**IOS的APPOTA Game SDK第四版本**

**Appota Game安卓的SDK4已展开并在开发人员预览版状态。第3版本还在[appota-ios-game-sdk3](https://github.com/appota/ios-game-sdk3).**

**重点**

* **Appota Login – 支持Appota, Google, Facebook, Twitter, Guest多种登录方式**
* **Appota Payment – 支持SMS, Card, E-Bank, Apple Payment多种支付方式**
* **Appota Analytics –可收集、可视化并了解用户以及应用数据(session, active, error log ...)的一种服务**
* **为IOS、安卓、window phone各种平台支持*Push Notification（*推送通知）**

**概括**

1. [**集成SDK**](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md#head1-integrate-sdk)
   * **1.1.** [**首先要求**](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md#head2-prerequisites)
   * **1.2.** [**接受 APP api key 和客户端 key**](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md#head2-obtain-app-api-key-and-client-key)
   * **1.3.** [**Xcode Project配置**](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md#head2-configure-your-xcode-project)
   * **1.4. 更新sdk 版本**
2. **客户端的 APIs**
   * **2.1初始化和配置SDK**
   * **2.2用户函数**
   * **2.3支付函数**
   * **2.4分析、统计函数**
   * **2.5push notification相关函数**
   * **2.6游戏里设置角色函数**
   * **2.7 其他函数**
3. **Class相关资料**[**FAQ and Glossary**](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md#head1-faq)**常见问题及词汇表**

**1.****集成SDK**

**示例项目[sample apps](https://github.com/appota/ios-game-sdk/blob/master/Sample)是为您设置的项目。您也可以集成 IOS SDK到现有应用**

**1.1****先决要求**

**开始使用IOS 的SDK, 您可设置SDK并开始新项目或者集成SDK到一个现有应用。您可以试一试来了解SDK的运行机制。Appota game IOS 的SDK支持下面软件:**

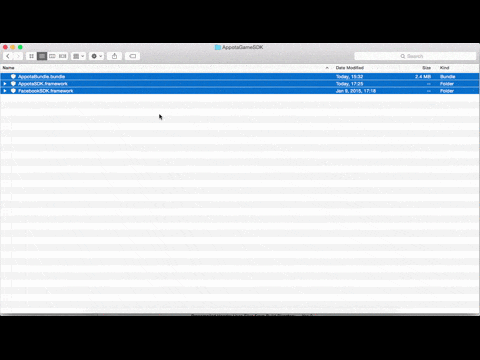
* **Xcode 6 以及更新版本**
* **iOS 6以及更新版本**

**###1.2.****获取app api key 和客户端key**

* **从应用管理页面取得ClientKey, APIKey, ClientSecret等Appota应用. 若您还没向Appota注册Appota应用，您应该[建设新应用]**
* **获取facebook Application ID.建设、获取及设置IOS的Facebook app info 具体查看**
* **获取 Google Client ID 和　Client Secret.建设、获取及设置IOS的Google app 具体查看**
* **获取Twitter Consumer Key 和 Twitter Consumer Secret Key . . 建设、获取及设置IOS的Twitter app 具体查看**

**1.3.****您Xcode Project的配置  
  
1.3.1.添加 Appota iOS SDK 到您的Xcode Project**

**打开[ios-game-sdk/Frameworks](https://github.com/appota/ios-game-sdk/blob/master/Frameworks)并添加[AppotaSDK.framework](https://github.com/appota/ios-game-sdk/blob/master/Frameworks/AppotaSDK.framework),** [**AppotaBundle.bundle**](https://github.com/appota/ios-game-sdk/blob/master/Frameworks/AppotaBundle.bundle)**,** [**FacebookSDK.framework**](https://github.com/appota/ios-game-sdk/blob/master/Frameworks/FacebookSDK.framework)**到您的项目. (注意选择Create groups for any added folders 和选 Copy items into destination group's folder (if needed))**

