**1. 入门**

**配置开发环境**

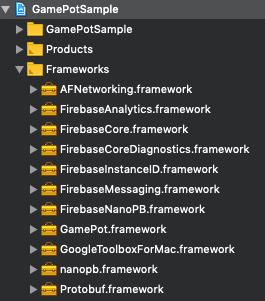
为了开发IOS APP首先需要安装开发工具（XCode）。

在IOS使用GamePot所需要的系统环境为如下。

[ 系统环境 ]

* 最低 : IOS 9.0 以上
* 开发环境 : XCode

**添加Freamwork**

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/system01.png)

通过拖动将下载的IOS SDK文件添加到Xcode项目文件夹目标中。

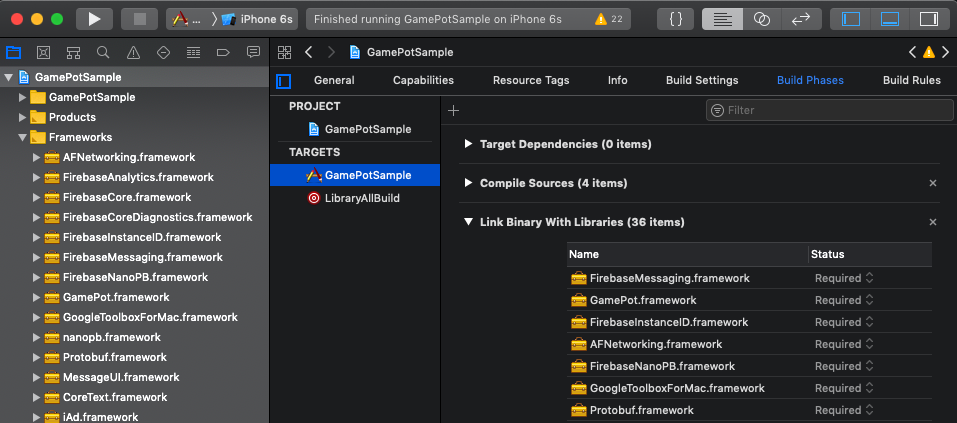
**添加Dependencies**

依照使用的功能，所需要的Dependencies列表有差异

请参考下列表格添加所需要的Dependencies

功能别Dependencies

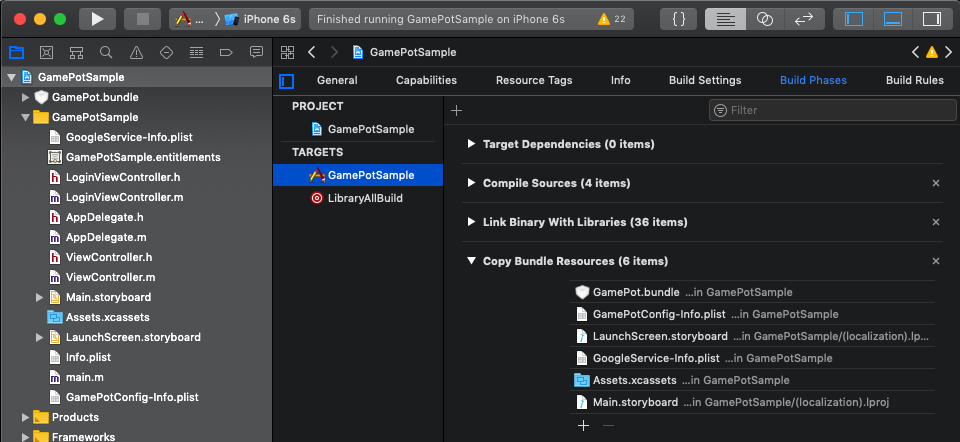
|  |
| --- |
|  |
| **Service** | **Framework** | **Dependencies** | **bundle** |
| 基本(Base) | AFNetworking.framework FirebaseAnalytics.framework FirebaseCore.framework FirebaseCoreDiagnostics.framework FirebaseInstanceID.framework FirebaseMessaging.framework FirebaseNanoPB.framework GamePot.framework GoogleToolboxForMac.framework nanopb.framework Protobuf.framework | libz.tbd WebKit.framework  UserNotifications.framework | GamePot.bundle |
| 登录(Login) | [ Base ] GamePotChannel.framework  [ Google Sign In ] GamePotGoogleSignIn.framework GoogleSignIn.framework GTMOAuth2.framework GTMSessionFetcher.framework  [ Facebook ] Bolts.framework FBSDKCoreKit.framework FBSDKLoginKit.framework GamePotFacebook.framework | [ Google Sign In ] SafariServices.framework [ Facebook ] SafariServices.framework | [ Google Sign In ] GoogleSignIn.bundle |
| 广告(AD) | [ Base ] GamePotAd.framework  [ Facebook ] Bolts.framework FBSDKCoreKit.framework  GamePotAdFacebook.framework  [Adbrix]  AdBrix.framework  GamePotAdAdbrix.framework  IgaworksCore.framework  [ Adjust ]  AdjustSdk.framework  GamePotAdAdjust.framework | [ Facebook ]  [ Adbrix ]  MessageUI.framework  libxml2.tbd  iAd.framework  CoreTelephony.framework  UIKit.framework  CoreGraphics.framework  CoreText.framework  MobileCoreServices.framework  SystemConfiguration.framework  Security.framework  [ Adjust ]  AdSupport.framework |  |
| GameCenter | GamePotGameCenter.framework |  |  |
| NaverCafe | AFNetworking.framework GamePotNavarCafe.framework NaverCafeSDK.framework | AVKit.framework AVFoundation.framework MediaPlayer.framework CoreMedia.framework AssetsLibrary.framework ImageIO.framework QuartzCore.framework ReplayKit.framework(设置为Optional) MobileCoreServices.framework SystemConfiguration.framework Security.framework WebKit.framework libNaverLogin.a NaverThirdPartyConstantsForApp.h NaverThirdPartyLoginConnection.h NLoginThirdPartyOAuth20InAppBrowserViewController.h NLoginThirdPartyOAuth20InAppBrowserViewController.m | NaverAuth.bundle NaverCafeSDK.bundle |
|  |  |  |  |

