



秒支付**S-Pay**开发手册
支持iOS, **Android**
v2.3

BeeCloud文档: <http://beecloud.cn/doc>
联系我们: support@beecloud.cn
更新日期: 3/9/2015

前期准备

微信支付

申请应用的微信支付功能需要申请并通过微信开放平台的开发者资质认证，申请开发者资质认证的主体必须属于下图中某一类，且为保证快速通过认证，请按照要求如实填写开发者信息。

微信 · 开放平台

1. 同意协议

2. 填写资料

3. 填写发票

4. 支付费用

● 类型（请从下列类型中选择一项）

☐ 企业

☐ 网店商家

☐ 媒体

☐ 政府及事业单位

☒ 其他组织

☒ 其他组织-免费类型（基金会、国外政府机构驻华代表处）

☐ 社会团体（组织机构代码证上机构类型为社团法人）

☐ 民办非企业（组织机构代码证上机构类型为民办非企业）

☐ 其他组织

- [微信APP支付接入商户服务中心](#)
- [微信支付App支付iOS开发文档](#)
- [微信支付App支付Android开发文档](#)

注册微信开放平台

注册地址：<https://open.weixin.qq.com>

申请开发者资质认证



基本资料

开发者资质认证

修改密码

验证状态

未认证

!

查看订单

现在申请

介绍

• 微信开放平台帐号的开发者资质认证提供更安全、更严格的真实性认证、也能够更好的保护企业及用户的合法权益

• 开发者资质认证通过后，微信开放平台帐号下的应用，将获得微信登录、智能接口、公众号第三方平台开发等高级能力

• 认证有效期：一年，有效期最后两个月可申请年审即可续期

• 审核费用：300元

认证通过的状态如下：

基本资料

开发者资质认证

修改密码

验证状态

已认证

✓

查看订单

介绍

• 微信开放平台帐号的开发者资质认证提供更安全、更严格的真实性认证、也能够更好的保护企业及用户的合法权益

• 开发者资质认证通过后，微信开放平台帐号下的应用，将获得微信登录、智能接口、公众号第三方平台开发等高级能力

• 认证有效期：一年，有效期最后两个月可申请年审即可续期

• 审核费用：300元

创建微信应用

- A. 填写应用的基本信息，上传应用图标(28x28，108x108)。
- B. 填写平台信息。选择应用平台，填写各平台对应的应用信息。
- C. 提交审核，等待审核结果，预计7个工作日。审核通过会获得AppID和AppSecret。

创建移动应用

还可创建10个移动应用

应用名称	状态	操作

管理中心 / 创建移动应用

✓

填写基本信息

✓

填写平台信息

3

提交成功

✓


您的应用已成功提交审核，预计在7天内完成审核

审核通过后，你才可以获得AppID和AppSecret，来进行开发

查看管理中心

创建移动应用

还可创建9个移动应用


应用名称	状态	操作
 比克	审核中	查看

申请微信支付功能

点击“查看”，进入应用详情页，申请开通微信支付、需要填写应用基本资料、企业审核资料、财务审核资料，三者分别审核，全部审核通过后，微信官方会通过邮件的方式，发送微信支付需要的要素。注：按照要求如实填写资料，有助于开速通过审核。

创建移动应用

还可创建9个移动应用

应用名称	状态	操作
 BeeCloud	已通过	查看 删除

申请通过后获得partnerID、appkey、支付专用签名串paySignKey、初始密钥partnerKey、安全证书(.pfx)。

在控制台填写微信支付要素



beecloud_der 切换



应用数据



应用分析



导入数据



支付管理



设置



应用信息

BC File

BC Config

微信支付

支付宝

银联支付

微信支付设置

提示:光标焦点移出输入框,内容将自动保存

微信支付管理ID 操作员ID

微信支付管理密码

微信支付Partner ID

微信支付Partner KEY

微信支付APP KEY

微信支付APP ID

微信支付Secret

微信支付证书密码

微信支付证书

微信支付证书

微信支付证书

字段说明:

key	value
微信管理ID	财付通商户后台设置的操作员ID。退款必须
微信管理员密码	财付通商户后台设置的操作员密码。退款必须
微信 PartnerID	商家ID
微信 PartnerKEY	商家私钥
微信 APPKEY	支付签名密钥
微信APPID	微信开放平台申请的APP ID
微信Secret	微信开放平台申请的APP Secret
微信证书密码	安全证书密码，退款必须。微信支付审核通过后，通过短信的方式发送到联系人的手机上，务必妥善保管，不能丢失。
微信证书	安全证书。微信支付审核通过后，微信发送的邮件附件就是安全证书。退款必须

支付宝支付

签约移动支付产品

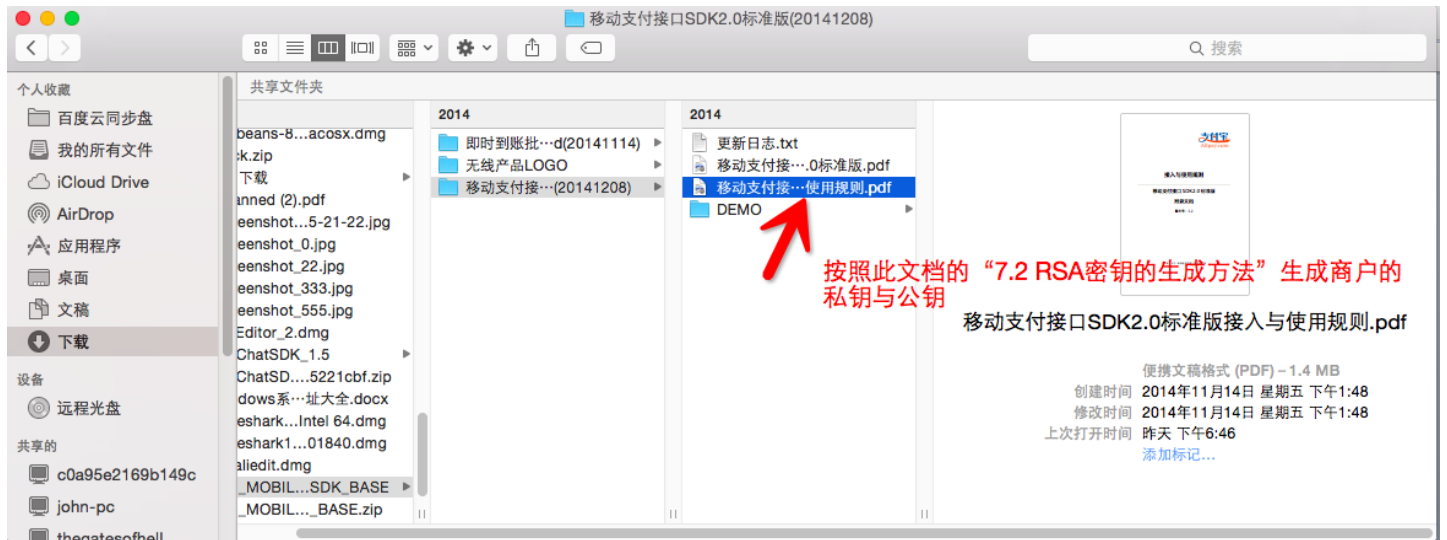
应用集成支付宝快捷支付(无线)，需要在<https://app.alipay.com/container/web/enterpriseIndex.htm>签约支付宝的移动支付产品。



