



安全支付服务 **Qt for Symbian** 应用开发指南

文件版本：1.0.0.0

支付宝（中国）网络技术有限公司

版权所有

2011-06-10

第 1 章 开发环境准备

1.1 支付服务介绍

安全支付服务是采用 Symbian ECom 框架实现的第三方应用程序插件，主要用来向其它的应用程序提供便捷、安全以及可靠的支付服务。本文主要描述安全支付服务应用开发接口的使用方法，供合作伙伴以及其它第三方应用开发者参考。

1.2 安装

从 Symbian S60 v5 & Symbian ^3 或 Symbian_S60_v3（与 QT 程序运行的平台对应）文件夹中提取 alipay_plugin SISX 安装包，直接安装到手机即可或者以内嵌 SIS 的方式打包到第三方应用安装包中一起安装到手机上。操作完成之后，在系统中检查“程序管理”，如果在已安装应用程序列表中，能找到“安全支付服务”，说明支付服务已经安装成功。

注意：symbian^3 系统集成了 QT 支持，如果是在 S60_V3(FP1 或 FP2)平台开发 QT 需要安装 QT 支持包。

1.3 搭建开发环境

➤ 新建 Qt for Symbian 应用项目

在 Qt Creator 集成环境中，选择新建一个 Mobile Qt Application，名称取为 AlipayTest.

➤ 添加 Interface 源文件

从文件目录 Qt for Symbian\AlipayTest 中提取 AlipayPluginObserver.h, AliPay.h, AliPay.inl, AlipayWrapperPrivate.h, AlipayService.h, AlipayWrapperPrivate.cpp, AlipayService.cpp 拷贝到工程目录下

➤ 调用支付服务接口

实例化 QAlipayService 对象、连接信号与槽并调用支付接口代码如下：

```
QAlipayService* iAlipayService = new QAlipayService();
connect(iAlipayService, SIGNAL(PayFinished(const
QByteArray&)), this, SLOT(payResultNotice ( const QByteArray&)));
iAlipayService->AliXPay(array,0);
```

备注: payResultNotice 为回调支付结果的槽。(可参照 AlipayTest)

➤ 绑定安全支付服务

签约商户需要进行以下一系列的绑定工作。(可参照 AlipayTest)

新建一个配置文件 (例如: PartnerConfig.h) :

1. 添加三个由支付宝分配给商户的参数项, 分别是: 合作商户 ID(PartnerID) 、账户 ID(SellerID)、 安全校验码 MD5(MD5_KEY)。商户可以用签约支付宝账号登录 <https://ms.alipay.com> 获取。

```
_LIT8(PartnerID, "#####");  
_LIT8(SellerID, "#####");  
_LIT8(MD5_KEY, "#####");
```

修改.h 文件 (例如: MainWindow.h) :

1. 在头文件中添加

```
#include "AlipayService.h"
```

2. 在头文件类中添加响应槽:

```
void payResultNotice ( const QByteArray& resultInfo );
```

3. 在头文件中添加成员函数和变量

```
QAlipayService* iAlipayService;
```

修改.pro 文件 (例如: AlipayTest.pro):

1. 添加下面的库

```
LIBS += -lecom.lib -lhash.lib
```

2. 添加>=下面的能力

```
TARGET.CAPABILITY += Location NetworkServices ReadDeviceData  
ReadUserData WriteDeviceData WriteUserData
```

3. 适当增大堆栈大小, 例如

```
TARGET.EPOCSTACKSIZE = 0x14000  
TARGET.EPOCHEAPSIZE = 0x020000 0x800000
```

修改.cpp 文件 (例如: mainwindow.cpp) :

1. 在实现的 CPP 中添加头文件

```
#include <hash.h>
```

```
#include "PartnerConfig.h"
```

2. 实例化 QAlipayService 对象、连接信号与槽 (例如: AlipayTest 中 MainWindow 构造函数)

```
iAlipayService = new QAlipayService();  
connect(iAlipayService, SIGNAL(PayFinished(const
```

```
QByteArray&)), this, SLOT(payResultNotice ( const QByteArray&)));
```

备注: 如果不在构造函数实例化 QAlipayService 对象, 建议 iAlipayService 初始化为 NULL;

3. 在适当的地方调用支付接口

```
iAlipayService->AliXPay(array, 0);
```

4. 在实现的文件添加响应槽函数定义获取返回结果 (例如: AlipayTest 中的 payResultNotice)