**[](https://github.com/appota/ios-game-sdk/blob/master/images/add_framework.gif)**

**1.3.2.添加并安装框架依赖**

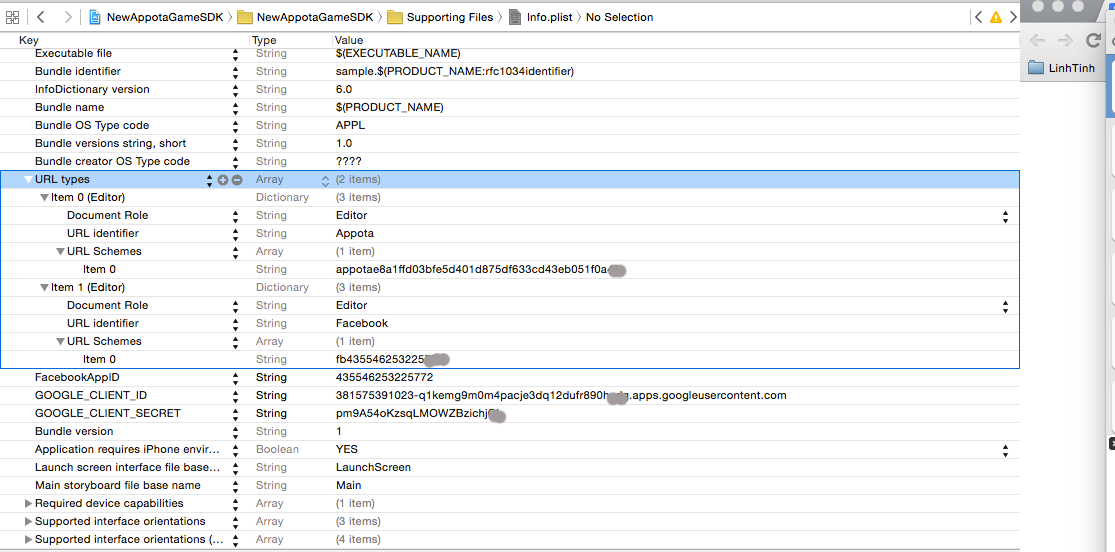
**从Project Navigator选择您的项目→并选择应用程序的目标。打开构建阶段选项卡，在链接的二进制库部分，添加以下框架：**

* **StoreKit.framework**
* **Accounts.framework**
* **Social.framework**
* **Twitter.framework**
* **ImageIO.framework**
* **CoreTelephony.framework**
* **AdSupport.framework**
* **SystemConfiguration.framework**
* **MessageUI.framework**
* **CoreMotion.framework**
* **Security.framework**

**打开 Build Settings, 找 Other Linker Flags, 添加flag-ObjC然后把 Link Frameworks Automatically, Enable Modules (C and Objectivce-C)) 都设为YES.**

**1.3.3. Info .plistInfo .plist文件配置**

* **打开 Info plist输入您的Facebook AppID到FacebookAppID key。**
* **打开 Info plist输入您的[Google Client Id and Sceret](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "head3-google-appid)到GOOGLE\_CLIENT\_ID和GOOGLE\_CLIENT\_SECRET key。**
* **打开 Info plist输入您的[Twitter consumer key and twitter consumer secret](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "head3-twitter-appid)到TWITTER\_CONSUMER\_KEY和TWITTER\_CONSUMER\_SECRET key。**
* **打开 Info plist输入您的[Appota APIKey](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "head3-appota-appid)到AppotaAPIKey key 。**
* **添加URL方案：**
  + **Facebook的标识，URL方案：fbYOUR\_FACEBOOK\_APP\_ID，[Facebook AppID](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "head3-facebook-appid)可以从Facebook应用程序仪表板获得.**

**一个完整的.plist文件就像下面图片或您可在[sample apps](https://github.com/appota/ios-game-sdk/blob/master/Sample)查看[](https://github.com/appota/ios-game-sdk/blob/master/images/plist.png)**

**1.4.****更新SDK版本**

**我们推出SDK新版本之时，您可看下面的描述改变**

* **在Xcode中选择以下框架和捆绑，然后点击键盘上的Delete。然后选择移动到回收站：**
  + **ApppotaFramework.framework**
  + **FacebookFramework.framework**
  + **AppotaBundle.bundle**
* **Include SDK的新版本按照上面的设置过程**

**2.客户端的 APIs**

**2.1初始化和配置****SDK**

**相关class 和函数：**

[**AppotaGameSDK class –初始化函数**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK-class.md#init-function)

