云闪付开放平台免密支付对接接口文档

一、 总述

1） 流程

1. 接入方定时更新 backendToken
2. 接入方通过用户授权获取 code
3. 获取 accesstoken
4. 访问指定接口

2) backendToken

基础服务令牌， backendToken 为 OAUTH2 授权用，其有效期前设置为 7200 秒，接入方放入缓存，定期更新即可。

a) 获取 backendToken

URL： https://open.95516.com/open/access/1.0/backendToken

输入参数说明：

参数	是否必须	说明
appId	是	接入方的唯一标识
nonceStr	是	生成签名的随机字符串
timestamp	是	生成签名的时间戳
signature	是	签名值，签名因子包括(appId, secret, nonceStr, timestamp)

secret 为给接入方分配的密钥。

说明：

1. 签名算法, nonceStr 如何生成请参考本文档最后一章的 FAQ；
2. timestamp 生成签名的时间戳，接入方生成从 1970 年 1 月 1 日 00:00:00 至今的秒数,即当前的时间，单位为秒；
3. 以下所有接口签名都可参考以上两点。

返回说明：

参数	描述
backendToken	OAUTH2 后台接口调用凭证
expiresIn	backendToken 接口调用凭证超时时间，单位（秒）

注意： backendToken 的有效期 expiresIn 现阶段返回 7200 秒，请做好缓存，频繁调用云闪付会将其列为黑名单拒绝访问。

二、OAUTH2 用户授权

1） 使用场景

基于商务约定，接入方可以通过如下 scope 进行云闪付授权。

upapi_contract:用户签约授权，会弹出授权页面。

2） 对接步骤

第一步：获取 code

H5 接入方跳转到此页面，请求获取 code（客户端通过 SDK 获取，请参考 SDK 相关文档、DEMO）：

https://open.95516.com/s/open/noPwd/html/open.html?appId=APPID&redirectUri=REDIRECTURI&responseType=code&scope=SCOPE&planId=123&state=STATE

强烈建议：跳转回调 redirectUri 使用 https 链接来确保授权 code 的安全性。

参数	是否必须	说明
appId	是	接入方的唯一标识
redirectUri	是	授权后重定向的回调地址，请使用 urlencode 对链接进行处理
responseType	是	返回类型，请填写 code
scope	是	应用授权作用域
planId	是	接入方定义的签约协议模板 Id，用于在签约页面中展示签约内容
state	否	重定向后会带上 state 参数，开发者可以填写 a-zA-Z0-9 的参数值，最多 128 字节

登录之后页面将跳转至 redirectUri ?code=CODE&state=STATE

参数 redirectUri?state=STATE&errmsg=XXXXYYYY

注意： 返回的 code 是通过 encodeURIComponent 经过 URL 编码的，接入方需要通过函数 decodeURIComponent 解码去获取 code.

第二步： 获取 accessToken 和 openId

商户后台请求云闪付后台 URL

https://open.95516.com/open/access/1.0/token

输入参数说明：

参数	是否必须	说明
appId	是	接入方的唯一标识
backendToken	是	第一章第 3 节获取的 backendToken
code	是	填写第一步获取的 code
grantType	是	接入方直接填写常量字符串 authorization_code

返回说明：

正确时返回的 JSON 数据包如下：

```
{
 “accessToken”:” ACCESSTOKEN”,
 “expiresIn”:” 7200”,
 “refreshToken”:” REFRESHTOKEN”,
 “openId”:” OPENID”,
 “scope”:” SCOPE”
}
```

参数	描述
accessToken	网页授权接口调用凭证
expiresIn	accessToken 接口调用凭证超时时间，单位（秒）
refreshToken	用户刷新 accessToken
openId	用户唯一标识

scope	用户授权的作用域

accessToken 有效期为 1 小时，当 accessToken 超时后，需要重新获取。

三、接口对接

调用如下接口请确保第二章 OAUTH2 已经完成，成功获取 accessToken 和 openId.

1） 申请签约

请求 URL
https://open.95516.com/open/access/1.0/contract.apply
输入参数说明:

参数	是否必须	说明
appId	是	接入方的唯一标识
accessToken	是	OAUTH2 中获取的 accessToken
openId	是	获取 accessToken 时同步获取的 openId
backendToken	是	第一章第 3 节获取的 backendToken
plan_id	是	协议模板 id，由云闪付录入模板并分配给接入方
contract_code	是	接入方侧的签约协议号，由接入方自行生成
moblie	否	接入方上送的手机号
certId	否	接入方上送的用户身份证号

返回参数

参数	说明
contract_code	接入方传来的签约协议号
plan_id	接入方传来的协议模板 id
openid	凭此获取云闪付 APP 中接入方的 appid 下用户的唯一标识 openid
operate_time	操作时间
contract_id	签约成功后，云闪付返回的委托免密支付协议 id，返回数据格式为 AN32