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/system02.png)

**添加Bundle Resource**

根据所要使用的功能，需要添加Bundle Resource文件。

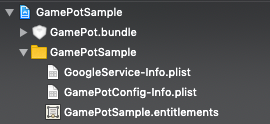
参考功能别Dependencies表格，添加Bundle Resource文件。

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/system03.png)

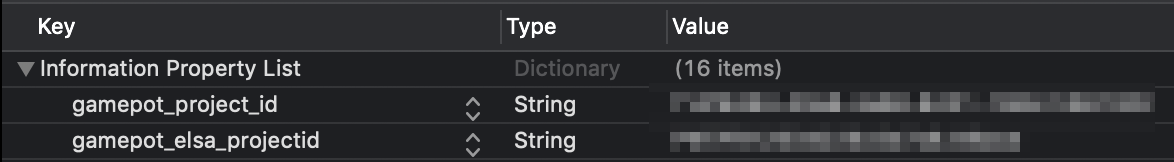
**添加InfoPlist**

基本上GamePot SDK使用Google Firebase。所以需要在项目添加设置Google Firebase后生成的GoogleService-Info.plist。

也需要添加包含着GamePot SDK的默认设置值的GamePotConfig-Info.plist文件。

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/system04.png)

**GamePotConfig-Info.plist 设置**

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/system06.png)

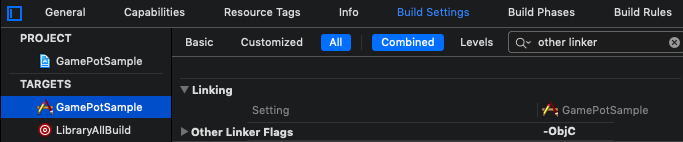
gamepot\_project\_id : Gamepot工程ID

gamepot\_elsa\_projectid : Gamepot日志工程ID (optional)