<https://b.alipay.com/order/appInfo.htm?salesPlanCode=2014110308141993&channel=ent>可以查看具体的签约流程



按照支付宝官方签约流程自助签约，审核通过后登录支付宝商户版<https://b.alipay.com/newIndex.htm>->签约管理标签查看会获得支付要素。 点击[支付宝移动支付开发文档](#)，下载技术集成文档。



查看上图中箭头导向的文档的相关章节可以按步骤生成商户密钥与公钥，需要上传商户公钥给支付宝，用于支付宝验证签名。上传商户公钥会获得支付宝的公钥，用于验证支付宝返回数据的合法性。

查看支付要素并添加商户密钥



点击上图中“查看PIDKey”，输入支付密码，可以查看PID等要素。



商家服务

首页

产品商店

营销推广

我的商家服务

如何签约



PID和Key

合作者身份(PID)

[Redacted]

复制

您需要输入支付密码才能查看安全校验码 - KEY

安全设置检测成功! 无需短信校验。

支付宝支付密码:

[忘记密码?](#)

确认



商家服务

首页

产品商店

营销推广

我的商家服务

如何签约



PID和Key

合作者身份(PID)

2088

[Redacted]

复制

合作伙伴密钥管理

安全校验码(Key)

默认加密:

[Redacted]

复制

[重新设置密钥](#)

RSA加密: [查看密钥](#) | [查看支付宝公钥](#)

DSA加密: [添加密钥](#) | [查看支付宝公钥](#)

未上传商户公钥时, 此处显示“添加密钥”

添加公钥成功, 点击“查看支付宝公钥”可以获取支付宝公钥。

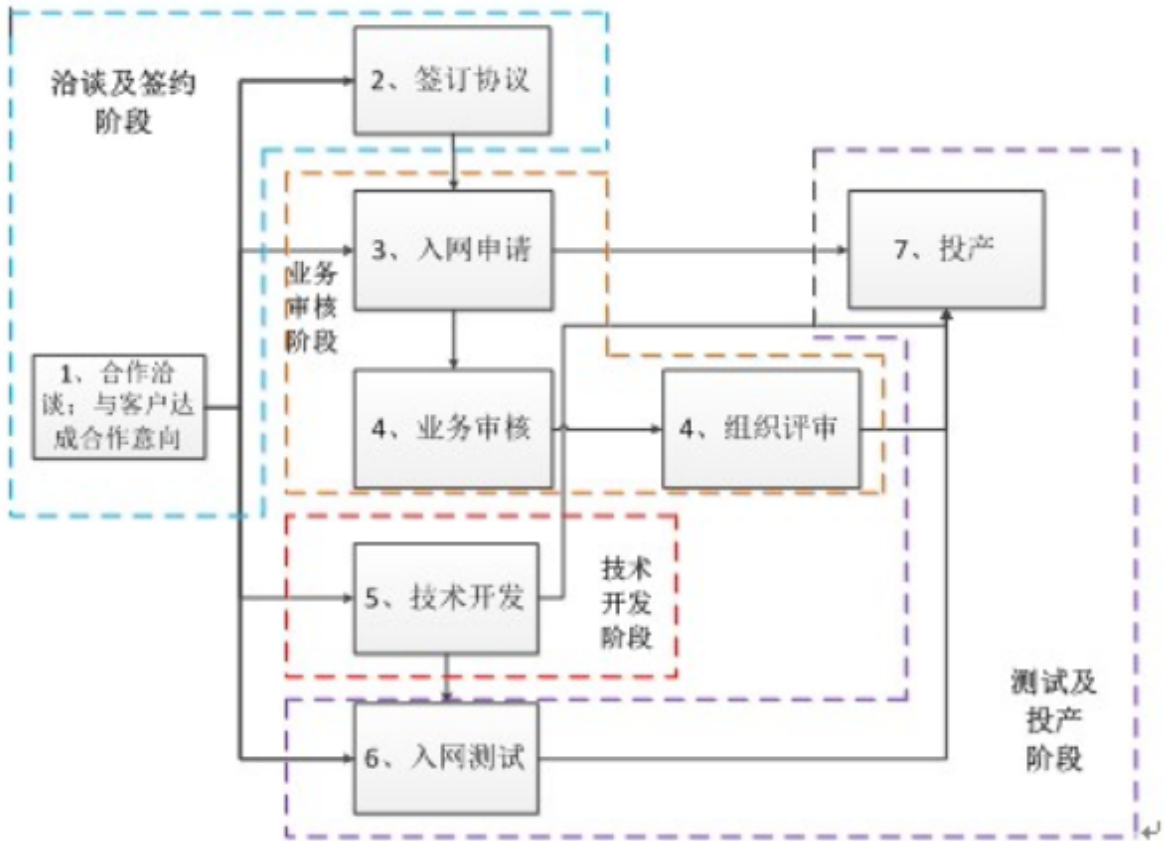
在控制台上传支付宝支付要素

beecloud_der 切换	点击隐藏应用菜单	支付宝设置
应用数据	应用信息	提示:光标焦点移出输入框,内容将自动保存
应用分析	BC File	支付宝Partner ID <input type="text" value=""/> PID
导入数据	BC Config	支付宝卖家ID <input type="text" value=""/>
支付管理	微信支付	支付宝私钥 <div>MIICdgIBADANBgkqhkiG9w0BAQE nI+ovCGzafYlhb17hV3m1rxq4U4v suFhYQeDEsDITs0aukSPWauhRW</div>
设置	支付宝	
	银联支付	