2） 申请解约

请求 URL
https://open.95516.com/open/access/1.0/contract.relieve

请求参数

参数	是否必须	说明
appId	是	接入方的唯一标识
openId	是	OAUTH2 中获取的 openId
backendToken	是	第一章第 3 节获取的 backendToken
contract_id	是	委托免密支付签约成功后由云闪付返回的委托免密支付协议 id
plan_id	是	协议模板 id，由云闪付录入模板并分配给接入方
contract_code	是	商户请求签约时传入的签约协议号，由商户侧生产且须唯一

返回参数

参数	说明
contract_code	接入方传来的签约协议号
plan_id	接入方传来的协议模板 id
openid	凭此获取云闪付 APP 中接入方的 appid 下用户的唯一标识 openid
operate_time	操作时间

3） 通知解约结果

调用第三方回调地址，通知解约结果

通知参数

参数	说明
appId	接入方的唯一标识
timestamp	生成签名的时间戳
nonceStr	生成签名的随机串
operate_time	操作时间
openId	OAUTH2 中获取的 openId
plan_id	接入方传来的协议模板 id
contract_code	接入方传来的签约协议号
signature	请使用银联的公钥验证签名，输出格式为 base64.

返回参数

参数	说明
resp	成功接收通知以后发送｛‘resp’：‘00’｝返回

4）用户状态查询

请求 URL
https://open.95516.com/open/access/1.0/contract.status

请求参数

参数	是否必须	说明
appId	是	接入方的唯一标识
openId	是	OAUTH2 中获取的 openId
backendToken	是	第一章第 3 节获取的 backendToken

返回参数

参数	说明
enable	如无未完成订单，返回 0；如存在未完成订单，返回 1

FAQ

1. 后台请求报文格式如何定义?

回答：

所有后台请求均为 POST，JSON 报文格式，应答报文 JSON 格式如下：

resp: ‘00’表示成功，其他的都表示错误。

msg: 成功或者失败详细描述信息。

params: 具体数据返回，JSON 格式。

2. SHA256 签名算法如何实现?

将所有待签名参数按照字段名的 ASCII 码从小到大排序（字典序）后，使用 URL 键值对的格式（即 key1=value1&key2=value2...）拼接成字符串 string1。 Key, value 均采用原始值，大小写不变，不进行 URL 转义。 最后对拼接字符串 string1 作 sha256 运算出 signature。

signature=sha256(string1)

以 upsdk 签名为例;

- appId=a5949221470c4059b9b0b45a90c81527
- nonceStr=Wm3WZYTPz0wzccnW
- timestamp=1414587457
- url=http://mobile.xxx.com?params=value
- frontToken=sM4AOVdWfPE4DxkXGEs8VMCPGGVi4C3VM0P37wVUCFvkVAy_90u5h9nbSlYy3-Sl-HhTdfl2fzFy1AOcHKP7qg

步骤 1. 对所有待签名参数按照字段名的 ASCII 码从小到大排序（字典序）后，使用 URL 键值对的格式（即 key1=value1&key2=value2...）拼接成字符串 string1:

```
appId=a5949221470c4059b9b0b45a90c81527&frontToken=sM4AOVdWfPE4DxkXGEs8VMCPGGVi4C3VM0P37wVUCFvkVAy_90u5h9nbSlYy3-Sl-HhTdfl2fzFy1AOcHKP7qg&nonceStr=Wm3WZYTPz0wzccnW&
timestamp=1414587457&url=http://mobile.xxx.com?params=value
```

步骤 2. 对 string1 进行 sha256 签名，得到 signature:

```
604147c81de8b7c28e3c2a37de03a1274f2efddd4e5a9fcc5cb98ef6198df3b9
```

签名用的 nonceStr 和 timestamp 必须与请求参数中的 nonceStr 和 timestamp 相同。

3. 如何生成签名用随机字符串 nonceStr?

请参考如下 PHP 实现

```
private function createNonceStr($length = 16) {
    $str = null;
    $strPol = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz";
    $max = strlen($strPol)-1;

    for($i=0;$i<$length;$i++){
        $str.=$strPol[rand(0,$max)];//rand($min,$max)生成介于 min 和 max 两个数之间的一个随机整数
    }

    return $str;
}
```