**初始化Appota SDK当应用通过[AppotaGameSDK](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK-class.md) class调用[AppotaGameSDK configure]时必要被调用一次。大多数使用下面方式来调用：AppDelegate'sapplication:didFinishLaunchingWithOptions:  
在AppDelegate.h文件添加 import #import <AppotaSDK/AppotaSDK.h>使用AppDelegate像一个 protocol:**

**@interface AppDelegate : UIResponder <UIApplicationDelegate, AppotaGameSDKCallback>**

**在AppDelegate.m:**

**- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions**

**{**

**// Your code is placed above**

**[AppotaGameSDK configure];**

**[AppotaGameSDK sharedInstance].delegate = self;**

**}**

**在function中添加 handle open URL : application:openURL:sourceApplication:annotation:**

**- (BOOL) application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation {**

**return [AppotaGameSDK handleOpenURL:url sourceApplication:sourceApplication annotation:annotation];**

**}**

**有 4种 function 可用来 control SDK follow:**

* **setKeepLoginSession:(BOOL)这个function将控制 Appota Login Session 在应用运行时可保留还是删除 (删除Login Session时用户要重新登录).**

**注意 :若不调用该函数，Login session将默认保留**

* **setAutoShowLoginDialog:(BOOL)****这个function 将控制 Appota Login View 在运行应用之时自动show (用户尚未登录)或您要调用[[AppotaGameSDK showLoginView]](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "show-login-view)才能show Login View.**

**注意：若您不调用这个函数，默认Login View将自动显示**

* **setSDKButtonVisibility:(BOOL)当您想要隐藏或显示SDK浮动按钮时调用这个函数。  
  注意：若您不调用这个函数，默认SDK Button将在运行应用之时自动显示**
* **setHideWelcomeView:(BOOL)当您想要隐藏或显示Welcome View是调用这个函数**
* **注意：若您不调用这个函数，默认Welcome View将在运行应用之时自动显示。**

**2.2.** **相关用户函数**

**相关class 和函数：**

* [**AppotaGameSDK- user function**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK-class.md#user-function)
* [**AppotaUserLoginResult**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaUserLoginResult-class.md)

**当完成SDK初始化时，开启使用登录功能。**

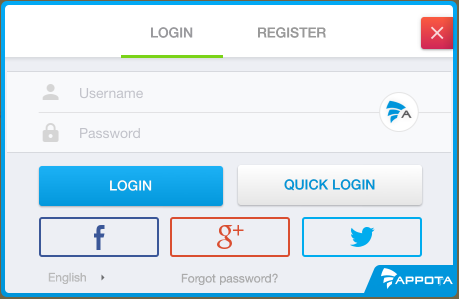
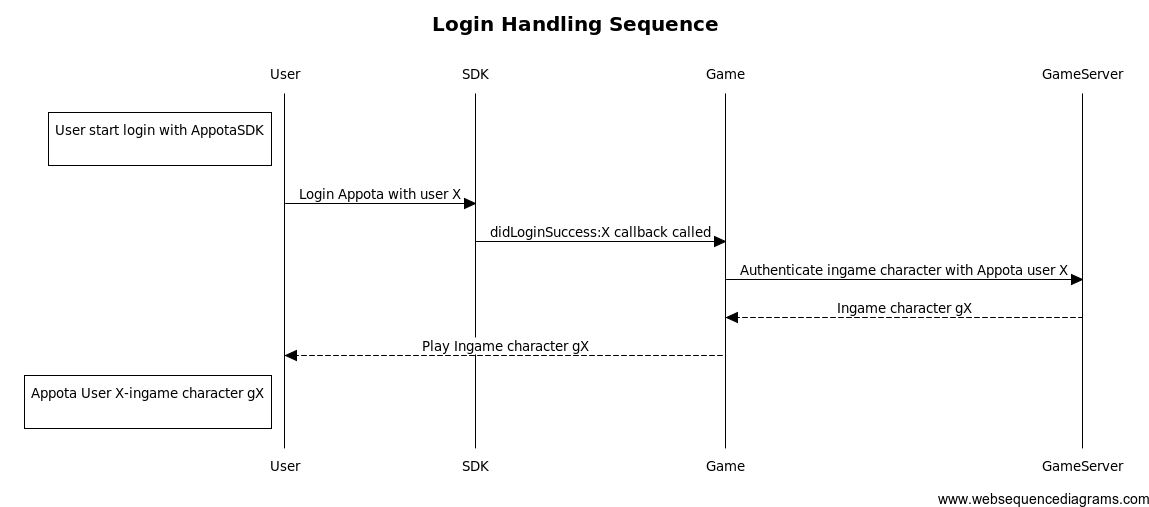
**2.2.1.显示登陆界面**

