

#### 微信账户信息

1. 商户需【申请微信支付】开通后，需提供以下微信相关信息：

|  |  |
| --- | --- |
| APPID | 应用ID |
| APPSECRET | 应用密钥 |
| MCHID | 商户号 |
| KEY | 商户支付密钥Key |
| apiclient\_cert.pem  apiclient\_key.pem  rootca.pem | 微信相关证书文件 |

1. 商户微信公众号需按照下面设置：

第一步：登陆微信公众平台-》按要求开通微信支付



第二步：开通后，进入开发者中心，找到“网页账号”，点击修改，配置授权回调页面域名opg.weilian.cn



第三步：填写微信支付授权目录



#### 银联账户信息

商户需通过CHINAPAY相关人员申请，申请后会得到以下相关信息：

A. web支付

|  |  |
| --- | --- |
| MerPrK.key | 以MerPrK开头的秘钥文件（用来生成商户号） |
| 分支机构号 | 如 00198660 |

B. Wap支付

|  |  |
| --- | --- |
| 商户号 | 如898450148160123 |
| 商户私钥证书 | .pfx后缀文件 |

#### 工商银行账户信息

商户需通过工行相关人员申请，申请后会得到以下相关信息：

a.支付接口

|  |  |
| --- | --- |
| 商户私钥文件 | 以.key为后缀的文件 |
| 商户公钥文件 | 以.cer为后缀的文件 |
| pass | 密钥口令 |
| 商户代码 | 如2102EC24340826 |
| 商户账号 | 如2102109019300405009 |

b.退款和转账

|  |  |
| --- | --- |
| 集团CIS号 | 如210290001534188 |
| 归属银行编号 |  |
| 证书ID |  |
| 商户代码 | 如2102EC24340826 |
| 商户账号 | 如2102109019300405009 |

# 支付接口

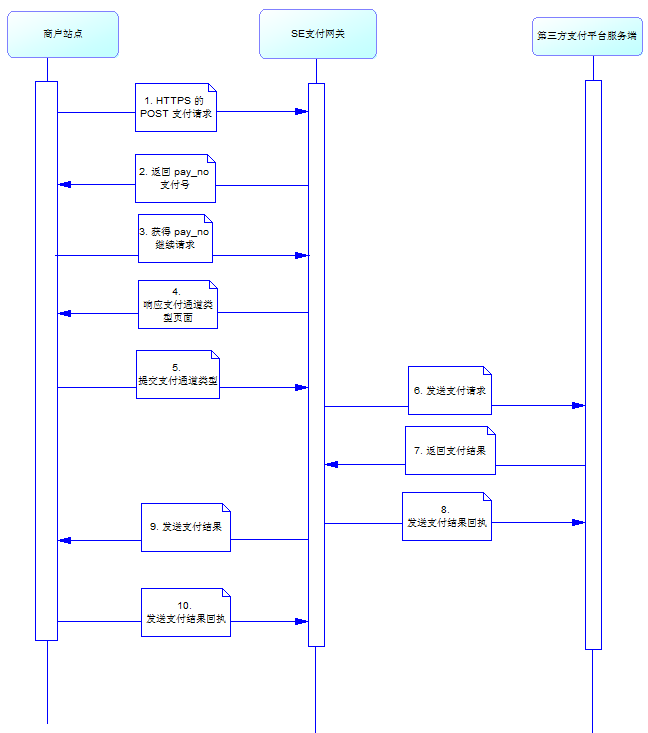
支付网关平台是以HTTP 基于SSL 层加密传输, 也就是传统的HTTPS 协议方式的, 所有的数据都是进行秘钥加密的数据, 以防止在途中被非法窃取篡改。

## 入口地址

~~测试地址: test.opg.suneee.com (172.19.6.125)，见最新的测试地址。~~

<http://pay.weilian.cn/> 正式环境

## 支付接口流程



## 支付接口说明

1. **商户站点调用SE支付网关的入口:**

[http://pay.weilian.cn/payment/pay](http://pay.weilian.cn/)

该入口主要是商户站点发送的支付请求地址, 此时支付网关验证参数成功, 同步返回一个支付号(**pay\_no)**.

