

IOS 计费 SDK 嵌入文档

更新时间: 2014.8.22 版本号 1.0.4

目录

1. 修订记录.....	3
2. 术语与缩写解释.....	4
3. 使用方法.....	5
3.1 SDK 文件清单	5
3.2 配置 Xcode 工程.....	6
4. 调用支付方法.....	8
5. API 文档.....	10
shareInstance.....	10
initSdk.....	10
pay.....	10
openURL.....	10
附录 I：支付错误代码.....	12
附录 II：Q&A.....	13

1. 修订记录

版本号	作者	更新时间	更新内容
1.0.0	刘旦	2014.7.29	文档初始化
1.0.1	韩玮	2014.7.30	调整函数调用方式，文档校对
1.0.2	韩玮	2014.8.11	增加支付宝和网游能力的相关说明
1.0.3	刘旦	2014.8.14	修改了嵌入过程中的一些注意事项
1.0.4	刘旦	2014.8.22	增加的嵌入过程中的 Q&A

请特别注意文档中红色字体的描述部分 !!!

2. 术语与缩写解释

术语	解释
计费文件	记录游戏计费信息的文件，需要在爱游戏的管理平台下载

3. 使用方法

3.1 SDK 文件清单

在 xCode 的工程目录中新建如图 3-1(A)和图 3-1(B)的文件夹 EgamePaySdk, 并在 xcode 工程中通过 Add Files to“工程名” 导入 sdk 所需文件。

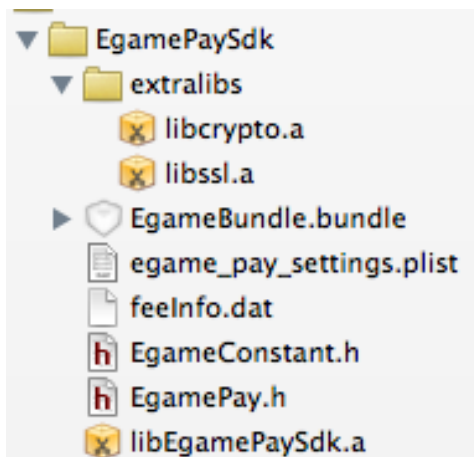


图 3-1(A) 只有短信支付能力的 SDK 文件列表

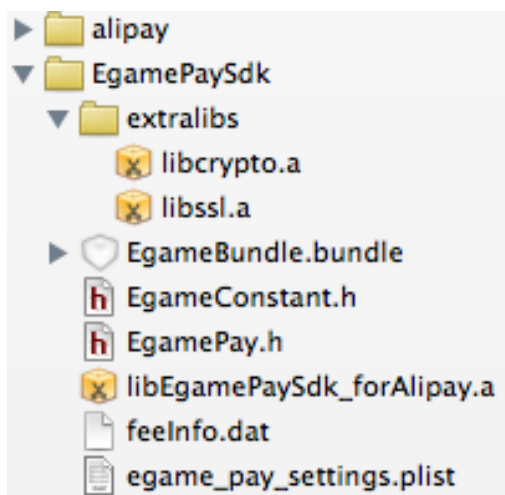


图 3-1(B) 短信和支付宝都有的 SDK 文件列表,

注意 alipay 文件夹和 libEgamePaySdk_forAlipay.a 文件

特别的: egame_pay_settings.plist 和 feeInfo.dat 两个文件必须要放在项目根目录中!

egame_pay_settings.plist 可在 Demo 项目中找到示例文件,feeInfo.dat 为游戏的计费文件.

3.2 配置 Xcode 工程

第一步：加入系统 Libraries

AppSupport.framework

(此框架为私有框架,需要从如下路径添加,不同版本的 SDK 可能路径会有不同:

/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS7.

1.sdk/System/Library/PrivateFrameworks)

CoreTelephony.framework

CFNetwork.framework

SystemConfiguration.framework

Security.framework

第二步：配置 Settings 文件

在 xCode 中打开 egame_pay_settings.plist 文件,编辑 “egame_channel” 字段的 value 值,以修改渠道号. 如图 3-2 所示

Key	Type	Value
▼ Root	Dictionary	(1 item)
egame_channel	String	00012243

图 3-2

第三步：配置 entitlements 文件：

Xcode5 生成.entitlements 文件的操作：选择 target 的 Capabilities 选项，找到 Keychain Sharing，开关设置 ON 状态。如图 3-2 所示：