**以上面所提，Login Dialog可用[[AppotaGameSDK sharedInstance] setAutoShowLoginDialog:](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "set-auto-show-login-dialog)控制。若您不要自动显示登录界面，需要时您可以手动调用showLoginView函数**

* **显示登陆界面**

**[AppotaGameSDK showLoginView];**

**Appota Login Dialog 包括Facebook, Google, Twitter, Appota User和 Quick Login等5种授权模式。**

**[](https://github.com/appota/ios-game-sdk/blob/master/images/login_dialog.png)  
根据这登录机制，请执行[didLoginSuccess](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "did-login-succeed)回调，以确认您已成功登录：[](https://github.com/appota/ios-game-sdk/blob/master/images/ios_login_sequence.png)**

**2.2.2. Show Register view**

**[AppotaGameSDK showRegisterView];**

**用这函数来 show register view.**

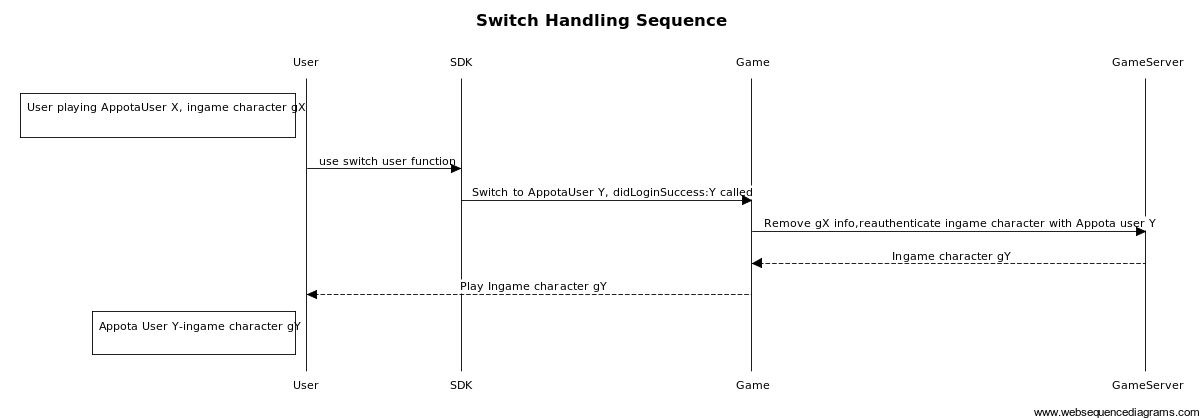
**2.2.3. Logout**

**[AppotaGameSDK logOut];**

**用这函数来log out.**

**2.2.4.切换账号  
用户已登录并允许登录到其他账号之时可用该函数。当您成功切换账号时，didLoginSuccess将回调，因此请退出您的游戏账号并确认已切换的新账号。**

**[AppotaGameSDK switchAccount];**

**[](https://github.com/appota/ios-game-sdk/blob/master/images/ios_switch_user_sequence.png)**

**2.2.5. Show User Info**

**用这函数来显示用户信息  
[AppotaGameSDK showUserInfoView];**

**2.2.6. Check user login**

**该函数将返回用户登录状态。若已登录将返回YES, 还没登录将返回NO**

**[AppotaGameSDK isUserLoggedIn];**

**2.2.7.获取已登录用户信息  
将返回AppotaUserLoginresult（如果还没登录将返回nil）**

**[AppotaGameSDK getUserInfo];**

**2.2.8. 显示交易历史**

**该函数显示玩家曾进行的交易 (当玩家登陆游戏).**

**[AppotaGameSDK showTransactionHistory];**

**2.2.9. Handle authenticaiton callbacks**

**AppotaSDK 提供4个回调以在[AppotaGameSDKCallback](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDKCallback-class.md)设置登录功能,请在Appdelegate.m设置该功能**