**支付入口的参数说明:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 是否必须 | 参数描述 | |
| 1 | bill\_id | string(32) | 是 | 商户站点提交的订单ID号.该字段会和SE支付网关的支付号关联. |
| 2 | bill\_type\_id | String(32) | 是 | 业务类型,考虑区分商户站点多业务ID, 该参数是由商户填写 | |
| 3 | subject | String(32) | 否 | 交易标题,默认 "SE商城支付" | |
| 4 | se\_payment\_code | String(32) | 是 | 商家在SE支付网关的授权编码. | |
| 5 | description | String(256) | 否 | 详情信息 | |
| 6 | amount | float | 是 | 用来支付现金的总额度. | |
| 7 | return\_url | String(256) | 否 | 同步回调页面 | |
| 8 | notify\_url | String(256) | 否 | 异步回调页面 | |
| 9 | show\_url | String(256) | 否 | 仅当支付方式为支付宝手机支付是为必填 | |
| 9 | payment\_type\_id | int(11) | 是 | 当先选择好支付方式时  1-支付宝  2-银联支付  3-微信支付  4-工行支付 | |
| 10 | created\_ts | datetime | 是 | 发送时间,会根据此参数用来检测是否超时.格式(时间戳) | |
| 11 | se\_sign | String(32) | 是 | SE支付网关签名md5 签名 | |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + bill\_id + bill\_type\_id + amount + created\_ts) + se\_private\_key) | | | | |

1. **商户站点再次回调SE支付网关接口(地址一样,参数不一样):**

[http://pay.weilian.cn/payment/pay](http://pay.weilian.cn/)

把第一步生成并返回的支付号(**pay\_no)**.传递到该接口地址, 验证通过后会响应支付方式选择页面, 是让用户去选择具体的支付方式(银联/支付宝/微信支付/招商银行支付)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 是否必须 | 参数描述 |
| 1 | pay\_no | string(32) | 是 | 商户站点提交的订单ID号.该字段会和SE支付网关的支付号关联. |
| 2 | created\_ts | datetime | 是 | 发送时间,会根据此参数用来检测是否超时.格式(时间戳) |
| 3 | se\_sign | string(32) | 是 | SE支付网关签名md5 签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + pay\_no + created\_ts) + se\_payment\_key) | | | |

1. **SE支付网关回调商户站点.**

支付完成后支付网关会调用商户站点的 return\_url 进行回调，商户技术人员可在此编写业务逻辑

return\_url: 由商户自己定义

**返回给到商户站点参数**

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 参数描述 |
| 1 | pay\_no | string(32) | SE支付网关支付号.该支付号是交易完成后最后返回给商户站点,需要商户站点保存该号,以后用于退款参数传递. |
| 2 | result | string(16) | SUCCESS/FAILED/XXX其他状态 |
| 3 | created\_ts | datetime | 创建时间. 格式(时间戳). |
| 4 | sign | String(32) | 签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + pay\_no + result + created\_ts) + se\_privatet\_key) | | |