gamepot\_api\_url : Gamepot API URL

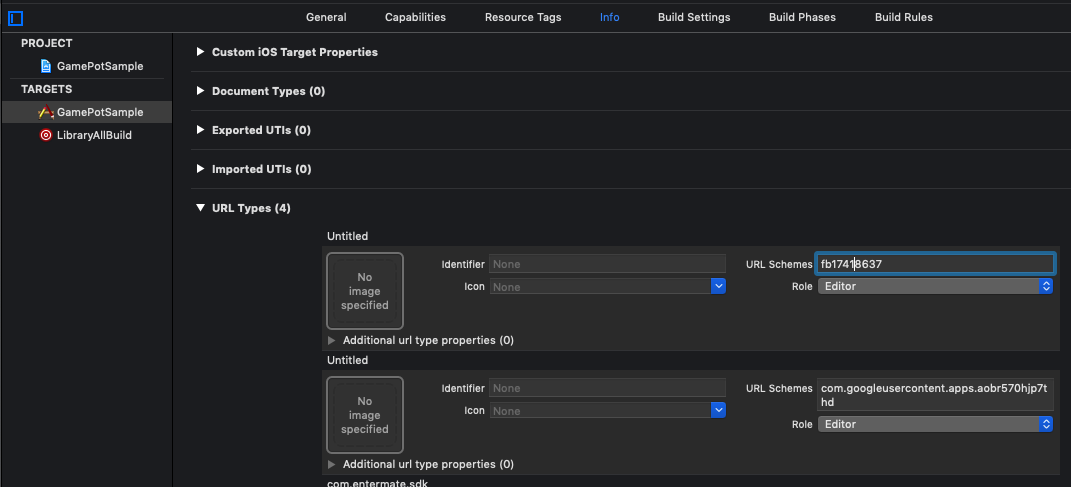
**添加Build选项**

Build Settings -> Linking -> Other Linker Flags 部分 –添加ObjC选项。

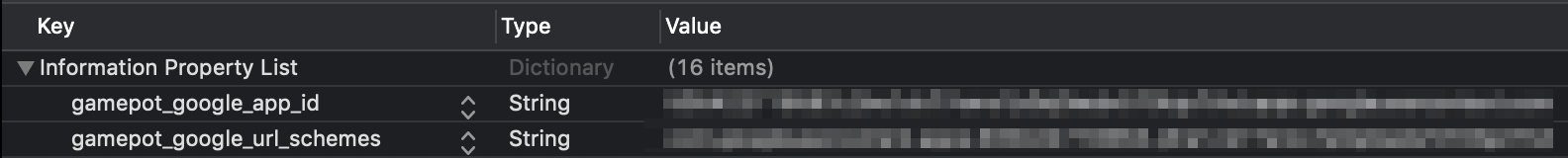
[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/system05.png)

**Google Sign In 登录环境设置**

参考功能别Dependencies表格中的Login >> Google Sign In。添加Framework和 Dependencies。

复制GoogleService-Info.plist文件的REVERSED\_CLIENT\_ID的值之后，在Info >> URL Types新加一个项目，然后在URL Schemes粘贴相关值。 [](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/login_google_setting.png)

**GamePotConfig-Info.plist 设置**

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/login_google_setting01.png)

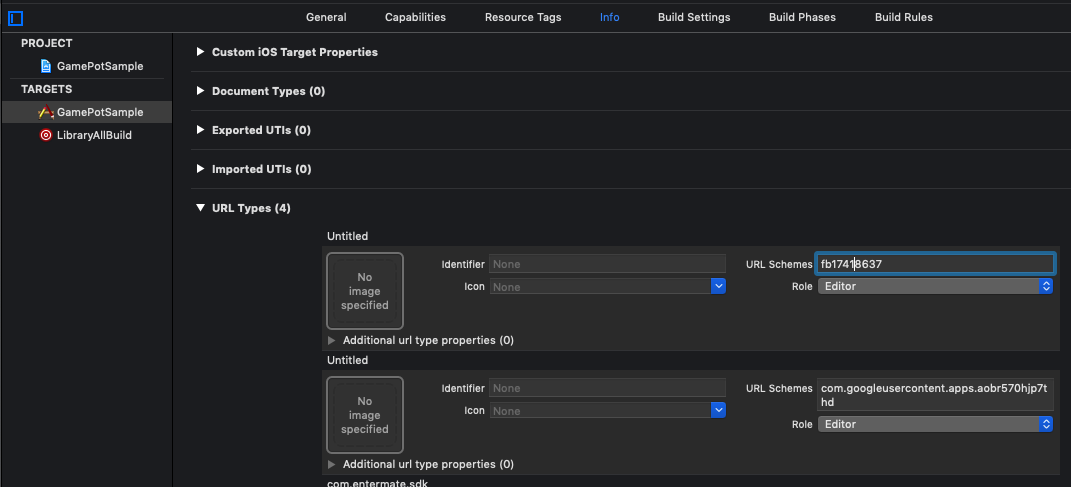
gamepot\_google\_app\_id : GoogleService-Info.plist 文件的 CLIENT\_ID 值

gamepot\_google\_url\_schemes : GoogleService-Info.plist 文件的 REVERSED\_CLIENT\_ID 值