**注： 2个回调为@required : @required和@required.**

**登录成功的回调**

* **从AppotaUserLoginResult获取Appota user info，然后上传到您的服务器来确认并且创造游戏玩家。在服务器集成玩家具体查看[Tích hợp user](https://github.com/appota/ios-game-sdk/wiki/Integrate-user-system)**
* **注：进行集成游戏用户之前记得确认Appota UserID, UserName和 Access token。**

**- (void) didLoginSuccess:(AppotaUserLoginResult\*) userLoginResult**

| **参数** | **描述** |
| --- | --- |
| userLoginResult | userLoginResult是含有玩家信息(username, user\_id, access token)的[AppotaUserLoginResult](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaUserLoginResult-class.md) object |

**Login fail callback**

* **当登录发生问题时的回调.**

**- (void) didLoginErrorWithMessage:(NSString \*)message withError:(NSError \*)error;**

| **参数** | **描述** |
| --- | --- |
| message | message: 是通知错误的通告 |
| error | error : 错误 |

**Logout callback**

* **玩家退出时的回调**

**- (void) didLogout:(NSString\*) userName**

| **参数** | **描述** |
| --- | --- |
| userName | userName是退出的用户名称 |

**Close login view callback**

* **玩家关闭应用运行过程中 login view的回调。**

**- (void) didCloseLoginView;**

**2.3. 支付相关的函数**

**相关 Class 和功能：**

* [**AppotaGameSDK - payment function**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK-class.md#payment-function)
* [**AppotaGameSDKCallback**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDKCallback-class.md)
* [**AppotaPaymentResult**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaPaymentResult-class.md)

**使用Appota SDK的支付功能需要了解Appota支付机制和支付配置。烦请在[Appota Payment Document](https://github.com/appota/ios-game-sdk/wiki/Passive-Confirmation-via-IPN)参考支付机制并且在[Appota Developer Portal](https://github.com/appota/ios-game-sdk/blob/master)参考支付配置。**

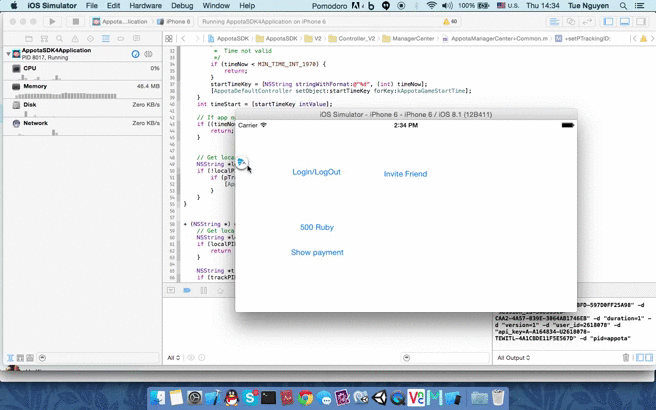
**2.3.1. 显示 payment view**

**显示payment view方式为3个。每个 payment view 表示一个或者一列payment package，每 package含有 game coin、amount 或者游戏里package 的信息。**

**Show default list payment view**

**您可以指定payment packages 列表显示 payment view (列表被配置在[Appota Developer Portal](https://github.com/appota/ios-game-sdk/blob/master)上)**