通过openssl工具生成的RSA商户密钥，不能有空格，制表符，换行

银联在线支付

银联在线支付的接入流程大致如下：



- 环节说明：
- 1、合作洽谈，与客户达成入网意向：包括客户选择、确定接入银联业务或产品、形成业务方案。
 - 2、签订协议：客户与银联签订合作协议、入网协议、清算协议、提供代理清算协议等。
 - 3、入网申请：客户以正式方式确认开通的业务信息和要素，以及提供证明客户具备开通该业务相应资质的材料。
 - 4、业务审核：针对某些特定业务，对客户是否具备开展相应业务资质、业务风险是否可接受以及分配客户开展银联业务的特定身份标识、权限的工作。
 - 5、技术开发：包括银联为了支持客户相应业务为进行的开发和开发阶段测试，以及客户为实现与银联系统对接而进行的开发和测试工作。

- 6、入网测试：在正式投产之前，为验证客户和银联系统应用及技术、业务参数配置的正确性、可用性，降低生产系统运营风险而进行的测试。
- 7、投产：包括支持客户业务的生产系统对接成功并技术上线、在系统中业务参数配置生效。
- 以上流程环节根据实际需要不同，可进行选择配置。一般情况下，2，3，4，5，6工作可以并行。

注：入网流程问题发送邮件至：operation@unionpay.com进行业务申请和咨询，银联有专门的人员进行处理。联系电话：021-50362408。在提交入网申请，进入业务审核阶段的时候，银联的技术客服会将您加入到客服全中，方便您及时地解决遇到的问题。审核通过后会收到“商户入网通知参数信息（请注意保密）--（公司名称）”标题的邮件，邮件中包含了一些商户信息，具体内容在邮件的附件中。

在控制台上传银联支付要素。

beecloud_der 切换

应用数据

应用分析

导入数据

支付管理

设置

点击隐藏应用菜单

应用信息

BC File

BC Config

微信支付

支付宝

银联支付

银联支付设置

提示:光标焦点移出输入框,内容将自动保存

银联商户代码

证书密码

银联证书

已上传到服务器

选取证书

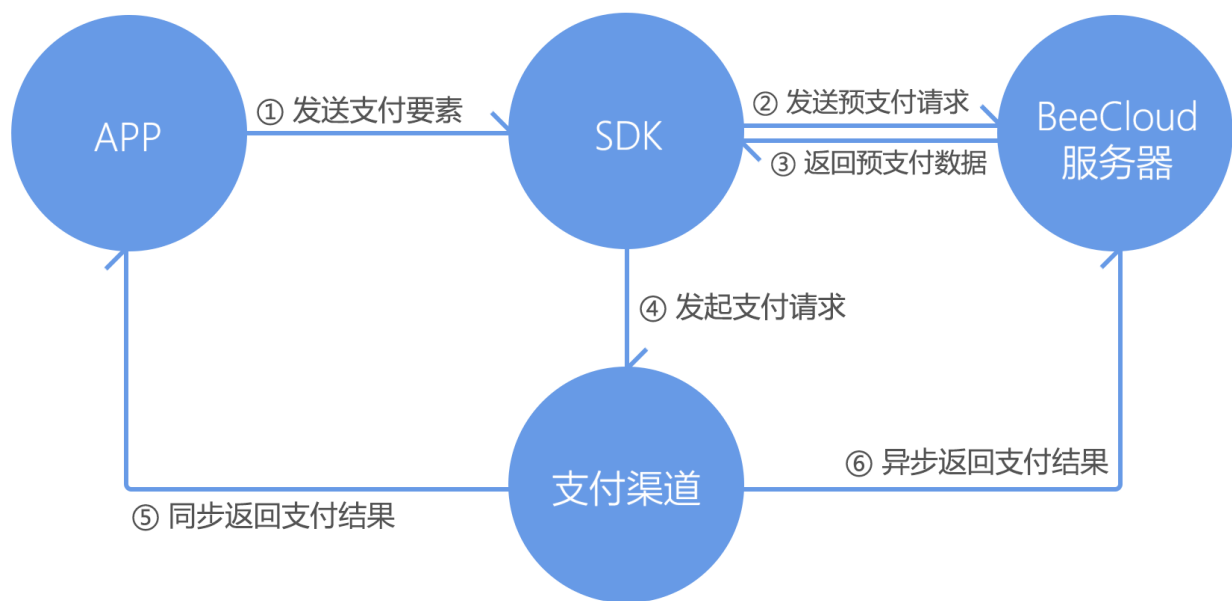
上传证书

隐藏

开始集成

秒支付

秒支付主要实现有关支付方式的相关方法的实现，当前主要支持内置购买、微信支付、支付宝支付等。



注册

两个步骤，2分钟轻松搞定：1. 注册开发者：猛击[这里](#)注册成为BeeCloud开发者。2. 注册应用：使用注册的账号登陆[控制台](#)后，点击"新应用+"创建新应用。

iOS SDK

下载安装SDK

方法一：

1. 下载SDK：猛击[这里](#) 按需下载最新版的BeeCloud SDK。BeeCloud.Framework 为基础包，必须下载，然后根据APP需要的支付渠道选择下面任意支付渠道下载。
2. 安装SDK：把下载下来的所有framework包增加到XCode的项目中（参考[这里](#)）
3. 添加系统framework依赖：
 - CoreGraphics.framework
 - CoreLocation.framework
 - MobileCoreServices.framework
 - Security.framework
 - SystemConfiguration.framework
 - UIKit.framework
 - CoreTelephony.framework

4. 添加系统lib依赖:

- libc++.dylib
- libz.dylib
- libsqlite3.dylib

方法二:

我们提供了更便捷的cocoapods的一键导入集成方式

对初次使用CocoaPods的同学,可以参考[这里](#)