**Facebook登录环境设置**

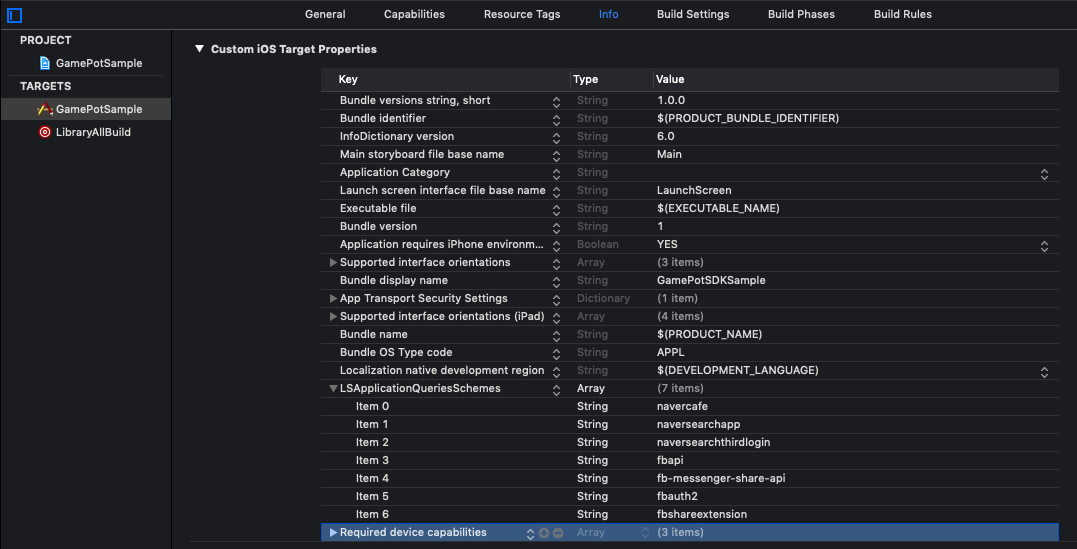
参考功能别Dependencies表格的Login >> Facebook。添加Framework和 Dependencies。

将Facebook App ID添加到Info >> URL Types中，添加格式为fb+Facebook App ID。

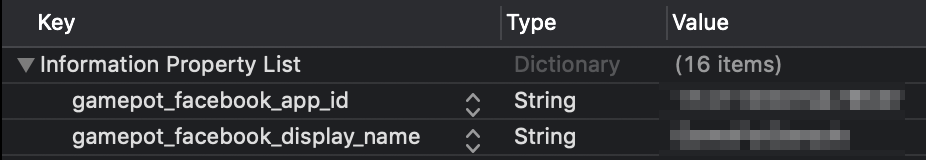
[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/login_google_setting.png)

Info >> iOS Target Property的LSApplicationQueriesSchemes中添加以下项目。

fbapi fb-messenger-share-api fbauth2 fbshareextension

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/login_facebook_setting.png)

**GamePotConfig-Info.plist 设置**

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/login_facebook_setting01.png)

gamepot\_facebook\_app\_id : Facebook App ID

gamepot\_facebook\_display\_name : Facebook display name

**2. 初始化**

AppDelegate文件里加上以下内容。

#import <GamePot/GamePot.h>

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

...