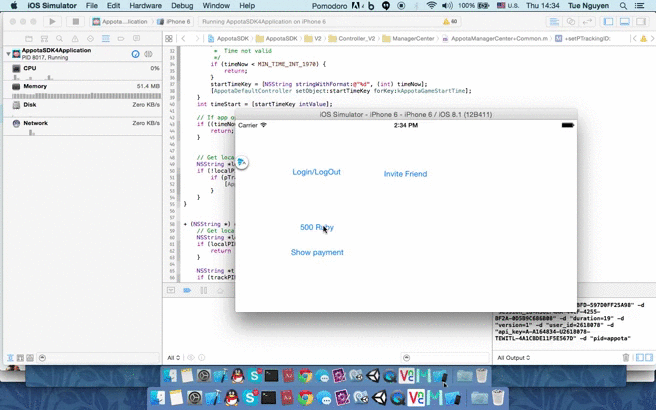
**[AppotaGameSDK showPaymentView]**

**[](https://github.com/appota/ios-game-sdk/blob/master/images/list_item.gif)**

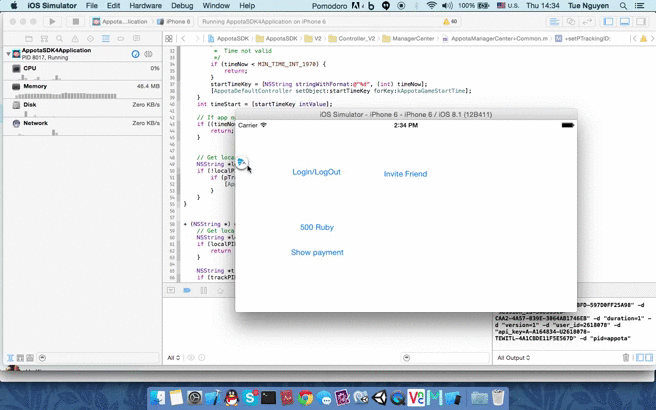
**2.3.2. 以一个默认package 显示 payment view**

**您可以属于您游戏机制的一个指定package显示payment view**

**[AppotaGameSDK showPaymentViewWithPackageID:];**

**[](https://github.com/appota/ios-game-sdk/blob/master/images/one_item.gif)**

**2.3.3. 从SDK按钮显示payment view**

**[](https://github.com/appota/ios-game-sdk/blob/master/images/list_item.gif)**

**2.3.4. 关闭 payment View**

**您可以用以下的函数将 payment view关闭:**

**[AppotaGameSDK closePaymentView];**

**2.3.5 处理 payment** **回调**

**所有成功交易的回调都在AppotaGameSDK delegate (设置在[init function](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "head2-init-sdk))里。回调函数要放在AppDelegate.m里。有一个@required函数被界定在[AppotaGameSDKCallback](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDKCallback-class.md)里**

**支付成功的回调**

**按照支付机制 (APN 还是 IPN,烦请参考[Appota Payment](https://github.com/appota/ios-game-sdk/wiki/Integrate-payment-system),进行验证支付或者向用户通知支付成功。**

**- (void) didPaymentSuccessWithResult:(AppotaPaymentResult\*) paymentResult withPackage:(NSString \*) packageID;**

| **参数** | **描述** |
| --- | --- |
| paymentResult | paymentResult是[AppotaPaymentResult](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaPaymentResult-class.md)，含有支付信息 (transaction id, amount, ...) |
| packageID | packageID是支付package的id, packageID被界定在支付配置的过程中 |

**支付发生错误之时的回调**

**- (void) didPaymentErrorWithMessage:(NSString \*)message withError:(NSError \*)error;**

| **参数** | **描述** |
| --- | --- |
| message | message is message of error |
| error | error the error payment problems |

**Payment state的回调**

**需要提供PAYMENT\_STATE 参数（支付参数）让我们 SDK 反馈相应 package的信息。按照您的 payment package，实施该函数并且反馈一个对的 payment state。查看[Tìm hiểu thêm về Payment state](https://github.com/appota/ios-game-sdk/blob/master/README-vi.md" \l "head3-payment-state).**

**比如： packageID: com.gold.package1 - (被界定在 developer页面 )相当于Y服用户X充值1000元宝。于是payment state为： 1000\_gold\_X\_Y (按照您的格式).**

**注： PAYMENT\_STATE不超过150个字符即可**

**- (NSString\*) getPaymentStateWithPackageID:(NSString \*) packageID;**

| **参数** | **描述** |
| --- | --- |
| packageID | packageID是已支付的 package ，packageID被界定在支付配置过程中 |

**2.4. 分析函数**

**Appota SDK 支持调用监视view和活动。监视信息好于游戏广告和监视。**

**Tracking View函数  
当您进入一个 view ，将该 view 发给 tracking**

**+ (void) sendViewWithName:(NSString\*) viewName;**

**Tracking event函数  
当用户做一个活动时将发送活动追踪**

**+ (void) sendEventWithCategory:(NSString\*) categoryName withEventAction:(NSString\*) action withLabel:(NSString\*) label withValue:(NSNumber\*) value;**