对于熟悉CocoaPods的同学,使用如下命令:

```
pod 'BeeCloud' , :git => 'https://github.com/beecloud/ios-sdk-cocoapods-release.git',  
:branch => 'core'  
pod 'BeeCloud_alipay' , :git => 'https://github.com/beecloud/ios-sdk-cocoapods-release  
.git', :branch => 'aliPay'  
pod 'BeeCloud_wxPay' , :git => 'https://github.com/beecloud/ios-sdk-cocoapods-release.  
git', :branch => 'wxPay'  
pod 'BeeCloud_UPay' , :git => 'https://github.com/beecloud/ios-sdk-cocoapods-release.g  
it', :branch => 'UnionPay'
```

`BeeCloud` 为基础库,必须加入,另外按需加入所需要的支付渠道方式的库, `BeeCloud_alipay` , `BeeCloud_wxPay` , `BeeCloud_UPay` 分别为支付宝,微信和银联。

全局初始化

首先,在需要使用BeeCloud API的地方,import头文件:

```
#import <BeeCloud/BeeCloud.h>
```

其次,在APP启动的时候(在 `applicationDidFinishLaunching` 函数里),初始化BeeCloud需要的App Key, 全局只需调用一次。每个应用对应一个App Key,可在控制台中相应App的 基本信息 里获得:

```
[BeeCloud initWithAppID:@"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx" andAppSecret:@"xxxxxxxx  
x-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"];
```

BeeCloud所有的网络请求默认超时为5秒,如果需要更改(如改为10秒),请调用以下函数,全局只需调用一次:

```
[BeeCloud setNetworkTimeout:10.0];
```

微信支付

BeeCloudSDK提供了快速集成微信支付的服务,其方法主要在 - `BCWXPay` 。

开发者只需调用简单的API就可以在应用中快速集成微信支付功能。

在使用接口之前需要做以下准备工作：

- 1、在微信开发平台注册应用。
- 2、开发者资质认证。
- 3、申请开通应用的支付功能。

注：由于支付功能的相关申请需要15个工作日,建议在应用开发准备阶段完成微信支付功能的相关申请。

具体参考[微信开放平台](#)

微信支付功能申请完成后，按以下步骤“秒”集成微信支付：

第一步:在网页控制台界面填写微信支付相关信息，包括微信开放平台创建应用获得的 `appid`、`appSecret`，以及申请微信支付功能获得的 `partnerID`、`partnerKey`、`appKey`。

第二步:点击项目名称,点击 `Info` 选项卡,在 `URL Types` 选项中,点击“+”，在 `URL Schemes` 中输入微信 `appid`，`Role` 选择 `Editor`，`Identifier`自定义。

第三步:在 `AppDelegate.m` 文件下添加如下代码：

```
.....
#import <BeeCloud/BeeCloud.h>
#import <BCWXPay/BCWXPay.h>
.....
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:
(NSString *)sourceApplication annotation:(id)annotation {
    return [BCWXPay handleOpenUrl:url withBlock:^(BOOL success, NSString *strMsg, NSError
    *error) {
        NSLog(@"strMsg = %@", strMsg);
    }];
}
.....
```

微信支付功能

添加下列代码发起微信支付

```

// @"自制白开水"为:商品描述
// totalFee为:支付金额*以分为单位*
// outTradeNo为需要您自行生成的订单号,*32个字符内、包含数字与字母,确保在商户系统中唯一*可根据[[BCUtil generateRandomUUID] stringByReplacingOccurrencesOfString:@"-" withString:@""]获取;
// traceid为:支付用户ID,必须保证在商户系统中的唯一性
// block支付结果回调
// 通过该方法实现对各种参数的初始化,然后发起微信支付,并跳转到微信。
- (void)wxPay {
    NSString *outTradeNo = [[BCUtil generateRandomUUID] stringByReplacingOccurrencesOfString:@"-" withString:@""];
    [BCWXPAY reqWXPayment:@"自制白开水" totalFee:@"1" outTradeNo:outTradeNo traceID:@"Be
eCloud" payBlock:^(BOOL success, NSString *strMsg, NSError *error) {
        if (success) {
            // 表明微信支付成功
        } else {
            // 表明支付过程中出现错误, strMsg为错误原因
        }
        NSLog(@"%s strMsg = %@", __func__, strMsg);
    }];
}

```

在支付成功后会在云端存放购买记录。

微信退款功能

需要申请退款时,可以调用退款接口来实现.在调用之前需要在云端先配置三个参数:

1. operatorID
2. 安全证书 (在商户审核通过后,收到的财付通邮件中有附件以商户号_XXXX.pfx文件即为安全证书)
3. operatorPassword

在客户端用户通过以下代码发起退款请求:

```

// outTradeNo:用户要退订的订单号
// outRefundNo:您自行生成的退订订单号,*32个字符内、包含数字与字母,确保在商户系统中唯一*可根据[[BCU
til generateRandomUUID] stringByReplacingOccurrencesOfString:@"-" withString:@""]获取;
// refundReason:退款原因
// refundFee:退款金额
// payBlock:发起退款结果回调, 用户在app发起退款请求成功后, 商户会收到退款请求, 商户可在后台确认, 完
成退款操作
- (void)wxRefund:(NSString*)outTradeNo {
    NSString *outRefundNo = [[BCUtil generateRandomUUID] stringByReplacingOccurrencesO
fString:@"-" withString:@""];
    NSLog(@"refund no = %@", outRefundNo);
    [BCWXPAY reqWXRefund:outTradeNo outRefundNo:outRefundNo refundReason:@"不好喝" refu
ndFee:@"1" payBlock:^(BOOL success, NSString *strMsg, NSError *error) {
        if (success) {
            // 退款成功
        } else {
            // 退款失败, strMsg为错误原因
        }
        NSLog(@"%s,%s,%d,%@", __FILE__, __func__, __LINE__, error);
    }];
}

```

在退款成功后, 会在云端生成退款记录.

支付宝支付

当用户在使用BeeCloudSDK进行支付宝支付时, 请注意以下几点:

第一步: 在网页控制台填写支付宝partnerID (PID), 支付宝卖家ID (邮箱或者手机号), 支付宝商家私钥(RSA私钥), 私钥的生成方法见官方文档.

第二步: 点击项目名称, 点击“Info”选项卡, 在“URL Types”选项中, 点击“+”, 在“URL Schemes”中输入应用标识, 如“BCPayDemo”.