## 支付代码例子(具体详见demo)

### PHP版本

**<?php**

**class DemoController extends PaymentController {**

**//配置**

**public $conf = array(**

**//支付URL**

**'pay\_url' =>'http://dev.opg.suneee.com/opg/payment/pay',**

**//同步回调页面**

**'pay\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/pay\_return\_url',**

**//异步回调页面**

**'pay\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/pay\_notice\_url',**

**//退款URL**

**'refund\_url' =>'http://dev.opg.suneee.com/opg/payment/refund',**

**//同步回调页面**

**'refund\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/refund\_return\_url',**

**//异步回调页面**

**'refund\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/refund\_notice\_url',**

**//商家授权号**

**'se\_payment\_code' =>'af70891aa4b8cc068dfd2af60c6270d7',**

**//商家秘钥**

**'se\_payment\_key' =>'2f5cc53277fcf5f24c8ecb93c366b9a1',**

**);**

**/\*\***

**\* 直接测试支付接口**

**\*/**

**publicfunction test\_pay()**

**{**

**//1. 组装参数请求**

**$params = array(**

**'se\_payment\_code' => $this->conf['se\_payment\_code'],**

**'bill\_id'=>time(),**

**'bill\_type\_id'=>'订单',**

**'amount' => 0.01,**

**'created\_ts' => time(),**

**'return\_url' => $this->conf['return\_url'],**

**'notify\_url' => $this->conf['notify\_url'],**

**'show\_url' => $post['pay\_show\_url'],**

**'payment\_type\_id' => $post['payment\_type\_id'],**

**);**

**$params['subject'] = '测试订单';**

**$params['description'] = '订单明细, a商品, b商品';**

**//组合签名**

**$params['se\_sign'] = md5(md5($params['se\_payment\_code'].$params['bill\_id'].$params['bill\_type\_id'].$params['amount'].$params['created\_ts']).$this->conf['se\_payment\_key']);**

**if(!$params['se\_sign'])**

**{**

**die('sign failed!');**

**}**

**$response = $this->curl($this->conf['pay\_url'], $params);**

**//2. 得到参数号并验证**

**if(isset($response) && (strlen($response)!==32 || !preg\_match('/^\w+$/', $response)))**

**{**

**die($response);**

**}**

**$pay\_no = $response;**

**//3. 商家应该保存当前交易号**

**//save\_pay\_no();**

**//4. 重新封装交易号请求**

**$params = array(**

**'se\_payment\_code' => $this->conf['se\_payment\_code'],**

**'pay\_no'=>$pay\_no,**

**'created\_ts' => time()**

**);**

**$params['se\_sign'] = md5(md5($params['se\_payment\_code'].$params['pay\_no'].$params['created\_ts']).$this->conf['se\_payment\_key']);**