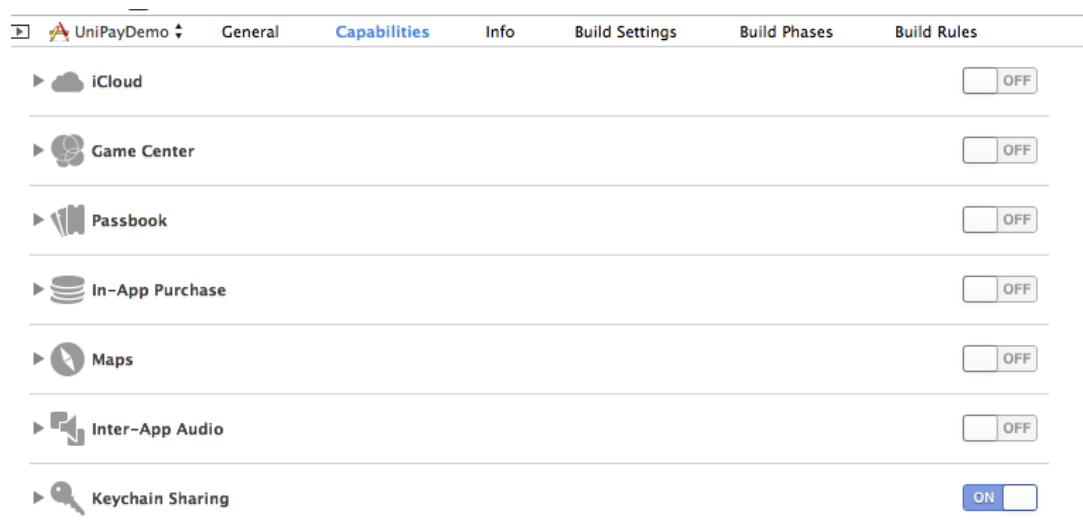


图 3-2

工程中会默认以工程名生成.entitlements 文件，打开.entitlements 文件，添加两项：

```
<key>com.apple.CommCenter.Messages-send</key><true/>
```

```
<key>com.apple.coretelephony.Identity.get</key><true/>"
```

工程target设置

选择工程 target 的 Build Settings，找到 Other Linker Flags，添加-lstdc++，如图 3-3 所示：

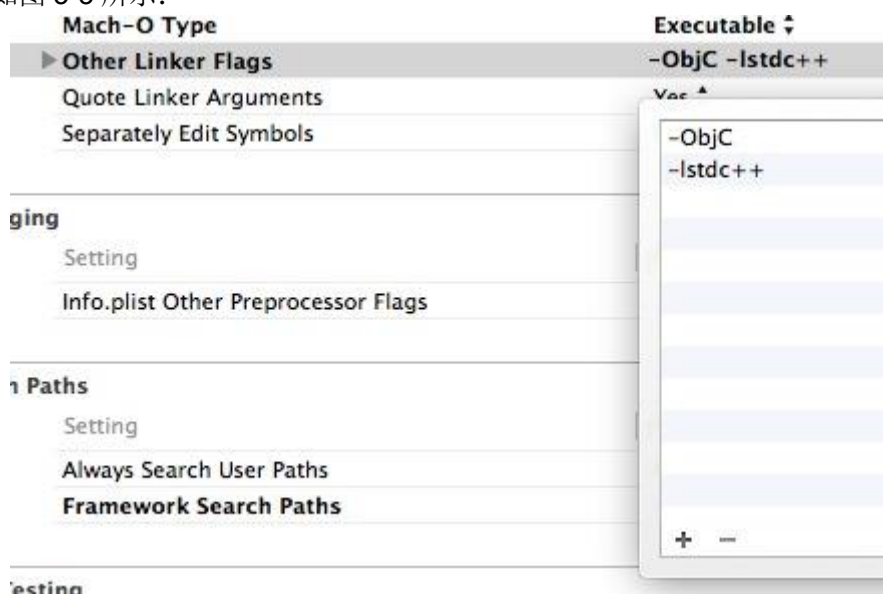


图 3-3

第四步：配置 URL Schemes(可选)：

如果使用的 SDK 带有支付宝支付能力，则需要配置此项。打开 Target 的 Info 选项卡，找到 URL Types 并添加一个新项，Identifier 为 alipay，URL Schemes 为自定义，建议为产品英文名字+egame(不支持下划线等特殊字符)，如图 3-4 所示：