第三步: 在AppDelegate.m 文件下添加如下代码:

```

.....
#import <BeeCloud/BeeCloud.h>
#import <BCAlipay/BCAlipay.h>
.....
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:
(NSString *)sourceApplication annotation:(id)annotation {
    return [BCAlipay handleOpenUrl:url withBlock:^(BOOL success, NSString *strMsg, NSE
rror *error) {
        NSLog(@"strMsg = %@", strMsg);
    }];
}

```

支付宝支付功能

当用户确定选择用支付宝进行支付时，可以调用以下接口进行集成支付宝支付。

```
// @param trace_id      支付用户ID，必须保证在商户系统中唯一。可通过trace_id查询订单详情。
// @param out_trade_no  商户系统内部的支付订单号，包含数字与字母，确保在商户系统中唯一
// @param subject       商品的标题/交易标题/订单标题/订单关键字等。该参数最长为 128 个汉字
// @param body          对一笔交易的具体描述信息。如果是多种商品，请将商品描述字符串累加传给body
// @param total_fee     该笔订单的资金总额，单位为RMB-Yuan。取值范围为[0.01,100000000.00]，精确到小数点后两位
// @param scheme        调用支付的app注册在info.plist中的scheme
// @param block         支付结果回调
- (void)aliPay {
    NSString *outTradeNo = [[BCUtil generateRandomUUID] stringByReplacingOccurrencesOfString:@"-" withString:@""];
    [BCAliPay reqAliPayment:@"BeeCloud" outTradeNo:outTradeNo subject:@"自制白开水" body:@"BeeCloud 自制白开水" totalFee:@"0.01" scheme:@"payTestDemo" payBlock:^(BOOL success, NSString *strMsg, NSError *error) {
        if (success) {
            // 表明支付宝支付成功
        } else {
            // 表明支付过程中出现错误，strMsg为错误原因
        }
        NSLog(@"AliPay strMsg = %@", strMsg);
    }];
}
```

支付宝退款功能

当需要进行支付宝退款时，调用以下代码：

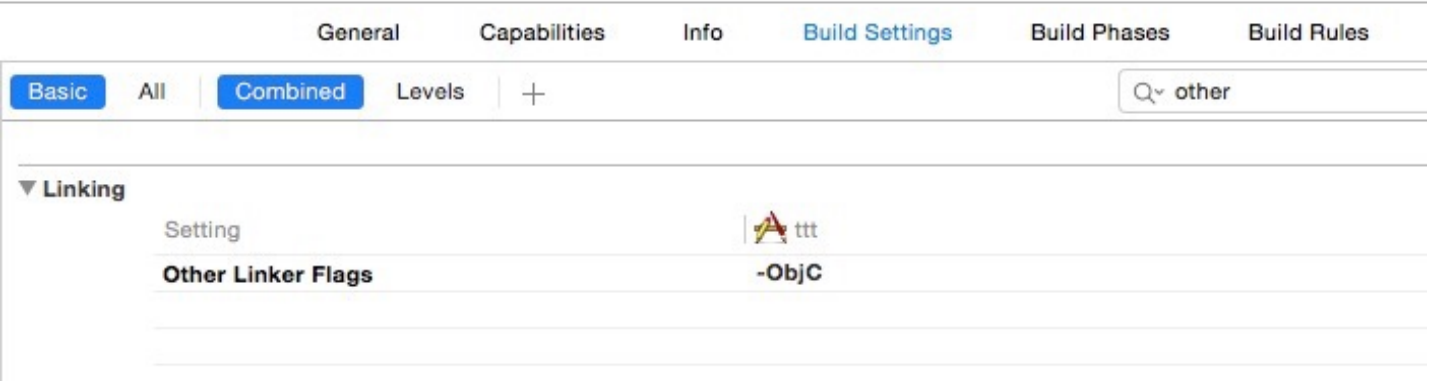
```
// 根据out_trade_no, refund_no, refund_reason, refund_fee查询订单是否可退款，允许退款情况下自动生成预退款订单，否则返回不可退款原因。预退款订单生成成功后，在BeeCloud商户后台对预退款订单进行处理。（订单状态trade_status="@TRADE_SUCCESS"）时支持退款，其他状态下不支持退款。
// @param out_trade_no  商户系统内部的支付订单号，包含数字与字母，确保在商户系统中唯一
// @param refund_no     格式为：退款日期(8位)+流水号(3~8位)。不可重复，且退款日期必须是当天日期(年月日)。流水号可以接受数字或英文字符，建议使用数字，但不可接受“000”。例如：201101120001
// @param refund_fee    退款金额
// @param refund_reason 退款原因
// @param block         退款结果回调
[BCAliPay reqAliRefund:out_trade_no refundNo:dateString refundFee:@"0.01" refundReason:@"不好喝" refundBlock:^(BOOL success, NSString *strMsg, NSError *error) {
    NSLog(@"AliRefund strMsg = %@", strMsg);
}];
```

银联支付

当用户在使用BeeCloudSDK进行银联支付时，需要以下几步：

第一步：在网页控制台填写 银联商户代码，证书密码，上传银联证书

第二步：在 Build Settings的Other Linker Flags选项添加条件“-ObjC”



第三步：在AppDelegate.m 文件下添加如下代码：

```
.....
#import <BeeCloud/BeeCloud.h>
#import <BCAliPay/BCAliPay.h>
.....
```

银联支付功能

当用户确定选择用银联进行支付时，可以调用以下接口进行集成银联支付.

```
// @param trace_id      支付用户ID，必须保证在商户系统中唯一. 可通过trace_id查询订单详情。
// @param body          商品的标题/交易标题/订单标题/订单关键字等。该参数最长为128个汉字
// @param out_trade_no  商户系统内部的支付订单号, 包含数字与字母, 确保在商户系统中唯一
// @param total_fee     支付金额, 以分为单位
// @param viewController 调起银联支付的页面, 一般为self
// @param block         接收支付结果回调
- (void)unionPay{
    NSString *outTradeNo = [[BCUtil generateRandomUUID] stringByReplacingOccurrencesOfString:@"-" withString:@""];
    [BCUnionPay reqUnionPayment:@"BeeCloud01" body:@"自制白开水" outTradeNo:outTradeNo totalFee:@"1" viewController:self payblock:^(BOOL success, NSString *strMsg, NSError *error) {
        if (success) {
            //支付成功
        } else {
            NSLog(@"UnionPay Faild:%@",error.description);
        }
        NSLog(@"Msg:%@", strMsg);
    }];
}
```

