

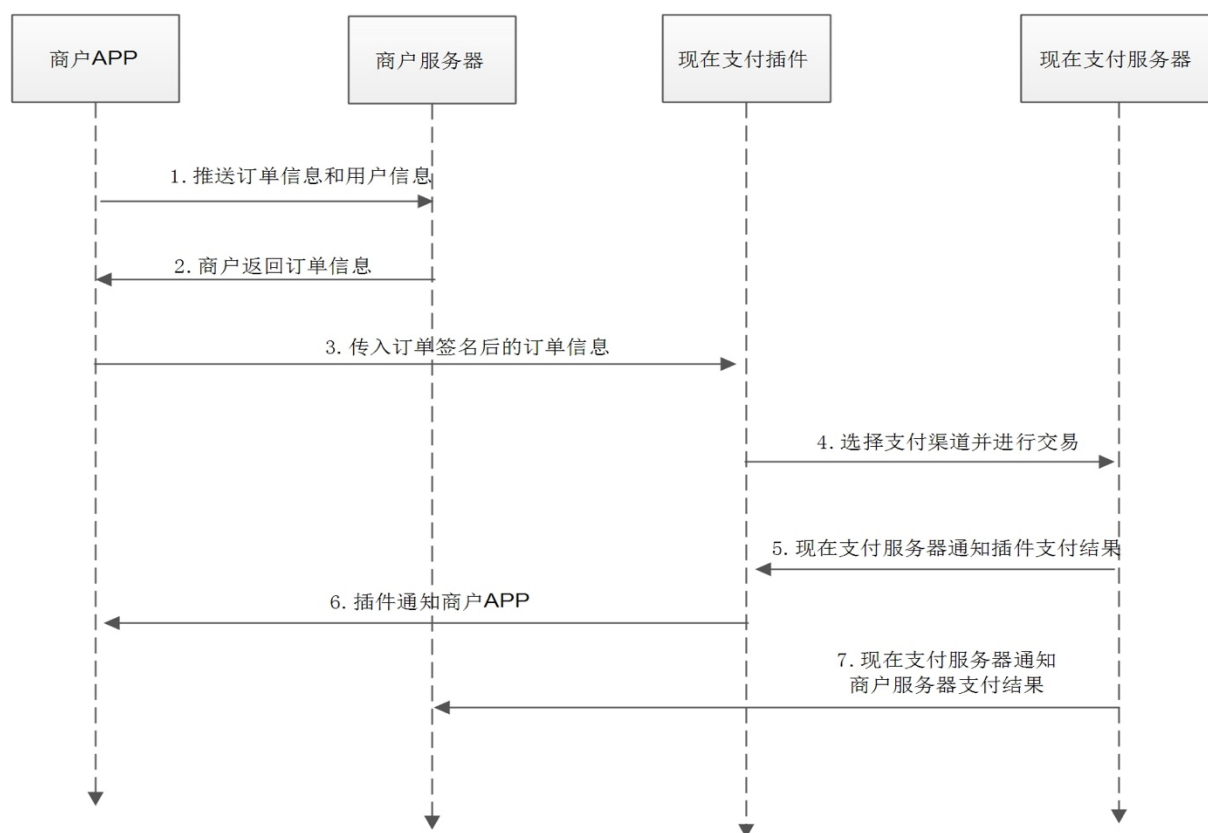
现在支付
聚合支付SDK
接入指南
V 3.0.0

[illegible]

一、概述

现在支付控件包括银联支付和支付宝支付，主要为开发者的手机客户端提供安全、便捷的支付服务，目前支付控件支持Android和iOS两个平台，用户通过输入银行卡号或支付宝账号等有效信息完成支付。

1.1支付流程介绍:



步骤说明:

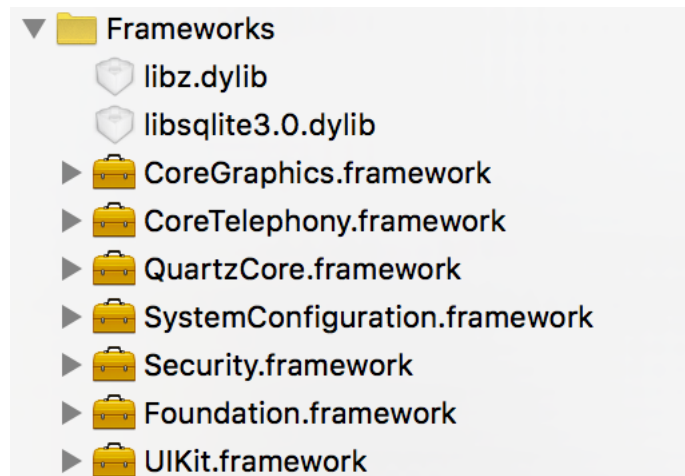
1. 商户APP向商户服务器发送订单信息以及账户信息。
2. 商户APP将符合插件规范的信息传入唤起方法，唤起现在支付支付插件。
3. 用户在插件中调起的支付渠道完成支付操作。
4. 支付完成后，现在支付支付插件接收服务器发送的支付结果通知。
5. 现在支付支付插件通知商户APP支付情况。**(交易状态以商户后台收到的支付结果通知为准)**
6. 支付成功后，现在支付服务器通知商户服务器交易信息。