图 3-4

4. 调用支付方法

调用支付代码主要分为以下步骤：

- 1) 导入头文件：在需要调用支付代码的文件中导入以下头文件

```
#import "EgamePay.h"
#import "EgameConstant.h"
```

- 2) 初始化 sdk：在首个界面的 `viewDidLoad` 调用 `sdk` 的初始化函数 `initSdk`

```
-(void)viewDidLoad{
    [super viewDidLoad];

    [[EgamePay sharedInstance] initSdk];
}
```

- 3) 设置游戏参数：在需要调用支付代码的文件中设置游戏参数

通过 `NSDictionary` 来存储游戏的参数。

以下是单机游戏需要传入的参数,道具的 `id` 为必选字段

```
NSDictionary *payParams = [NSDictionary dictionaryWithObjectsAndKeys:
    @"此处为道具的id", PAY_PARAMS_KEY_TOOLS_ALIAS,
    @"此处为道具的描述", PAY_PARAMS_KEY_TOOLS_DESC, nil];
```

以下是网络游戏需要传入的参数,充值金额和订单流水号为必选字段,其中充值金额单位为(元)

```
NSDictionary *payParams = [NSDictionary dictionaryWithObjectsAndKeys:
    @"此处为充值金额", PAY_PARAMS_KEY_TOOLS_PRICE,
    @"此处为订单流水号", PAY_PARAMS_KEY_CP_PARAMS,
    @"此处为道具的描述", PAY_PARAMS_KEY_TOOLS_DESC, nil];
```

调用 SDK 的 `pay` 方法并处理支付结果：在需要购买的道具中点击增加如下调用支付的接口及支付结果处理代码

```
[[EgamePay sharedInstance] pay:payParams
    payCallback:^(EgamePayResult state,
        NSDictionary *payParams,
        int failedReason) {

    if(state == PaySuccess) {
        //此处为支付成功
    } else if (state == PayCanceled) {
        //此处为支付取消
    } else {
        //此处为支付失败
        //failedReason为错误代码
    }
}];
```

SDK 回调结果：根据 SDK 回调的结果判断支付状态，从而决定是否发放道具。

上述支付接口会以 block 方式返回支付状态，支付状态在 EgameConstant.h 中

```
typedef enum{  
    PaySuccess,      // 支付成功  
    PayFailed,       // 支付失败  
    PayCanceled,     // 支付取消  
} EgamePayResult;
```

如果支付失败,则会通过 failedReason 返回错误码. 有关错误码的信息, 请参考附录 I .

4) 在 AppDelegate.m 类中重写如下方法(可选)

如果使用的 SDK 带有支付宝支付能力, 必须要添加如下代码:

```
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url  
sourceApplication:(NSString *)sourceApplication annotation:(id)annotation {  
    [[EgamePay sharedInstance] openURL:url application:application];  
    return YES;  
}
```

5. API 文档

方法摘要

+ (id)	shareInstance 单例模式，返回唯一实例
- (void)	initSdk 在应用启动的时候调用，可以在第一个界面的 ViewDidLoad 中调用，也可以在需要的界面的 ViewDidLoad 中调用，越早越好
- (void)	pay: (NSDictionary *) payParams payCallback: (EgamePayCallbackBlock) callback 调用支付函数，在相应的购买按钮按下后执行
- (void)	openURL: (NSURL *)url application: (UIApplication *)app 处理支付宝回调结果

方法详细信息

shareInstance

+ (id) shareInstance
返回 sdk 的支付对象的单例

initSdk

- (void) initSdk
在应用启动的时候调用，可以在第一个界面的 ViewDidLoad 中调用，也可以在需要的界面的 ViewDidLoad 中调用，越早越好

pay

- (void) pay: (NSDictionary *) payParams
payCallback: (EgamePayCallbackBlock) callback
调用支付函数，在相应的购买按钮按下后执行

参数:

(NSDictionary *) payParams - 计费信息和游戏参数
(EgamePayCallbackBlock) callback - 计费的回调函数

openURL

- (void) openURL: (NSURL *)url application: (UIApplication *)app

处理支付宝回调结果

参数:

(NSURL *)url - AppDelegate 中 application 函数中的 NSURL
(UIApplication *)app - AppDelegate 的实例