5. 在析构函数中添加释放接口代码:

```
if(iAlipayService)  
{
```

```

delete iAlipayService;
iAlipayService = NULL;
}

```

第 2 章 支付服务接口

2.1 支付服务接口列表

目前，S60_QT 平台上的安全支付服务接口如下表所示：

表2-1 支付服务接口表

接口名称	接口描述
QAlipayService	提供外部商户订单的支付功能。后续可能还会增加其它相关功能。

2.2 AlixPay 方法描述

表2-2 AlixPay()方法信息描述

方法原型	void AliXPay(const QByteArray& aOrderInfo, TUint32 alaplId);
方法功能	提供外部商户订单的支付。
参数	<p>const QByteArray& aOrderInfo 主要包含外部商户的订单信息，key="value"形式，以&连接。示例如下： partner="2088002007260245"&seller="2088002007260245"&out_trade_no="500000000006548"&subject="商品名称"&body="这是商品描述"&total_fee="30"&notify_url="http://notify.java.jpxx.org/index.jsp"&sign="kU2Fa3x6V985g8ayTozI1eJ5fHtm8%2FJGeJQf9in%2BcVmRjJHaExbirnGGKJ%2F7B63drqc4Kjlk%2FSg6vtSIk0tdvVBrRpYaKxXVqkJTzRYgUwrrpMudbIj9aMS203dHG0GPyl4Zb6jKDYXHabGG0aBJY3QA7JuTJ23t6SqV%2B5f1xg%3D"&sign_type="RSA" 参考 表 2-4 订单信息描述表 查看各个字段的含义。</p> <p>TUint32 alaplId 接入点 Id，如果用户在外程序客户端已经选择了接入点，传入接入点 id，避免再次弹出接入点选择对话框，默认为 0</p>
备注	该方法是安全支付插件提供的核心方法，是线程安全的。多个线程对此方法的调用会按顺序排队进行，前一个调用返回后，下一个调用才会开始。

2.3 信号 PayFinished 描述

表2-3 信号PayFinished描述

信号原型	void PayFinished(const QByteArray& resultInfo);
方法功能	支付完成（成功或者失败）时发出该信号。
参数	resultInfo, 返回支付结果。参考表 2-5 支付结果信息描述表查看支付结果的详细信息。
备注	注意该信号与响应槽的连接。

表 2-4 订单信息描述表

字段名称	描述	属性	备注
partner	合作商户 ID。 应用开发商与支付宝签约接入支付服务时, 由支付宝分配。	字符串	非空、16 位数字。 商户可以用签约支付宝账号登录 https://ms.alipay.com 获取。
seller	账户 ID。订单付款成功后, 钱会打到本账号中。	字符串	非空、16 位数字。 商户可以用签约支付宝账号登录 https://ms.alipay.com 获取。
out_trade_no	商家自己产生的订单编号, 由商家统一定义, 但此号不能有重复。	字符串	非空、64 位字母、数字和下划线组成
subject	商品名称。 由商家统一定义, 可重复。	字符串	非空、64 位字符 (含中文字符)
body	商品的具体描述信息。	字符串	非空、1024 位字符 (含中文字符)
total_fee	本次支付的总费用。所有商品的费用总和, 以人民币元为单位。如 1.50。	字符串	非空、大于 0 的数字 (精度不超过两位的小数, 如 1.00)
notify_url	商家提供的 url。 订单支付结束时, 支付宝服务端会回调这个 url, 通知商家本次支付的结果。	字符串	非空、255 位, 需要符合 url 编码规则。

sign	上述订单信息的签名。对整个订单按支付宝约定的方式的签名，签名包内容如下： partner=""&seller=""&out_trade_no=""&subject=""&body=""&total_fee="30"¬ify_url=""	字符串	非空。 商家自己生成一对公私钥。用这个私钥对订单信息签名，公钥提供给支付宝，供支付宝在验证订单签名时使用。需要符合 url 编码规则，并且字符编码为 UTF-8。
sign_type	签名类型。由商户选择，以下算法目前可用（MD5/RSA）。	字符串	非空