**//5. 从$params参数数组中移除不需要的商家授权号参数**

**unset($params['se\_payment\_code']);**

**//6. 以GET方式请求到支付类型选择页面**

**//a. 嵌入式载入**

**//$response = $this->curl($conf['pay\_url'], $params, 'GET');**

**//echo $response;**

**//b. 跳转载入**

**header('Location:'.$this->conf['pay\_url'].'?'.http\_build\_query($params));**

**exit;**

**}**

**/\*\***

**\* 支付同步回调**

**\*/**

**publicfunction pay\_return\_url()**

**{**

**$pay\_no = $\_GET['pay\_no'];**

**$result = $\_GET['result'];**

**$creatd\_ts = $\_GET['created\_ts'];**

**$se\_sign = $\_GET['se\_sign'];**

**//商户自己去处理以上的参数**

**if($result === 'SUCCESS')**

**{**

**//成功处理**

**//建议用户再对签名进行验证一次**

**}**

**else**

**{**

**//失败处理**

**}**

**}**

**/\*\***

**\* 支付异步回调, 和同步回调方式一致,该函数只有第三方系统支持异步才被调用**

**\*/**

**publicfunctionpay\_notify\_url()**

**{**

**$pay\_no = $\_GET['pay\_no'];**

**$result = $\_GET['result'];**

**$creatd\_ts = $\_GET['created\_ts'];**

**$se\_sign = $\_GET['se\_sign'];**

**//商户自己去处理以上的参数**

**if($result === 'SUCCESS')**

**{**

**//成功处理**

**//建议用户再对签名进行验证一次**

**}**

**else**

**{**

**//失败处理**

**}**

**}**

**?>**

# 退款接口

## 入口地址

生产地址: pay.weilian.cn

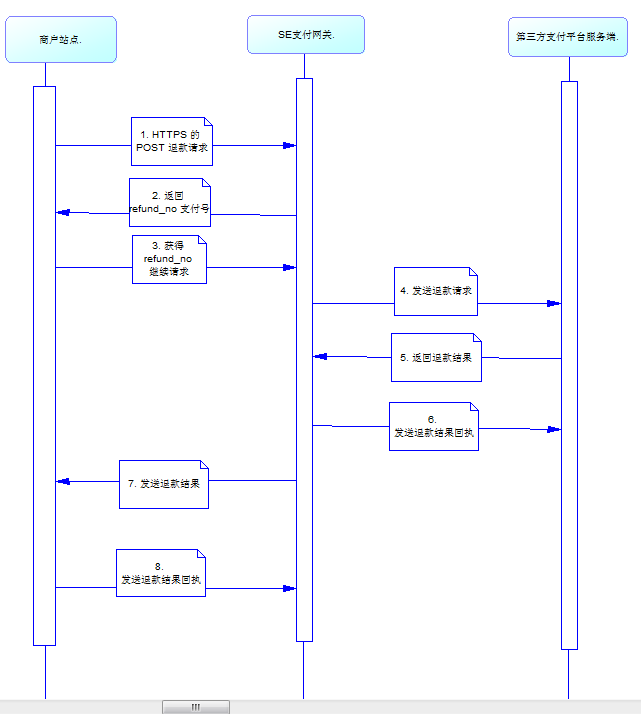
**商户站点调用SE支付网关的退款接口:**

[http://pay.weilian.cn/payment/refund](http://pay.weilian.cn/)

商户站点申请退款并通过审核成功后, 由商户站点对SE支付网关发送一个退款请求,会调用此接口.该接口组装第三方平台接口需要的参数, 调用第三方平台退款接口进行退款.

注意：SE支付网关支持单个支付号多次退款的模式.

## 退款接口流程



## 退款接口说明

1. **商户站点调用SE支付网关的入口:**

[http://pay.weilian.cn/payment/refund](http://pay.weilian.cn/)

该入口是商户站点发送的退款请求地址, 如果支付网关验证参数成功, 同步返回一个退款号(refund\_no**)**

**退款接口参数说明：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 是否必须 | 参数描述 |
| 1 | pay\_no | string(64) | 是 | 支付号,SE支付网关根据支付号 |
| 2 | subject | String(64) | 否 | 交易标题,默认 "SE商城支付" |
| 3 | description | String | 否 | 第三方参数 (预留参数) |
| 4 | amount | Float | 是 | 退款金额.（考虑多次退款）,  只能根据支付成功的支付号去进行退款操作 |
| 6 | return\_url | String(256) | 否 | 同步回调页面 |
| 7 | notify\_url | String(256) | 否 | 异步回调页面 |
| 8 | created\_ts | Datetime | 是 | 创建时间,会根据此参数用来检测是否超时. 格式(时间戳) |
| 9 | se\_sign | String(32) | 是 | SE支付网关签名md5 签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code+pay\_no+amount+created\_ts) + se\_private\_key)  参数没有带支付平台授权号, 需要通过支付网关数据库找到商户的支付平台授权号. | | | |

1. **商户站点再次回调SE支付网关退款接口(地址一样,参数不一样):**

商户站点把流程1获得退款号以参数形式重定向到该接口地址, 验证通过后直接调用第三方的退款接口.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 是否必须 | 参数描述 |
| 1 | refund\_no | 1 | refund\_no | 1 |
| 2 | created\_ts | 2 | created\_ts | 2 |
| 3 | se\_sign | 3 | se\_sign | 3 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + refund\_no + created\_ts) + se\_private\_key) | | | |

1. **SE支付网关回调商户站点.**

退款完成后SE支付网关自动调用商户站点的 return\_url 进行回调

return\_url: 由商户自己定义

**退款接口的返回值:**

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 描述 |
| 1 | pay\_no | String(32) | 支付号 |
| 2 | refund\_no | String(32) | 退款批次号 |
| 3 | result | string(16) | SUCCESS/FAILED/其他错误代码 |
| 4 | created\_ts | Datetime | 创建时间, 格式(时间戳) |
| 5 | sign | String(32) | 数字签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + pay\_no + refund\_no + result + created\_ts) + se\_private\_key) | | |