注意:为了安全考虑，推荐商户服务器收到信息后，根据插件唤起接口规范说明（见附录A）组合信息，并对指定字段进行MD5签名。

二、iOS客户端接入流程

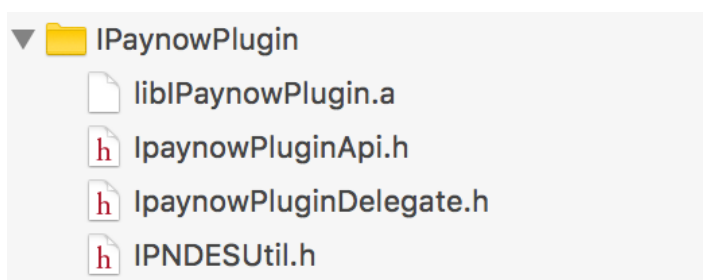
1. 文件引用

添加所依赖的系统库文件（必选）：



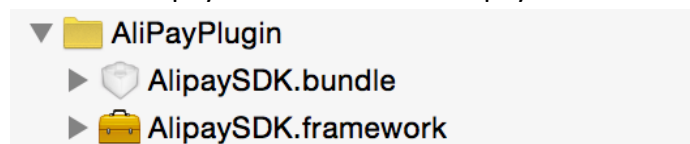
添加基础插件包（必选）：

其中包括libIPayNowPlugin.a、IpaynowPluginApi.h、IpaynowPluginDelegate.h及IPNDESUtil.h。



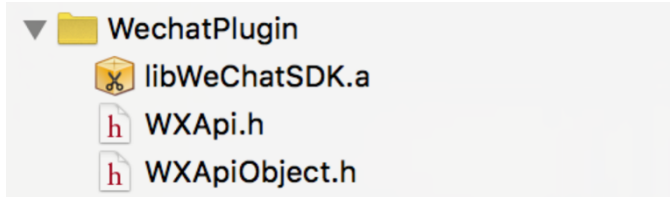
添加支付宝插件包（使用支付宝时引用，可选）：

其中包括AlipaySDK.framework、AlipaySDK.bundle、CoreMotion.framework

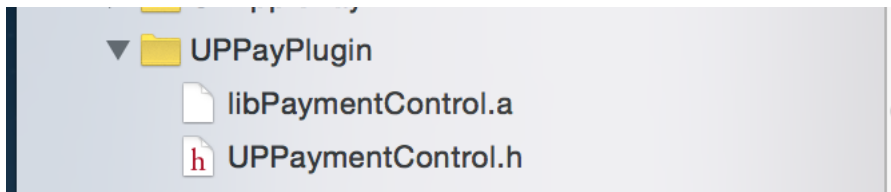


添加微信插件包（使用微信SDK支付时引用，可选）：

其中包括libWeChatSDK.a、WXApi.h和WXApiObject.h。

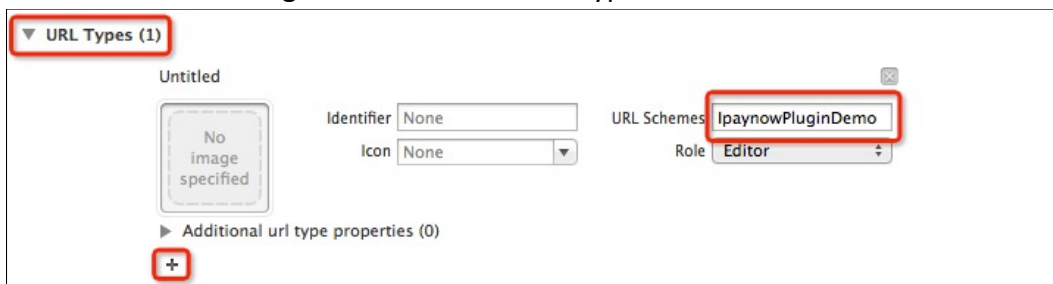


添加银联插件包（使用银联支付时引用，可选）：
其中包括libPaymentControl.a、UPPayPlugin.h文件。



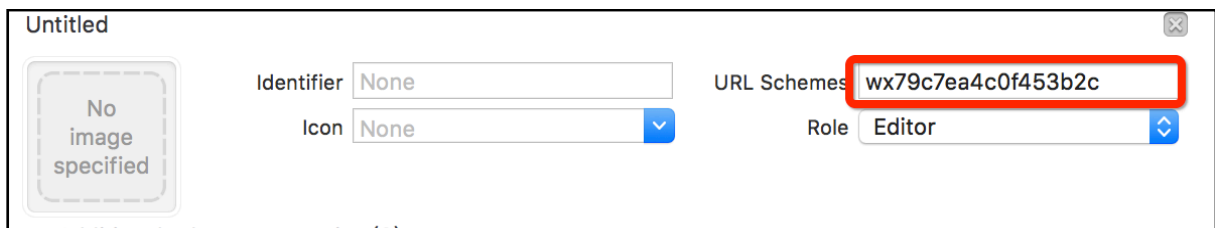
2. 工程设置

在工程的Build Settings中找info，设置URL Types，添加自定义URL Scheme。

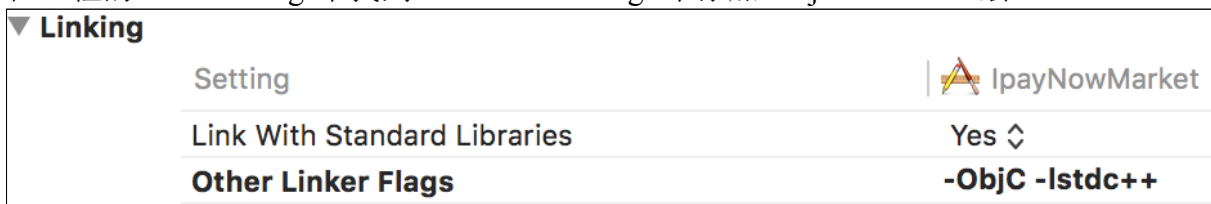


URL Scheme在回调结果使用，建议起名稍复杂一些，尽量避免同其他程序冲突。

若使用微信SDK模式，则需要填写在微信平台上的APPID。



在工程的Build Settings中找到Other Linker Flags 中添加-ObjC、-lstdc++宏。



3. 调用支付接口

第一步：请求服务器生成订单待签名字符串。

客户端将订单信息传递到APP服务器，得到服务器根据订单信息生成的签名字符串。

注意：Demo中为了方便演示该过程是在本地进行。

参考插件中的 IPNPreSignMessageUtil 类中generatePreSignMessage()方法进行待签名串的生成。

参考Demo格式

```
NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init];
[dateFormatter setDateFormat:@"yyyyMMddHHmmss"];

IPNPreSignMessageUtil *preSign=[[IPNPreSignMessageUtil alloc]init];
preSign.appId=@"1408709961320306";
preSign.consumerId=@"IPN_001";
preSign.consumerName=@"1号消费者";
preSign.mhtOrderNo=[dateFormatter stringFromDate:[NSDate date]];

preSign.mhtOrderName=@"IOS插件测试用例";
preSign.mhtOrderType=@"01";
preSign.mhtCurrencyType=@"156";
preSign.mhtOrderAmt=@"10";
preSign.mhtOrderDetail=@"关于订单验证接口的测试";
preSign.mhtOrderStartTime=[dateFormatter stringFromDate:[NSDate date]];
preSign.version=@"1.0.1";