表2-5 支付结果信息描述表

字段名称	描述	属性	备注
resultStatus	本次操作的状态返回值。	字符串	用来标识本次调用的结果，具体可能的取值，请查看附录 A 错误代码列表。
memo	本次操作的状态描述。	字符串	
result	本次操作返回的结果数据。	字符串	<p>订单支付结果信息。字符串格式，形式一般如下：</p> <pre>partner=""&seller=""&out_trade_no=""&subject=""&body=""&total_fee="30"&notify_url=""&success="true"&sign_type="RSA"&sign="xxx"</pre> <p>注意，这里的 &success="true"&sign_type="RSA"&sign="xxx" 之前的部分为商户的原始数据。</p> <p>success，用来标识本次支付成功失败信息。</p> <p>sign="xxx"为支付宝对本次支付结果(标红部分)的签名，商户可以使用签约时支付宝提供的公钥进行验证。</p>

特别说明：

调用本方法的应用程序，需要通过resultStatus以及result字段的值来综合判断并确定支付结果。在resultStatus=9000，并且success="true"以及sign="xxx"校验通过的情况下，证明支付成功。其它情况归为失败。较低安全级别的场合，也可以只通过检查resultStatus

以及success="true"来判定支付结果。比如，以下是一个订单支付成功完整信息的示例：

```
resultStatus={9000};memo={交 易 成 功};result={partner="2088002007260245"&seller="2088002007260245"&out_trade_no="600000000006891"&subject="商 品 名 称 "&body=" 这 是 商 品 描 述 "&total_fee="1"&notify_url="http://notify.java.jpxx.org/index.jsp"&success="true"&sign_type="RSA"&sign="00I1APPVQcK5bbSgdeFx9HB3Yu/U2+akTZ3T0/P7v3g7XD7TsQCprb609Nybr8CDI rztdUseQN/TCXuEvCU2cvCt1xX9UUyI6f0xXxQF1Dwx7IE2S7Zo5wOeVWmMBnQCQCV8iDjcNxGHwhtCT09bVVf0wbaOiHXvAYzW1vPhyR+0="}
```

2.4 RSA 验证签名

定义：RSA 是一种非对称的签名算法，即签名密钥（私钥）与验签名密钥（公钥）是不一样的，私钥用于签名，公钥用于验签名。使用这种算法签名在起到防数据篡改功能的同时，还可以起到防抵赖的作用，因为私钥只有签名者知道。

- 用集成文档压缩包中 openssl 工具生成一套 RSA 公私钥。登录 无线商户签约平台，把公钥用 txt 文件上传至[我的产品](#)>>密钥管理>>安全校验码（RSA）>>商户公钥。并获取 安全校验码（RSA）>>支付宝公钥

RSA 密钥生成命令：

生成 RSA 私钥

```
openssl genrsa -out rsa_private_key.pem 1024
```

生成 RSA 公钥

```
openssl rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem
```

将 RSA 私钥转换成 PKCS8 格式

```
openssl pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM -nocrypt
```

第 3 章 支付服务集成

3.1 集成方式介绍

目前，我们为第三方应用客户端提供了捆绑安装的方式。即将安全支付服务安装包内嵌入到第三方应用的 sis 包中。在第三方应用客户端安装过程中一并安装安全支付服务。

注：集成安全支付服务的第三方 APP 必须签名，否则会影响支付功能的正常使用。

3.2 捆绑安装

捆绑安装的具体步骤如下所示：

1. 将 alipay_plugin.sisx 拷贝到第三方应用程序客户端 pkg 文件所在目录。
2. 修改 pkg 文件，添加行：
例：@"alipay_plugin2022_022101.sisx",(0x2003AB2E)

其中 alipay_plugin2022_022101.sisx 是 sisx 文件的全路径

0x2003AB2E 是该 sisx 文件的 UID

3. 生成 sis 文件，签名并发布。

附录 A 错误代码列表

以下为安全支付服务所定义的错误代码：

表A-1 系统定义的错误代码表

错误代码	含义
9000	操作成功
4000	系统异常
4001	数据格式不正确
4003	该用户绑定的支付宝账户被冻结或不允许支付
4004	该用户已解除绑定
4005	绑定失败或没有绑定
4006	订单支付失败
4010	重新绑定账户。
6000	支付服务正在进行升级操作。
6001	用户中途取消支付操作。
6002	网络连接异常。

附录 B 安全支付商户示例

为了帮助合作伙伴以及第三方应用开发商基于安全支付服务快速开发自己的收费应用，我们提供了一个在使用安全支付服务的 QT 客户端例子，在压缩包内可以找到，位置在 Qt for Symbian\AlipayTest。该例子程序以简单、清晰的代码演示了使用安全支付服务的步骤，供程序开发者参考。

附录 C 版本修改历史

版本	作者	修改日期	修改内容
1.0.0.0	赵兰东（任飞）	2011-6-10	创建