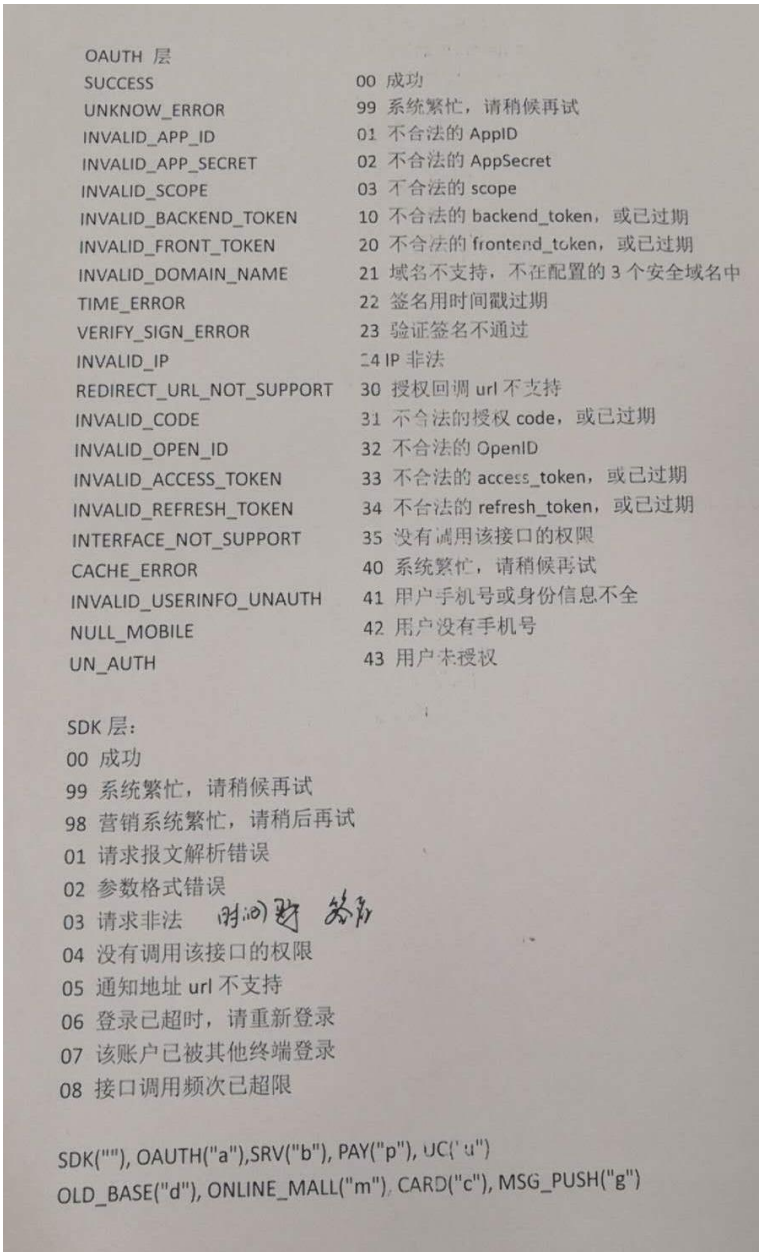
4. UPSDK 签名算法

参考 2. SHA256 签名算法如何实现?

1．签名用的 url 必须是调用 JS 接口为当前网页的 URL，不包含#及其后面部分，

2．接入方只有页面跳转并且 URL 发生改变才需要重新签名，签名与页面 URL（不包含#锚点）相绑定，只要 URL 没变， 签名验证通过以后长时间可用。当然你也可以页面每次加载时都做签名，但这样对于接入方的页面加载以及钱包后台都有影响。

3．如果您的应用基于单页面开发，页面切换通过描点改变（比如#main， #order 等等）， 在页面整个生命周期里面， 您只需要做一次签名就可以了。



OAUTH 层

SUCCESS	00 成功
UNKNOW_ERROR	99 系统繁忙，请稍候再试
INVALID_APP_ID	01 不合法的 AppID
INVALID_APP_SECRET	02 不合法的 AppSecret
INVALID_SCOPE	03 不合法的 scope
INVALID_BACKEND_TOKEN	10 不合法的 backend_token，或已过期
INVALID_FRONT_TOKEN	20 不合法的 frontend_token，或已过期
INVALID_DOMAIN_NAME	21 域名不支持，不在配置的 3 个安全域名中
TIME_ERROR	22 签名用时间戳过期
VERIFY_SIGN_ERROR	23 验证签名不通过
INVALID_IP	24 IP 非法
REDIRECT_URL_NOT_SUPPORT	30 授权回调 url 不支持
INVALID_CODE	31 不合法的授权 code，或已过期
INVALID_OPEN_ID	32 不合法的 OpenID
INVALID_ACCESS_TOKEN	33 不合法的 access_token，或已过期
INVALID_REFRESH_TOKEN	34 不合法的 refresh_token，或已过期
INTERFACE_NOT_SUPPORT	35 没有调用该接口的权限
CACHE_ERROR	40 系统繁忙，请稍候再试
INVALID_USERINFO_UNAUTH	41 用户手机号或身份信息不全
NULL_MOBILE	42 用户没有手机号
UN_AUTH	43 用户未授权

SDK 层：

00 成功
99 系统繁忙，请稍候再试
98 营销系统繁忙，请稍后再试
01 请求报文解析错误
02 参数格式错误
03 请求非法
04 没有调用该接口的权限
05 通知地址 url 不支持
06 登录已超时，请重新登录
07 该账户已被其他终端登录
08 接口调用频次已超限

SDK(""), OAUTH("a"),SRV("b"), PAY("p"), UC("u")
OLD_BASE("d"), ONLINE_MALL("m"), CARD("c"), MSG_PUSH("g")