// GamePot SDK Initialize

[[GamePot getInstance] setup];

// Push Permission

if(SYSTEM\_VERSION\_GRATERTHAN\_OR\_EQUALTO(@"10.0"))

{

UNUserNotificationCenter \*center = [UNUserNotificationCenter currentNotificationCenter];

center.delegate = self;

[center requestAuthorizationWithOptions:(UNAuthorizationOptionSound | UNAuthorizationOptionAlert | UNAuthorizationOptionBadge) completionHandler:^(BOOL granted, NSError \* \_Nullable error){

if(!error){

dispatch\_async(dispatch\_get\_main\_queue(), ^{

[[UIApplication sharedApplication] registerForRemoteNotifications];

});

}

}];

}

else

{

// Code for old versions

UIUserNotificationType allNotificationTypes = (UIUserNotificationTypeSound | UIUserNotificationTypeAlert | UIUserNotificationTypeBadge);

UIUserNotificationSettings \*settings = [UIUserNotificationSettings settingsForTypes:allNotificationTypes categories:nil];

[application registerUserNotificationSettings:settings];

[application registerForRemoteNotifications];

}

...

}

// Push

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

{

...

[[GamePot getInstance] handleRemoteNotificationsWithDeviceToken:deviceToken];

...

}

- (void)applicationWillEnterForeground:(UIApplication \*)application {

[[GamePotChat getInstance] start];

}

- (void)applicationDidEnterBackground:(UIApplication \*)application {

[[GamePotChat getInstance] stop];

}

**3. 登录/退出/注销**

Google/Facebook/Naver等各种登录SDK可以集成使用。

**设置**

// AppDelegate.m

#import <GamePotChannel/GamePotChannel.h>

// 使用Google Login的时候

#import <GamePotGoogleSignIn/GamePotGoogleSignIn.h>

// 使用Facebook Login的时候

#import <GamePotFacebook/GamePotFacebook.h>

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

...

// GamePotSDK频道初始化，必须要对于所使用的频道别使用 addChannel才可以。

// Guest方式默认包含

// Google Login初始化

GamePotChannelInterface\* google = [[GamePotGoogleSignIn alloc] init];

[[GamePotChannelManager getInstance] addChannelWithType:GOOGLE interface:google];

// Facebook登录初始化

GamePotChannelInterface\* facebook = [[GamePotFacebook alloc] init];

[[GamePotChannel getInstance] addChannelWithType:FACEBOOK interface:facebook];

// 处理登录所需要。

[[GamePotChannel getInstance] application:application didFinishLaunchingWithOptions:launchOptions];

...

}

- (BOOL)application:(UIApplication \*)app openURL:(NSURL \*)url options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> \*)options

{

// 处理登录所需要。

BOOL nChannelManagerResult = [[GamePotChannel getInstance] application:app openURL:url options:options];

return nChannelManagerResult;

}

**登录**

点击登录按钮的时候调用

#import <GamePotChannel/GamePotChannel.h>

// 定义登录类型

// GamePotChannelType.GOOGLE

// GamePotChannelType.FACEBOOK

// GamePotChannelType.GUEST

// 点击Google登录按钮的时候调用

[[GamePotChannel getInstance] Login:GOOGLE viewController:self success:^(GamePotUserInfo\* userInfo) {

// 登录完成

} cancel:^{

// 登录过程中用户取消登录

} fail:^(NSError \*error) {

// 登录过程中出现错误

// TODO : 使用游戏内弹窗提示失败原因。

// TODO : 提示请使用[error localizedDescription]

}];

**自动登录**

GamePot支持自动登录

#import <GamePotChannel/GamePotChannel.h>

// 获取最终登录过的信息，根据此内容尝试自动登录

// lastLoginType : 可以获取最后登录的值

GamePotChannelType type = [[GamePotChannel getInstance] lastLoginType];

if(type != NONE)

{

// 使用最后登录的登录类型登录的方式

// 处理自动登录时，请调用为如下。

[[GamePotChannel getInstance] Login:type viewController:self success:^(GamePotUserInfo\* userInfo) {

} cancel:^{

} fail:^(NSError \*error) {

// TODO : 使用游戏内弹窗提示失败原因

// TODO : 提示内容请使用[error localizedDescription]

}];

}

else

{

// 没有最后登录的信息。自动移动至有登录按钮的界面。

}

**退出**

退出目前所登录的账号.