银联退款功能

```

// 银联预退款，支持部分退款或者全额退款。如果提供的支付订单的交易状态不支持退款，在block中返回具体的
// 信息;如果支持退款，生成预退款订单，商户在管理后台管理预退款订单。
// @param out_trade_no 商户系统内部的支付订单号,包含数字与字母,确保在商户系统中唯一
// @param refund_fee 退款金额
// @param out_refund_no 商户系统内部的退款订单号,包含数字与字母,确保在商户系统中唯一
// @param refund_reason 退款原因
// @param block 接收预退款订单生成结果
- (void)unionRefund:(NSString*)outTradeNo {
    NSString *outRefundNo = [[BCUtil generateRandomUUID] stringByReplacingOccurrencesOf
fString:@"-" withString:@""];
    [BCUnionPay reqUnionRefund:outTradeNo refundFee:@"0.01" outRefundNo:outRefundNo re
fundReason:@"不好喝" refundBlock:^(BOOL success, NSString *strMsg, NSError *error) {
        if (success) {
            //申请退款成功
        } else {
            //申请退款失败，返回失败原因
            NSLog(@"UnionRefund Faild:%@",error.description);
        }
        NSLog(@"Msg:%@", strMsg);
    }];
}

```

内置购买

BeeCloud提供一个便捷集成的内置购买服务 - BCPay。

开发者只需调用简单的API,就可以快速完成应用的内置购买。

按照苹果官方提供的思路，[官方文档地址](#)开发者需要了解自己要卖的产品是什么，比如是可消费的还是不可消费的，自动订阅的还是free订阅的等等，然后设计产品，购买界面，恢复购买等等。

但总结起来一般都只需要完成下面五步：

1. 去itunes connect添加产品，获得产品的唯一id
2. 设计应用内的购买界面
3. 初始化产品，通过第一步获得的id获取产品实例
4. 完成产品购买的请求以及回调
5. 完成恢复购买的请求以及回调

其中 1, 2步需要开发者自行完成； 3,4,5步可以使用BCPay完成

初始化产品列表

开发者可以自行选择初始化产品的时间，可以在程序启动时，也可以在load自己的购买界面时，通过上文第1步生成的产品id或者多个id构造一个 `NSArray:someProductIds`，然后调用以下代码：

```
// someProductIds 是productId的列表， 如果不传block， 则默认将结果存储在bcPay实例的products(NS
Array)和idToProduct(NSMutableDictionary)
// 用户可以通过- (SKProduct *)getProductById:(NSString *)productId 获得某个productId对应的
product
BCPay * bcPay = [BCPay sharedInstance] initWithProducts:@"someProductIds" withBlock:^(NSAr
ray *products, NSArray *failedIds, NSError *error) {
    //your logic here
}
```

开发者可以传递一个 `nil` 的block，BCPay会将获得的产品列表存储在BCPay实例的 `products`里面，并且可以通过

```
- (SKProduct *)getProductById:(NSString *)productId 方法获得某个productId对应的product
```

购买产品

当用户在购买界面触发购买事件（比如点击购买button）之后，开发者可以添加下面这段代码完成内置购买的请求，block是请求完成后的处理，其中state 0代表购买成功，1代表购买失败，2代表是恢复购买 购买成功后会在云端存储购买记录。

```
// 方法一：通过 (SKProduct *) product实现购买
// product 可以通过- (SKProduct *)getProductById:(NSString *)productId 获得
[[BCPay sharedInstance] purchase:product traceID:@"BeeCloud" withBlock:^(NSString *pro
ductId, NSInteger state, NSError *error) {
    //your code here
}];
// 方法二：直接通过产品ID (productId) 实现购买
[[BCPay sharedInstance] purchaseWithId:productId traceID:@"BeeCloud" withBlock:^(NSStr
ing *productId, NSInteger state, NSError *error) {
    //your code here
}];
```

恢复购买

当用户在购买界面触发恢复购买事件（可以恢复一个或多个或所有产品的购买记录）时，开发者可以添加下面这段代码完成恢复购买的请求，其中productIds为需要恢复购买的产品id列表，state与上相同。

```
// productIds即为要恢复的产品id列表 (NSArray)
[[BCPay sharedInstance] restore:[NSArray arrayWithObject:productId] traceID:@"BeeCloud
" withBlock:^(NSString * productId, NSInteger state, NSError * error) {
    //恢复购买
}];
```

查询订单

当用户需要进行订单查询时，可根据 `traceID` 或者 `out_trade_no` 或者 `refund_no` 分别调用各自的查询接口来进行查询。注：该接口可实现微信、内置购买、支付宝，银联四种渠道的订单查询，同时可支持查询微信、支付宝、银联的退款订单查询操作，提供同步和异步两种查询方式。

```
/**
 * 同步查询支付订单或退款订单。内置购买只支持查询支付订单表。
 *
 * @param key 根据key查询。trace_id,out_trade_no,our_refund_no
 * @param value 要查询的值
 * @param orderType 支付平台的支付订单或者退款订单。
 *
 * @return 符合条件的订单列表
// Refund status
REFUND_START = 0; // 退款开始
REFUND_REJECT = 1; // 退款被商家拒绝
REFUND_ACCEPT = 2; // 退款被商家同意
REFUND_SUCCESS = 3; // 退款成功
REFUND_FAIL = 4; // 退款被渠道拒绝
REFUND_RETRY = 5; // 退款被渠道拒绝，但原因不明， 需要用原退款单号重试
REFUND_NEED_OFFLINE = 6; // 用户银行卡已注销，现金回流到商户账户，需要走线下人工操作
*/
+ (NSArray *)query(xx)OrderByKey:(BCPayOrderKey)key value:(NSString *)value orderType:
(BCPayOrderType)type;

/**
 * 异步查询支付订单或退款订单。内置购买只支持查询支付订单表。
 *
 * @param key 根据key查询。trace_id,out_trade_no,our_refund_no
 * @param value 要查询的值
 * @param ca 支付平台的支付订单或者退款订单
 * @param block 接收查询结果
 */
+ (void)query(xx)OrderByKeyAsync:(BCPayOrderKey)key value:(NSString *)value orderType:(
BCPayOrderType)type block:(BCArrayResultBlock)block;
```