## 退款入口例子(具体详见demo)

### PHP版本

**class DemoController extends PaymentController {**

**//配置**

**public $conf = array(**

**//支付URL**

**'pay\_url' =>'http://dev.opg.suneee.com/opg/payment/pay',**

**//同步回调页面**

**'pay\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/pay\_return\_url',**

**//异步回调页面**

**'pay\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/pay\_notice\_url',**

**//退款URL**

**'refund\_url' =>'http://dev.opg.suneee.com/opg/payment/refund',**

**//同步回调页面**

**'refund\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/refund\_return\_url',**

**//异步回调页面**

**'refund\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/refund\_notice\_url',**

**//商家授权号**

**'se\_payment\_code' =>'af70891aa4b8cc068dfd2af60c6270d7',**

**//商家秘钥**

**'se\_payment\_key' =>'2f5cc53277fcf5f24c8ecb93c366b9a1',**

**);**

**/\*\***

**\* 直接测试退款接口**

**\*/**

**publicfunction test\_refund()**

**{**

**//1. 组装参数请求**

**$params = array(**

**'se\_payment\_code' => $this->conf['se\_payment\_code'],**

**'pay\_no'=>“”, //suneee支付号**

**'amount' => 0.01,**

**'created\_ts' => time(),**

**'return\_url' => $this->conf[' refund\_return\_url '],**

**'notify\_url' => $this->conf[' refund\_notify\_url '],**

**);**

**$params['subject'] = '退款标题';**

**$params['description'] = '退款详情';**

**//组合签名**

**$params['se\_sign'] = md5(md5($params['se\_payment\_code'].$params['bill\_id'].$params['bill\_type\_id'].$params['amount'].$params['created\_ts']).$this->conf['se\_payment\_key']);**

**if(!$params['se\_sign'])**

**{**

**die('sign failed!');**

**}**

**//2. 请求退款**

**$response = $this->curl($this->conf['refund\_url'], $params);**

**//3. 得到退款号并验证**

**if(isset($response) && (empty($response) || !preg\_match('/^\w+$/', $response)))**

**{**

**die($response);**

**}**

**$refund\_no = $response;**

**//3. 商家应该保存当前退款号**

**//save\_refund\_no();**

**//4. 重新封装退款号请求**

**$params = array(**

**'se\_payment\_code' => $this->conf['se\_payment\_code'],**

**'refund\_no'=>$refund\_no,**

**//'is\_embed'=>true, //是否嵌入式载入方式**

**'created\_ts' => time()**

**);**

**$params['se\_sign'] = md5(md5($params['se\_payment\_code'].$params['refund\_no'].$params['created\_ts']).$this->conf['se\_payment\_key']);**