#import <GamePotChannel/GamePotChannel.h>

[[GamePotChannel getInstance] LogoutWithSuccess:^{

//退出完成之后移动至最初界面

} fail:^(NSError \*error) {

// 退出失败

// TODO : 请使用游戏内提示窗提示失败原因。

// TODO : 提示内容请使用[error localizedDescription]

}];

**注销**

注销目前登录的账号

#import <GamePotChannel/GamePotChannel.h>

[[GamePotChannel getInstance] DeleteMemberWithSuccess:^{

// 账号注销成功，移动至登录界面

} fail:^(NSError \*error) {

// 账号注销失败

// TODO : 请使用游戏内提示窗提示失败原因。

// TODO : 提示内容请使用[error localizedDescription]

}];

**4. 账号绑定**

一个游戏账号可以绑定/解除多个社交账户（Google/FaceBook等）的功能。（至少可以绑定一个社交账户）

游戏内需要开发相关绑定界面。在游戏界面用户点击绑定的时候调用以下代码。

**账号绑定**

可以使用Google / Facebook等账号绑定账号。

#import <GamePotChannel/GamePotChannel.h>

// 定义类型

// GamePotChannelType.GOOGLE

// GamePotChannelType.FACEBOOK

[[GamePotChannel getInstance] CreateLinking:GOOGLE viewController:self success:^(GamePotUserInfo \*userInfo) {

// TODO : 绑定完成。使用游戏内的弹窗提示绑定完成结果。 (ex. 계정 연동에 성공했습니다.)

} cancel:^{

// TODO : 用户取消

} fail:^(NSError \*error) {

// TODO : 绑定失败。使用游戏内的弹窗提示绑定失败原因。

// TODO : 提示内容请使用[error localizedDescription]。

}];

**被绑定的列表**

通过此API可以确认绑定状态

#import <GamePotChannel/GamePotChannel.h>

// 定义类型

// GamePotChannelType.GOOGLE

// GamePotChannelType.FACEBOOK

// 返回定义类型的绑定结果

BOOL isGoogleLinked = [[GamePotChannel getInstance] isLinked:GOOGLE];

// 对绑定的类型以JsonString返回。

NSString\* linkedList = [[GamePotChannel getInstance] getLinkedListJsonString];

**解除绑定**

解除目前已绑定的账号

#import <GamePotChannel/GamePotChannel.h>

[[GamePotChannel getInstance] DeleteLinking:GOOGLE success:^{

// TODO : 解绑完成。使用游戏内弹窗提示解绑完成结果。 (ex. 계정 연동을 해제했습니다.)

} fail:^(NSError \*error) {

// TODO : 解绑失败。使用游戏内弹窗提示解绑失败原因。

// TODO : 提示请使用[error localizedDescription]

}];

**5. 广告平台**

Facebook / Adjust / Adbrix等各种广告平台的SDK可以集成使用。

**设置**

// AppDelegate.m

#import <GamePotAd/GamePotAd.h>

// 使用Facebook 广告平台时

#import <GamePotAdFacebook/GamePotAdFacebook.h>

// 使用Adjust 广告平台时

#import <GamePotAdAdjust/GamePotAdAdjust.h>

// 使用Adbrix 广告平台时

#import <GamePotAdAdbrix/GamePotAdAdbrix.h>

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

...

// Facebook 广告平台初始化

GamePotAdInterface\* adFacebook = [[GamePotAdFacebook alloc] init];

[[GamePotAd getInstance] addAds:adFacebook];

// Adjust 广告平台初始化

GamePotAdInterface\* adAdjust = [[GamePotAdAdjust alloc] init];

[[GamePotAd getInstance] addAds:adAdjust];

// Adbrix 广告平台初始化

GamePotAdInterface\* adAdbrix = [[GamePotAdAdbrix alloc] init];

[[GamePotAd getInstance] addAds:adAdbrix];

...

}

### 使用Facebook广告平台的时候

除了上面初始化部分之外没有其他追加工作。

### 使用Adbrix广告平台的时候

在GamePotConfig-Info.plist放入Adbrix的key值

[image-20181017121606495](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/ad_adbrix01.png)

gamepot\_adbrix\_appid : Adbrix 앱 ID

gamepot\_adbrix\_hashkey : Adbrix Hash Key

## EventTracking传达

Event Tracking如下一样按照情况下所调用的代码会不同

请参考下面代码之后调用.