查询订单后返回内容如下：

key	value
subject	商品 名称
body	商品描述
buyer_id	买家ID
buyer_email	买家email
trace_time	订单创建时间
outtradeno	商家自定义订单号
partner	商家ID
refund_status	退款状态
seller_email	卖家email
seller_id	卖家ID
total_fee	支付金额
trace_id	商户平台用户名
trade_state	交易状态
trace_no	交易流水号
refund_finish	退款操作是否完成
refund_fee	退款金额
refund_reason	退款原因
reject_reason	拒绝退款原因
refund_status	退款实时状态

退款状态：

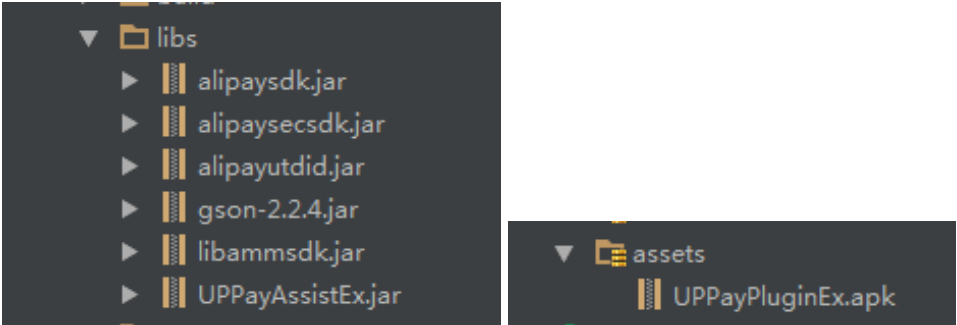
status	说明
REFUND_START = 0	退款开始
REFUND_REJECT = 1	退款被商家拒绝
REFUND_ACCEPT = 2	退款被商家同意
REFUND_SUCCESS = 3	退款成功
REFUND_FAIL = 4	退款被渠道拒绝
REFUND_RETRY = 5	退款被渠道拒绝，但原因不明，需要用原退款单号重试
REFUNDNEEDOFFLINE = 6	用户银行卡已注销，现金回流到商户账户，需要走线下人工操作

Android SDK

下载安装SDK

1. 下载SDK：猛击 [这里](#) 下载最新版的BeeCloud SDK, 标准SDK以jar包方式提供给商户应用工程集成，里面包括微信支付、支付宝支付相关的jar包，支付宝支付相关的jar包为 `alipaysdk.jar`、`alipaysecsdk.jar`、`alipayutdid.jar` 这三个jar包。微信支付为 `libammsdk.jar`，客户无需直接调用这些jar包，这些将统一通过 `BeeCloud.jar` 包进行调用。目前支持微信支付、支付宝支付，银联支付。
2. 安装SDK：
将 `alipaysdk.jar`、`alipaysecsdk.jar`、`alipayutdid.jar`、`libammsdk.jar`、`gson-2.2.4` 增加到libs中，右击将jar包作为库文件引入。银联支付要额外将UPPayPluginEx.apk复制到客户端工程的

assets目录下。如下图：



配置工程

在 `AndroidManifest` 文件中引入如下设置：

```

<uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<activity
    android:name="cn.beecloud.PaymentActivity"
    android:launchMode="singleTop"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity-alias
    android:name=".wxapi.WXPayEntryActivity"
    android:exported="true"        android:targetActivity="cn.beecloud.PaymentAct
ivity">
</activity-alias>

```

全局初始化

首先，在APP启动的时候，初始化BeeCloud需要的 **App Key**，全局只需调用一次。每个应用对应一个 **App Key**，可在控制台中相应App的基本信息 里获得：

```
BeeCloud.setAppIdAndSecret("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx ", this);
```

BeeCloud所有的网络请求默认超时为5秒，如果需要更改（如改为10秒），请调用以下函数，全局只需调用一次：

```
BeeCloud.setNetworkTimeout(10000);
```

微信支付

微信支付功能

添加下列代码发起微信支付：


```

// @"自制白开水"为:商品描述
// totalFee为:支付金额*以分为单位*
// outTradeNo为需要您自行生成的订单号,*32个字符内、包含数字与字母,确保在商户系统中唯一*可根据BCUtil
.generateRandomUUID() stringByReplacingOccurrencesOfString:@"-" withString:@""]获取;
// traceid为:支付用户ID,必须保证在商户系统中的唯一性
// 通过该方法实现对各种参数的初始化,然后发起微信支付,并跳转到微信。
BCPay.getInstance(MainActivity.this).reqWXPaymentAsync("自制白开水", "1", BCUtil.generateRandomUUID().replace("-", ""), "BeeCloud-Android", new BCPayCallback() {
    @Override
    public void done(boolean b, String s) {
        System.out.println("微信支付:" + b + "|" + s);
    }
});

```

在支付成功后会在云端存放购买记录。

微信退款功能

需要申请退款时,可以调用退款接口来实现.在调用之前需要在云端先配置三个参数:

1. operatorID
2. 安全证书 (在商户审核通过后,收到的财付通邮件中有附件以商户号_XXXX.ptx文件即为安全证书)
3. operatorPassword 在客户端用户通过以下代码发起退款请求:

```

// outTradeNo:用户要退订的订单号
// outRefundNo:您自行生成的退订订单号,*32个字符内、包含数字与字母,确保在商户系统中唯一
// refundReason:退款原因
// refundFee:退款金额
// payBlock:发起退款结果回调, 用户在app发起退款请求成功后,商户会收到退款请求,商户可在后台确认,完成退款操作
BCPay.getInstance(MainActivity.this).refundWxPayStartRefundAsync(out_trade_no, "201101120001", "1", "不好喝", new BCPayCallback() {
    @Override
    public void done(boolean b, String s) {
        System.out.println("微信退款:" + b + "|" + s);
    }
});