**//5. 从$params参数数组中移除不需要的商家授权号参数**

**unset($params['se\_payment\_code']);**

**//6. 以GET方式请求到支付类型选择页面**

**//a. 嵌入式载入**

**//$response = $this->curl($conf['refund\_url'], $params, 'GET'); //echo $response;**

**//b. 跳转载入**

**header('Location:'.$this->conf['refund\_url'].'?'.http\_build\_query($params));**

**exit;**

**}**

**/\*\***

**\* 退款同步回调**

**\*/**

**publicfunction refund\_return\_url()**

**{**

**$refund\_no = $\_GET['refund\_no '];**

**$result = $\_GET['result'];**

**$creatd\_ts = $\_GET['created\_ts'];**

**$se\_sign = $\_GET['se\_sign'];**

**//商户自己去处理以上的参数**

**if($result === 'SUCCESS')**

**{**

**//成功处理**

**//建议用户再对签名进行验证一次**

**}**

**else**

**{**

**//失败处理**

**}**

**}**

**/\*\***

**\* 退款异步回调, 和同步回调方式一致,该函数只有第三方系统支持异步才被调用**

**\*/**

**publicfunction refund\_notify\_url()**

**{**

**$refund\_no = $\_GET['refund\_no '];**

**$result = $\_GET['result'];**

**$creatd\_ts = $\_GET['created\_ts'];**

**$se\_sign = $\_GET['se\_sign'];**

**//商户自己去处理以上的参数**

**if($result === 'SUCCESS')**

**{**

**//成功处理**

**//建议用户再对签名进行验证一次**

**}**

**else**

**{**

**//失败处理**

**}**

**}**

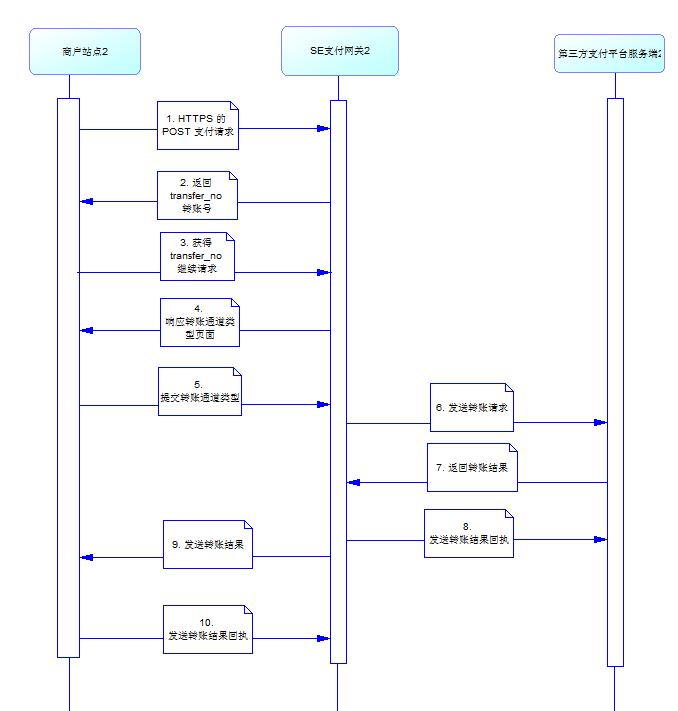
# 转账接口

支付网关平台是以HTTP 基于SSL 层加密传输, 也就是传统的HTTPS 协议方式的, 所有的数据都是进行秘钥加密的数据, 以防止在途中被非法窃取篡改。

## 入口地址

[http://pay.weilian.cn/payment/transfer](http://pay.weilian.cn/)

## 转账接口流程



## 转账接口说明

1. **商户站点调用SE支付网关的入口:**

[http://pay.weilian.cn/payment/transfer](http://pay.weilian.cn/)

该入口主要是商户站点发送的转账请求地址, 此时支付网关验证参数成功, 同步返回一个转账号(transfer**\_no)**.

**转账入口的参数说明:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 是否必须 | 参数描述 |
| 1 | rec\_account\_no | string(34) | 是 | 商户站点提交的转账用户的账号 |
| 2 | rec\_account\_name | string(34) | 是 | 转账用户的账号名称 |
| 3 | subject | String(32) | 否 | 交易标题,默认 "Suneee转账平台" |
| 4 | se\_payment\_code | String(32) | 是 | 商家在SE支付网关的授权编码. |
| 5 | description | String(256) | 否 | 详情信息 |
| 6 | amount | float | 是 | 转账金额. |
| 7 | return\_url | String(256) | 否 | 同步回调页面 |
| 8 | notify\_url | String(256) | 否 | 异步回调页面 |
| 9 | payment\_type\_id | int(11) | 是 | 当先选择好支付方式时 |
| 10 | created\_ts | datetime | 是 | 发送时间,会根据此参数用来检测是否超时.格式(时间戳) |
| 11 | se\_sign | String(32) | 是 | SE支付网关签名md5 签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + rec\_account\_no + amount + created\_ts) + se\_private\_key) | | | |

1. **商户站点再次回调SE支付网关接口(地址一样,参数不一样):**

[http://pay.weilian.cn/payment/transfer](http://pay.weilian.cn/)

把第一步生成并返回的转账号(transfer**\_no)**.传递到该接口地址, 验证通过后会响应转账方式选择页面, 是让用户去选择具体支持的转账方式(银联/支付宝/微信支付/工商银行支付)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 是否必须 | 参数描述 |
| 1 | transfer\_no | string(32) | 是 | 商户站点提交的订单ID号.该字段会和SE支付网关的支付号关联. |
| 2 | created\_ts | datetime | 是 | 发送时间,会根据此参数用来检测是否超时.格式(时间戳) |
| 3 | se\_sign | string(32) | 是 | SE支付网关签名md5 签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + transfer\_no + created\_ts) + se\_private\_key) | | | |