#import <GamePotAd/GamePotAd.h>

// 一般情况

TrackerEvent\* event = [[TrackerEvent alloc] init];

[event setEvent:@"test"];

// [event setAdjustKey:@"3m586u"]; // 使用Adjust的化请传达相关值

[[GamePotAd getInstance] tracking:EVENT obj:event];

// 角色升级的时候

TrackerLevel\* level = [[TrackerLevel alloc] init];

[level setLevel:@"12"];

// [level setAdjustKey:@"x7en7q"]; // 使用Adjust的化请传达相关值

[[GamePotAd getInstance] tracking:LEVEL obj:level];

// 完成新手教程的时候

TrackerTutorial\* tutorialEvent = [[TrackerTutorial alloc] init];

[tutorialEvent setContentData:@"튜토리얼 완료"];

[tutorialEvent setContentId:@"1"];

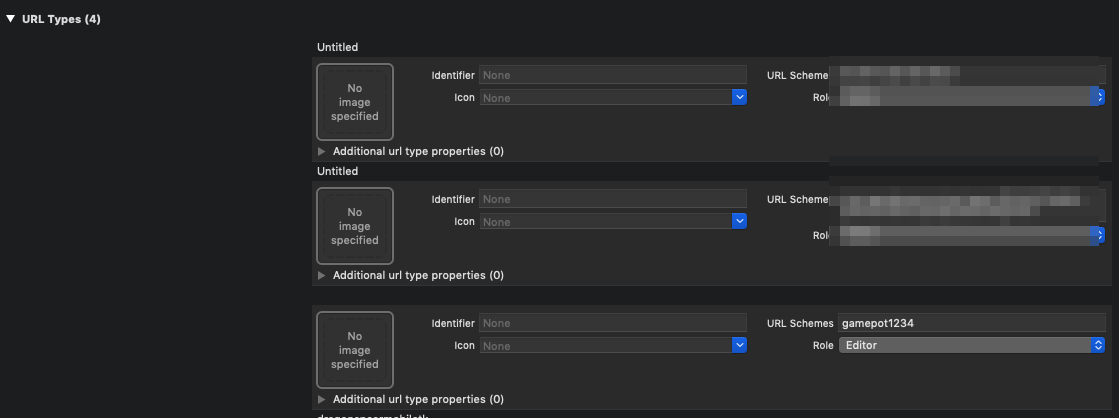
[tutorialEvent setSuccess:YES];

// [tutorialEvent setAdjustKey:@"byoplo"]; // 使用Adjust的化请传达相关值

[[GamePotAd getInstance] tracking:TUTORIAL\_COMPLETE obj:tutorialEvent];

## Deep Link

Inpo.plist >> URL types项目里请添加URL Schemes

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/deeplink001.png)

AppDelegate.m文件里请添加为如下

// AppDelegate.m

#import <GamePotAd/GamePotAd.h>

- (BOOL)application:(UIApplication \*)app openURL:(NSURL \*)url options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> \*)options

{

...

// 使用DeepLinking的时候加上

[[GamePotAd getInstance] application:app openURL:url options:options];

...

}

**6. 支付**

**设置**

支付的结果值以Delegate格式使用。所以请把Delegate添加为如下。

#import <GamePot/GamePot.h>

@interface ViewController () <GamePotPurchaseDelegate>

@end

@implementation ViewController

- (void)viewDidLoad

{

...

[[GamePot getInstance] setPurchaseDelegate:self];

...

}

- (void)GamePotPurchaseSuccess:(GamePotPurchaseInfo \*)\_info

{

// 支付成功

[[GamePotAd getInstance] tracking:BILLING obj:\_info];

}

- (void)GamePotPurchaseFail:(NSError \*)\_error

{

// 支付错误

// TODO : 使用游戏内弹窗提示失败原因。

// TODO : 提示请使用[error localizedDescription]

}

- (void)GamePotPurchaseCancel

{

// 尝试支付中取消

// 결제가 취소 되었습니다. 支付取消的韩文提示请使用游戏内弹窗提示给用户。

}

@end

**尝试支付**

#import <GamePot/GamePot.h>

// productid使用各个渠道的商品ID

[[GamePot getInstance] purchase:productid];