**界定一个操作的category, action, label**

**当用户做一个活动时将发送活动追踪**

**+ (void)sendEventWithCategory:(NSString \*)categoryName withEventAction:(NSString \*)action withLabel:(NSString \*)label;**

**界定一个操作的category, action, label**

**2.5. 推送通知函数（Push notification）**

**注册群推送通知的功能  
注册推送通知 (目前只支持iOS). 使用群名称来选择推送用户群(比如：只推送1服的用户，群名称将为 = "server 1")**

**+ (void) registerPushNotificationWithGroupName:(NSString\*) groupName;**

**推送通知跟数据配置**

**+ (NSString\*) configurePushNotificationWithTokenData:(NSData\*) deviceTokenData;**

**推送通知跟数据配置**

**玩家注册推送通知之后将回调推送通知跟数据配置。您将该函数添加到您的application delegate (一般是在 AppDelegate.m里)。**

**- (void)application:(UIApplication\*)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData\*)deviceToken**

**{**

**[AppotaGameSDK configurePushNotificationWithTokenData:deviceToken];**

**}**

**推送通知处理**

**+ (void) handlePushNotification:(NSDictionary \*)receiveDictionary;**

**您将该函数添加到application delegate (一般是在 AppDelegate.m里).**

**- (void)application:(UIApplication \*)application didReceiveRemoteNotification:(NSDictionary \*)userInfo{**

**[AppotaGameSDK handlePushNotification:userInfo];**

**}**

**2.6.在游戏里设置角色**

**当您要设置游戏里的角色具有server name, server ID , character name, character ID等信息的时候需要使用该函数**

**+ (void) setCharacterWithCharacterName:(NSString \*)characterName characterID:(NSString \*)characterID**

**serverName:(NSString \*)serverName serverID:(NSString \*)serverID**

**onCompleteBlock:(AppotaSDKDictionaryBlock ) completeBlock**

**onErrorBlock:(AppotaSDKErrorBlock ) errorBlock;**

**2.7. 其他函数**

**所需的其他函数都在[AppotaGameSDK+Advance-Class](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK+Advance-Class.md)。**

**邀请 facebook 好友玩游戏**

**调 [AppotaGameSDK inviteFacebookFriendsWithCompleteBlock:]; 邀请facebook好友玩游戏 (用户要登录才能使用该功能)。**

**- (void)inviteFacebookFriendsWithCompleteBlock:(AppotaInviteFriendBlock) inviteBlock;**

**显示 Facebook登录**

**调 [AppotaGameSDK showFacebookLogin]; 显示无Appota Login UI 的 Facebook 登录。**

**+ (void) showFacebookLogin;**

**显示 Google 登录**

**调 [AppotaGameSDK showGoogleLogin]; 显示无Appota Login UI 的Google 登录.**

**+ (void) showGoogleLogin;**

**显示 Twitter 登录**

**调 [AppotaGameSDK showTwitterLogin];显示无Appota Login UI显示Twitter 登录。**

**+ (void) showTwitterLogin;**

**3. 相关class资料**

* [**AppotaGameSDK-class**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK-class.md)
* [**AppotaGameSDKCallback-class**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDKCallback-class.md)
* [**AppotaUserLoginResult-class**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaUserLoginResult-class.md)
* [**AppotaPaymentResult**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaPaymentResult-class.md)
* [**AppotaGameSDK+Advance-Class**](https://github.com/appota/ios-game-sdk/blob/master/class-document/AppotaGameSDK+Advance-Class.md)

**4. FAQ 和Glossary**

* **IPN` 是Appota系统为玩家增加元宝的机制. 具体查看[IPN](https://github.com/appota/ios-game-sdk/wiki/Passive-Confirmation-via-IPN)**
* **PackageID每个游戏内的支付包应该附带 package ID (提供SDKTool) 来确认包。**
* **PaymentState****要在` getPaymentStateWithPackageID函数实施。使用`PackageID`和游戏服务器信息来建设 PaymentState.**
* **比如 packageID: com.gold.package1 - (在集成过程中确认)对应于1000元宝档位和Y服的X玩家。支付状态设为： com.gold.package1\_1000\_gold\_X\_Y (按您的格式).**