1. **SE支付网关回调商户站点.**

转账完成后支付网关会调用商户站点的 return\_url 进行回调，商户技术人员可在此编写业务逻辑

return\_url: 由商户自己定义

**返回给到商户站点参数**

|  |  |  |  |
| --- | --- | --- | --- |
| NO. | 参数名称 | 类型 | 参数描述 |
| 1 | transfer\_no | string(32) | SE支付网关转账号.该转账号是交易完成后最后返回给商户站点,需要商户站点保存该号,以后用于退款参数传递. |
| 2 | result | string(16) | SUCCESS/FAILED/XXX其他状态 |
| 3 | created\_ts | datetime | 创建时间. 格式(时间戳). |
| 4 | sign | String(32) | 签名 |
| MD5的32位加密签名方式 | md5( md5(se\_payment\_code + transfer\_no + result + created\_ts) + se\_private\_key) | | |

## 转账代码例子(具体详见demo)

### PHP版本

**<?php**

**class DemoController extends PaymentController {**

**//配置**

**public $conf = array(**

**//支付URL**

**'pay\_url' =>'http://dev.opg.suneee.com/opg/payment/pay',**

**//同步回调页面**

**'pay\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/pay\_return\_url',**

**//异步回调页面**

**'pay\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/pay\_notice\_url',**

**//退款URL**

**'refund\_url' =>'http://dev.opg.suneee.com/opg/payment/refund',**

**//同步回调页面**

**'refund\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/refund\_return\_url',**

**//异步回调页面**

**'refund\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/refund\_notice\_url',**

**//转账URL**

**'transfer\_url' =>'http://dev.opg.suneee.com/opg/payment/transfer',**

**//同步回调页面**

**'transfer\_return\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/transfer\_return\_url',**

**//异步回调页面**

**'transfer\_notify\_url' =>'http://dev.opg.suneee.com/opg/payment/demo/transfer\_notify\_url',**

**//商家授权号**

**'se\_payment\_code' =>'af70891aa4b8cc068dfd2af60c6270d7',**

**//商家秘钥**

**'se\_payment\_key' =>'2f5cc53277fcf5f24c8ecb93c366b9a1',**

**);**

**/\*\***

**\* 直接测试转账接口**

**\*/**

**Public function test\_transfer()**

**{**

**//1. 组装参数请求**

**$params = array(**

**'se\_payment\_code' => $this->conf['se\_payment\_code'],**

**'rec\_account\_no'=>’6222022102013545167’,**

**'rec\_account\_name'=>’亮妙虐’,**

**'amount' => 0.01,**

**'created\_ts' => time(),**

**'return\_url' => $this->conf['return\_url'],**

**'notify\_url' => $this->conf['notify\_url'],**

**'payment\_type\_id' => ‘4’,**

**);**

**$params['subject'] = '测试订单';**

**$params['description'] = '订单明细, a商品, b商品';**

**//组合签名**

**$params['se\_sign'] = md5(md5($params['se\_payment\_code'].$params['rec\_account\_no'].$params['amount'].$params['created\_ts']).$this->conf['se\_payment\_key']);**

**if(!$params['se\_sign'])**

**{**

**die('sign failed!');**

**}**

**$response = $this->curl($this->conf['transfer\_url'], $params);**

**//2. 得到参数号并验证**

**if(isset($response) && (empty($response) || !preg\_match('/^\w+$/', $response)))**

**{**

**die($response);**

**}**

**$transfer\_no = $response;**

**//3. 商家应该保存当前交易号**

**//save\_pay\_no();**

**//4. 重新封装交易号请求**

**$params = array(**

**'se\_payment\_code' => $this->conf['se\_payment\_code'],**

**'transfer\_no'=>$transfer\_no,**

**'payment\_type\_id'=>$params['payment\_type\_id'],**

**'created\_ts' => time()**

**);**

**$params['se\_sign'] = md5(md5($params['se\_payment\_code'].$params['transfer\_no'].$params['created\_ts']).$this->conf['se\_payment\_key']);**

**//5. 从$params参数数组中移除不需要的商家授权号参数**

**unset($params['se\_payment\_code']);**

**//6. 以GET方式请求到支付类型选择页面**

**//a. 嵌入式载入**

**//$response = $this->curl($conf['pay\_url'], $params, 'GET');**

**//echo $response;**

**//b. 跳转载入**

**header('Location:'.$this->conf[transfer\_url'].'?'.http\_build\_query($params));**

**exit;**

**}**

**?>**