preSign.mhtCharset=@"UTF-8";
preSign.mhtOrderTimeOut=@"3600";

NSString *originStr=[preSign generatePresignMessage];
```

注：若要传入mhtReserved字段且字段中含有"*"、“&”、“=”，等特殊符号，在获取到签名之后请单独对该字段进行一次url编码，再调用支付，否则可能导致验签失败。

补充说明：通过payChannelType字段可指定跳转到某支付渠道。微信渠道有两种模式SDK模式和H5模式，商户若是传了mhtSubAppId就使用SDK模式，否则使用H5模式。SDK渠道需要引用微信官方的SDK，H5渠道则不需要。

第二步：使用签名字符串调起插件支付方法。

```
[IpaynowPluginApi pay:payData AndScheme:@"TestPlugin" viewController:self  
delegate:self];
```

参数说明：

NSString *data
 主要包含商户的订单信息，key=“value”形式，以&连接。

NSString *scheme
 商户程序注册的URL protocol，供支付完成后回调商户程序使用。

UIViewController*viewController
 商户应用程序调试手机支付的当前UIViewController。

id<IpaynowPluginDelegate>delegate
 实现IpaynowPluginDelegate方法的UIViewController。

第三步：实现插件通知接口

接收通知接口应用内结果通知接口为IpaynowPluginDelegate，包含如下方法：

```
- (void)iPaynowPluginResult:(IPNPayResult)result errorCode:(NSString  
    *)errorCode errorInfo:(NSString *)errorInfo;  
IPNPayResult result: 支付结果枚举  
NSString *errorCode : 支付失败或未知时返回错误码  
  
NSString *errorInfo : 支付不成功返回错误信息
```

应用间结果通知接口如下：

iOS9之前使用：

```
+ (BOOL)application:(UIApplication*)application openURL:(NSURL *)url  
    sourceApplication:(NSString *)sourceApplication annotation:(id)annotation;
```

iOS9及以后使用：

```
+ (BOOL)application:(UIApplication*)application openURL:(NSURL *)url options:  
(NSDictionary<NSString *,id> *)options
```

在AppDelegate.h中添加相应方法实现，通过在AppDelegate.m中实现来完成独立返回url异步通知。

注①：选择微信支付后会跳转到微信客户端进行支付，支付完成需要手动切回App，故请开发者慎重选择开启微信支付。

微信支付需要在AppDelegate.m中添加[IpaynowPluginApi willEnterForeground]方法。

附录A

在下文中出现的具体报文格式描述中，“必填”列包含的值的含义如下表所示

符号	请求约束	服务方约束
M	必须包含该域	必须校验该域是否存在和内容的合法性
C	如果条件符合必须包含该域	当条件满足时，必须校验该域是否存在 当该域存在时，必须检查其内容的合法性
O	该域可选	当该域存在时，必须检查其内容的合法性
#		必须与先前报文中对应域的值相同的域