附录 I：支付错误代码

```
// 0 没有错误,支付成功或支付取消时收到此值
// -2 SDK正在初始化,暂时无法支付

// -101 计费文件未找到或数据读取异常
// -103 应用校验失败,请确认申报产品信息和当前产品是否一致
// -104 SIM异常,可能的原因:未越狱 未插卡 SIM卡不支持等
// -105 道具别名错误
// -106 道具价格错误

// -200 没有调用初始化方法
// -201 计费回调对象为空
// -202 计费道具别名错误,请确认产品申报时是否已经录入此道具别名
// -203 渠道ID数据异常,请确认产品申报时是否已经录入此渠道
// -204 SERVICE_CODE数据异常
// -205 自定义参数格式异常
// -206 计费方法调用过快,请确认前一次计费已经返回
// -207 计费短信发送失败,请检查手机是否越狱 SIM卡状态是否正常 是否开启了飞行模式 是否欠费等
// -300 获取计费流水号网络异常
// -301 获取计费流水号数据异常
// -400 手机未越狱

#define REASON_CODE_NO_ERROR 0
#define REASON_CODE_INIT_DOING -2
#define REASON_CODE_FEEINFO_IS_NIL -101
#define REASON_CODE_APPKEY_INVALID -103
#define REASON_CODE_IMSI_ERROR -104
#define REASON_CODE_TOOLSALIAS_IS_NIL -105
#define REASON_CODE_TOOLSPRICE_IS_NIL -106

#define REASON_CODE_INIT_FAILED -200
#define REASON_CODE_LISTENER_IS_NIL -201
#define REASON_CODE_TOOLSCODE_INVALID -202
#define REASON_CODE_CHANNELID_INVALID -203
#define REASON_CODE_SERVICE_CODE_INVALID -204
#define REASON_CODE_CPPARAM_INVALID -205
#define REASON_CODE_PAY_CALL_FAST_ERROR -206
#define REASON_CODE_PAY_SMS_SEND_ERROR -207

#define REASON_CODE_GET_SERIAL_NUMBER_NET_ERROR -300
#define REASON_CODE_GET_SERIAL_NUMBER_ERROR -301

#define REASON_CODE_NO_JAILBREAK -400
```

附录 II: Q&A

1) Q: 为什么在模拟器上运行报错?

A: 目前版本 SDK 暂时支持真机调试, 不支持模拟器。

2) Q: 为什么测试机无法使用短代功能?

A: 目前短代仅支持越狱的机型, 未越狱机型无法使用短代功能。

3) Q: 为什么无法使用短代功能, 点击短代界面的支付按钮就提示报错?

A: 是否正确添加 **AppSupport.framework** (此方法为私有方法, 请通过路径 /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS7.1.sdk/System/Library/PrivateFrameworks 不同版本 SDK 路径可能有所不同。)

是否添加 **CoreTelephony.framework**

是否在 **entitlements** 文件中添加如下两个标识:

```
<key>com.apple.CommCenter.Messages-send</key><true/>
<key>com.apple.coretelephony.Identity.get</key><true/>"
```

4) Q: 为什么初始化提示计费文件未找到?

A: 检查是否已在工程中添加计费文件, 并且注意 **feeinfo.dat** 文件的存放位置, 可以参考 **Demo** 中所放位置, 若位置不正确, 可能会导致读取不到文件。若修改位置后, 仍提示找不到计费文件, 请 **clean** 当前工程, 并将测试机器上的应用卸载, 重新安装。

5) Q: 为什么在嵌入 SDK 时, 原本工程会提示在 **libEgamePaySdk.a** 中链接不到部分函数?

A: 针对 **cocos2d-x** 中嵌入, 需要在 **Other Linker Flags** 中增加 "-ObjC" 标识, 会把静态库中所有的 **Objective-C** 类和分类都加载到最后的可执行文件中。

6) Q: 为什么在嵌入 SDK 后, 原工程会提示报错: **Apple Mach -O Linker Error**

A: 可能由于增加了 **Other Linker Flags** 中的参数, 导致部分库文件中所引用的 **framework** 没有引入, 请根据错误的提示信息, 增加相应的 **framework**。

7) Q: 为什么跳转支付宝的快捷支付或者支付宝钱包之后, 回到应用时, 无法获得回调的结果?

A: 检查是否在 **AppDelegate.m** 重写 **application:openURL:sourceApplication:annotationz** 这个方法, 是否在 **URL Types** 中配置 **URL Schemes** 与 **Identifier**, 注意此处的 **Identifier** 要设置为 **alipay** 注意都是小写, **URL Schemes** 中不要用下划线, 中文等特殊符号, 尽量复杂, 不与其他应用冲突, 建议采用应用英文名称+**egame**。