**7. 其他API**

**Naver论坛SDK**

为了使用此功能，首先需要对接所需要的Naver论坛SDK相关值。

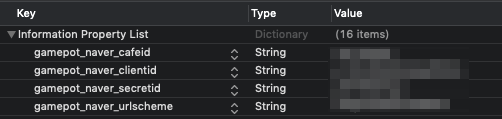
GamePotConfig-Info.plist 文件里请添加所需要的值.

gamepot\_naver\_cafeid // Naver论坛ID

gamepot\_naver\_clientid // 使用naver账号登录时所使用的client ID

gamepot\_naver\_secretid // 使用naver账号登录时所使用的secret ID

gamepot\_naver\_urlscheme // 使用naver账号登录时所使用的urlscheme

[](https://github.com/itsbio/GamePotGuide/blob/master/Android_iOS/assets_ios/navercafe001.png)

参考功能别Dependencies表格中的Naver Café项目。添加Framework和Dependencies。

// AppDelegate.m

#import <GamePotNavarCafe/GamePotNavarCafe.h>

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

...

// Naver论坛初始化

[[GamePotNaverCafe getInstance] setup];

...

}

- (BOOL)application:(UIApplication \*)app openURL:(NSURL \*)url options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> \*)options

{

BOOL nChannelManagerResult = [[GamePotChannelManager getInstance] application:app openURL:url options:options];

BOOL nNaverCafeResult = [[GamePotNaverCafe getInstance] application:app openURL:url options:options];

return nChannelManagerResult || nNaverCafeResult;

}

以下形式调用Naver论坛SDK

#import <GamePotNavarCafe/GamePotNavarCafe.h>

[[GamePotNaverCafe getInstance] start:self];

登录成功后，加上下面代码的化，可以在Naver论坛的管理员菜单识别用户。

#import <GamePotNavarCafe/GamePotNavarCafe.h>

[[GamePotNaverCafe getInstance] setUserId:[userInfo memberid]];

**兑换码**

#import <GamePot/GamePot.h>

[[GamePot getInstance] coupon:/\*用户所输入的兑换码\*/ handler:^(BOOL \_success, NSError \*\_error) {

if(\_success)

{

// TODO : 通过message返回兑换码使用结果。请使用游戏内弹窗提示相关内容。

}

else

{

// TODO : \_error返回兑换码使用失败的原因。

// [\_error localizedDescription]的内容请用游戏内弹窗提示给用户。

}

}];

## Push

#import <GamePot/GamePot.h>

// 开关推送 On / Off

[[GamePot getInstance] setPushEnable:YES success:^{

} fail:^(NSError \*error) {

}];

// 开关夜间推送 On / Off

[[GamePot getInstance] setNightPushEnable:YES success:^{

} fail:^(NSError \*error) {

}];

// 开关广告推送 On / Off

[[GamePot getInstance] setAdPushEnable:YES success:^{

} fail:^(NSError \*error) {

}];

// 推送 / 夜间推送 / 广告推送 一并设置

[[GamePot getInstance] setPushStatus:YES night:YES ad:YES success:^{

<#code#>

} fail:^(NSError \*error) {

<#code#>

}];

**成就**

成就是GameCenter的功能。首先需要接入GameCenter。

参考功能别Dependencies表格的GameCenter。添加Framework和 Dependencies。

以下初始化代码请添加到AppDelegate文件中。

// AppDelegate

#import <GamePotChannel/GamePotChannel.h>

#import <GamePotGameCenter/GamePotGameCenter.h>

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

...

// 请调用GamePot setup api之后调用。

GamePotChannelInterface\* gamecenter = [[GamePotGameCenter alloc] init];

[[GamePotChannel getInstance] addChannelWithType:GAMECENTER interface:gamecenter];

[[GamePotChannel getInstance] enableGameCenter:YES];

...

}

**成就列表界面**

点击游戏内成就按钮的时候调用如下

#import <GamePotChannel/GamePotChannel.h>

[[GamePotChannel getInstance] showAchievement:self];

**达到成就时**

达到成就条件的时候请调用以下代码.

#import <GamePotChannel/GamePotChannel.h>

[[GamePotChannel getInstance] achievementSubmitPercent:/\*成就ID\*/ percent:/\*成就达成率\*/ handler:^(id data, NSError \*error) {

if(error)

{

// Error 发生

return;

}

// 成功

}];