字段名称	字段Key	格式	必填	备注
功能码	funcode	String(5)	M	定值：WP001
版本号	version	String(5)	M	例：1.0.0
商户应用唯一标识	appld	String(1,40)	M	
商户订单号	mhtOrderNo	String(1,40)	M	
商户子商户号	mhtSubMchld	String(64)	O	现在支付子商户号，机构商模式使用
商户商品名称	mhtOrderName	String(1,40)	M	
商户交易类型	mhtOrderType	String(1,40)	M	01普通消费 05代理消费
商户订单币种类型	mhtCurrencyType	String(3)	M	156人民币
商户订单交易金额	mhtOrderAmt	Number(22)	M	单位(人民币)：分 整数，无小数点

商户订单详情	mhtOrderDetail	String(0,200)	O	
商户订单超时时间	mhtOrderTimeOut	Number(4)	O	60~3600秒，默认3600
商户订单开始时间	mhtOrderStartTime	String(14)	M	yyyyMMddHHmmss
商户后台通知URL	notifyUrl	String(1,200)	M	HTTP协议或者HTTPS协议，POST方式提交报文
商户字符编码	mhtCharset	String(1,16)	O	默认值UTF-8
设备类型	deviceType	String(4)	M	定值：01
用户所选渠道类型	payChannelType	String(2)	M	12支付宝、13微信、20银联
商户保留域	mhtReserved	String(100)	O	商户可以对交易进行标记，现在支付将原样返回必须，必须严格遵循以下格式： mchBankId=123&商户自定义参数
商户的appId	mhtSubAppId	String(40)	C	商户在支付宝或微信的appId
是否支持信用卡支付	mhtLimitPay	String(2)	O	0：禁用信用卡 1：允许使用信用卡
消费者ID	consumerId	String(40)	O	消费者在商户系统的ID，非必填，但是推荐填写，以便于辅助数据分析
消费者名称	consumerName	String(40)	O	消费者在商户系统的名称，非必填，但是推荐填写，以便于辅助数据分析
商品标记	mhtGoodsTag	String(0,32)	O	用于营销活动
商户签名方法	mhtSignType	定值	M	定值MD5
商户数据签名	mhtSignature	String(1,64)	M	透传商户的签名，除mhtSignature字段外，所有非空参数均参与MD5签名。

附录B

第一步：对参与MD5签名的字段按字典升序排序后，分别取值后并排除值为空的字段键值对，最后组成key1=value1&key2=value2....keyn=valuen "表单字符串"。

第二步：对MD5密钥进行加密得到"密钥MD5值"。

第三步：最后对 第一步中得到的表单字符串&第二步得到的密钥MD5值 做MD5签名

PS：MD5密钥是用户在注册应用的时候生成的，每个应用一个MD5密钥。

样例：

```
appId=8888888888888888&mhtCharset=UTF-8&mhtOrderNo=20140821161747&mhtOrderName=%E9%99%B6%E6%A0%91%E5%BC%BA&mhtOrderType=01&mhtCurrencyType=156&mhtOrderAmt=1&mhtOrderDetail=%E5%85%B3%E4%BA%8E%E8%AE%A2%E5%8D%95%E9%AA%8C%E8%AF%81%E6%8E%A5%E5%8F%A3%E7%9A%84%E6%B5%8B%E8%AF%95&mhtOrderStartTime=20140821161747&notifyUrl=http%3A%2F%2Flocalhost%3A10802%2F&mhtSignature=72e3b9fea03b81b88224fe0eab1459d9&mhtSignType=MD5
```

使用简便流程注意：

如用安卓文档中推荐的简便接入流程的话，服务器只要根据商户APP上送的待签名字符串根据以下公式生成签名值即可：

签名公式：MD5（待签名串+"&"+MD5（现在支付提供的密钥））；

生成签名值后，根据以下公式拼接好发送给商户APP：

拼接公式：mhtSignature=（签名公式得到的值）+"&"+ mhtSignType=MD5；

样例：mhtSignature=1519adb35c04e0b962d8ca68476d9d56&mhtSignType=MD5

商户使用简便流程时，商户后台需要先做UTF-8的url解码后再做签名处理