```

在退款成功后,会在云端生成退款记录。

支付宝支付

支付宝支付功能

当用户确定选择用支付宝进行支付时,可以调用以下接口进行集成支付宝支付。

```

// @param trace_id      支付用户ID, 必须保证在商户系统中唯一. 可通过trace_id查询订单详情。
// @param out_trade_no  商户系统内部的支付订单号, 包含数字与字母, 确保在商户系统中唯一
// @param subject       商品的标题/交易标题/订单标题/订单关键字等。该参数最长为 128 个汉字
// @param body          对一笔交易的具体描述信息。如果是多种商品, 请将商品描述字符串累加传给body
// @param total_fee     该笔订单的资金总额, 单位为RMB-Yuan。取值范围为[0.01,100000000.00], 精确到小数点后两位
// @param scheme        调用支付的app注册在info.plist中的scheme
BCPay.getInstance(MainActivity.this).reqAliPaymentAsync("test", BCUtil.generateRandomUUID().replace("-", ""), "自制白开水", "BeeCloud 自制白开水", "0.01", new BCPayCallback() {
    @Override
    public void done(boolean b, String s) {
        System.out.println("支付宝支付:" + b + "|" + s);
    }
});

```

支付宝退款功能

当需要进行支付宝退款时, 调用以下代码:

```

// 根据out_trade_no, refund_no, refund_reason, refund_fee查询订单是否可退款, 允许退款情况下自动生成预退款订单, 否则返回不可退款原因。预退款订单生成成功后, 在BeeCloud商户后台对预退款订单进行处理。(订单状态trade_status=@”TRADE_SUCCESS”)时支持退款, 其他状态下不支持退款。
// @param out_trade_no  商户系统内部的支付订单号, 包含数字与字母, 确保在商户系统中唯一
// @param refund_no     格式为: 退款日期(8位)+流水号(3~8位)。不可重复, 且退款日期必须是当天日期(年月日)。
// 流水号可以接受数字或英文字符, 建议使用数字, 但不可接受“000”。例如: 201101120001
// @param refund_fee    退款金额
// @param refund_reason 退款原因
// @param block         退款结果回调
BCPay.getInstance(MainActivity.this).reqAliRefundAsync(out_trade_no, "201101120001", "0.01", "不好喝", new BCPayCallback() {
    @Override
    public void done(boolean b, String s) {
        System.out.println("支付宝退款:" + b + "|" + s);
    }
});

```

银联支付

银联支付功能

当用户确定选择用银联进行支付时, 可以调用以下接口进行集成银联支付。

```

/**
 * 银联在线支付
 *
 * @param trace_id      支付用户ID, 必须保证在商户系统中唯一. 可通过trace_id查询订单详情。
 * @param body          商品的标题/交易标题/订单标题/订单关键字等。该参数最长为128个汉字
 * @param out_trade_no  商户系统内部的支付订单号, 包含数字与字母, 确保在商户系统中唯一
 * @param total_fee     支付金额, 以分为单位
 * @return
 */BCPay.getInstance(MainActivity.this).reqUnionPaymentAsync("Android-UPPay", "自制白开水",
    BCUtil.generateRandomUUID().replace("-", ""), "1", new BCPayCallback() {
        @Override
        public void done(boolean b, String s) {
            System.out.println("btnUPPay:" + b + "|" + s);
            int ret = Integer.valueOf(s);
            if (ret == PLUGIN_NEED_UPGRADE || ret == PLUGIN_NOT_INSTALLED) {
                // 需要重新安装控件
                Message msg = mHandler.obtainMessage();
                msg.what = 3;
                mHandler.sendMessage(msg);
            }
        }
    });

```

银联退款功能

当需要进行银联退款时, 调用以下代码:

```

/**
 * 银联预退款, 支持部分退款或者全额退款。如果提供的支付订单的交易状态不支持退款,
 * 在block中返回具体的信息; 如果支持退款, 生成预退款订单, 商户在管理后台管理预退款订单。
 *
 * @param out_trade_no  商户系统内部的支付订单号, 包含数字与字母, 确保在商户系统中唯一
 * @param refund_fee    退款金额
 * @param out_refund_no  商户系统内部的退款订单号, 包含数字与字母, 确保在商户系统中唯一
 * @param refund_reason 退款原因
 */
BCPay.getInstance(MainActivity.this).reqUnionRefundAsync(out_trade_no, "1", "201101120001", "不好喝",
    new BCPayCallback() {
        @Override
        public void done(boolean b, String s) {
            Message msg = mHandler.obtainMessage();
            msg.what = 2;
            msg.obj = s;
            mHandler.sendMessage(msg);
        }
    });

```

查询订单

当用户需要进行订单查询时，可根据 `traceID` 或者 `out_trade_no` 或者 `refund_no` 分别来进行查询。

注：该接口可实现微信、支付宝两种渠道的订单查询，同时可支持查询支付订单和查询退款订单两项操作查询方式。

```
/**
 * 异步查询支付订单或退款订单。内置购买只支持查询支付订单表。
 *
 * @param key 根据key查询。trace_id,out_trade_no,our_refund_no
 * @param value 要查询的值
 * @param ca 支付平台的支付订单或者退款订单
 * @param block 接收查询结果
 */
BCPay pay = BCPay.sharedInstance(MainActivity.this);
pay.queryOrderByKeyAsync(BCPay.BCPayOrderKey.OrderKeyTraceID, "BeeCloud-Android", BCPay.BCPayOrderType.BCPayWxPay, new BCPayResultCallback() {
    @Override
    public void done(List<Map<String, Object>> lstMaps) {
        System.out.println("微信订单数: " + lstMaps.size());
    }
});
```

查询订单后返回内容如下：

key	value
subject	商品 名称
body	商品描述
buyer_id	买家ID
buyer_email	买家email
trace_time	订单创建时间
outtradeno	商家自定义订单号
partner	商家ID
refund_status	退款状态
seller_email	卖家email
seller_id	卖家ID
total_fee	支付金额
trace_id	商户平台用户名
trade_state	交易状态
trace_no	交易流水号
refund_finish	退款操作是否完成
refund_fee	退款金额
refund_reason	退款原因
reject_reason	拒绝退款原因
refund_status	退款实时状态

退款状态：

status	说明
REFUND_START = 0	退款开始
REFUND_REJECT = 1	退款被商家拒绝
REFUND_ACCEPT = 2	退款被商家同意
REFUND_SUCCESS = 3	退款成功
REFUND_FAIL = 4	退款被渠道拒绝
REFUND_RETRY = 5	退款被渠道拒绝，但原因不明，需要用原退款单号重试
REFUNDNEEDOFFLINE = 6	用户银行卡已注销，现金回流到商户账户，需要走线下人工操作

Happy Coding!