

iOS-SDK接入文档之Facebook分享篇

引入 SDK 文件

/**

FBSDK13.0.0 及以上版本已不再支持 Xcode12，须使用 xcode13.2.1 或以上版本接入。

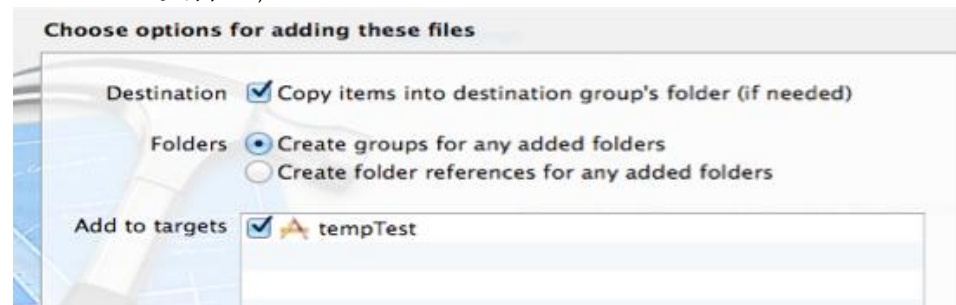
FBSDK15.1.0 及以上版本已不再支持 Xcode13，须使用 xcode14.0 或以上版本接入。

FBSDK17.0.0 及以上版本已不再支持 Xcode14，须使用 xcode15.0 或以上版本接入(FBSDK17.0.2 要求 Xcode15.3+)。

*/

将 SDK 文件夹下 FBSDKCoreKit.framework, FBAEMKit.framework, FBSDKCoreKit_Basics.framework, FBSDKShareKit.framework 文件引入到接入工程并选择正确的 target。

(如果已接入 Facebook 登录或者统计，相同的文件可忽略仅需引入不同的 framework 文件。)



FBSDK从17.3.0版本开始改用动态库版本，请以动态库的方式加载即把FBxxxx.framework设置为Embed & Sign。

检查target->Build Setting-> Runpath search path是否包含 @executable_path/Frameworks如果没有请添加，若有请继续下一步骤。

包含 UnityFramework 动态库特别说明:

FBSDKCoreKit.framework/FBAEMKit.framework/FBSDKCoreKit_Basics.framework/FBSDKShareKit.framework 为动态库 TargetMembership 需要关联到 UnityFramework 和 Unity_iPhone。

UnityFramework 仅关联即可，不需要也不可以设置为 Embed & Sign，否则提交 AppStore 时会报错。

Unity-iPhone 不仅需要关联，需要且必须设置为 Embed & Sign，否则启动就崩溃。

接入工程配置

FBSDK 需要接入工程支持 Swift 和 OC 混编。以下 2 种方案任选其一

方案 1: 需要在工程中创建一个 swift 文件, 然后根据 Xcode 提示创建 OC 和 swift 桥接文件。

方案 2: target->Build Settings->Library Search Path 添加以下语句:

```
$(TOOLCHAIN_DIR)/usr/lib/swift/$(PLATFORM_NAME)
```

```
$(SDKROOT)/usr/lib/swift
```

Target->Build Settings->Runpath Search Path 最上面添加以下语句:

请注意确保/usr/lib/swift 加在@executable_path/Frameworks 的上面, 否则游戏运行时可能会崩溃。

```
/usr/lib/swift
```

包含 UnityFramework 动态库特别说明:

所有的操作都在 UnityFramework 上进行。

方案 1: 需要在工程中创建一个 swift 文件, 可只关联到 UnityFramework 或同时关联到 UnityFramework 和 Untiy-iphone。

方案 2: UnityFramework->Build Settings->Library Search Path 添加以下语句:

```
$(TOOLCHAIN_DIR)/usr/lib/swift/$(PLATFORM_NAME)
```

```
$(SDKROOT)/usr/lib/swift
```

UnityFramework->Build Settings->Runpath Search Path 最上面添加以下语句:

请注意确保/usr/lib/swift 加在@executable_path/Frameworks 的上面, 否则游戏运行时可能会崩溃。

```
/usr/lib/swift
```

FacebookAppID:facebook 后台应用参数;

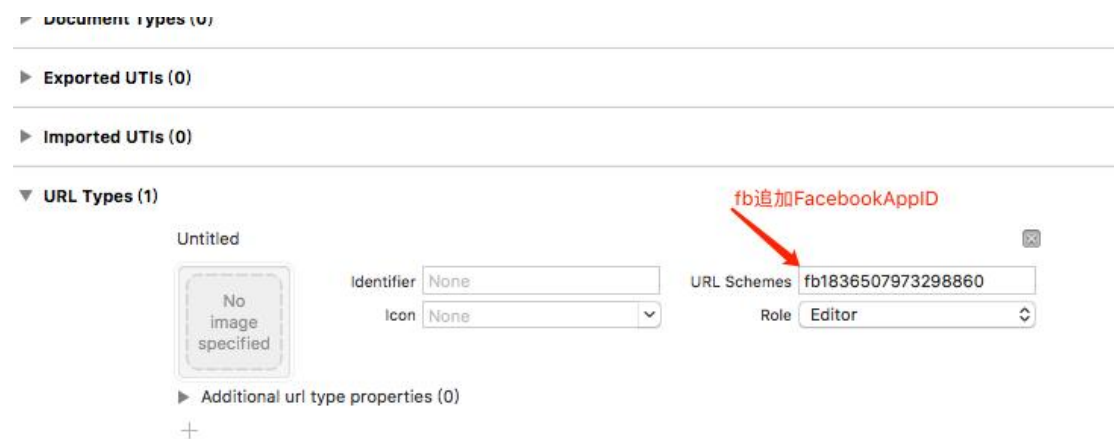
FacebookDisplayName:facebook 后台配置的应用名称{游戏名称};

FacebookClientToken:在 facebook 后台应用设置->高级->客户端口令查看;

LSApplicationQueriesSchemes:Facebook 相关白名单。

✓ LSApplicationQueriesSchemes	◇	Array	(5 items)	
Item 0		String	fbapi	
Item 1		String	fb-messenger-api	
Item 2		String	fbauth2	
Item 3		String	fbshareextension	
Item 4		String	fb-messenger-share-api	
DTSDKBuild	◇	String	19A339	
FacebookAdvertiserIDCollectionEnabled	◇	Boolean	1	
Bundle version string (short)	◇	String	1.0.0	
> CFBundleSupportedPlatforms	◇	Array	(1 item)	
> Supported interface orientations	◇	Array	(1 item)	
BuildMachineOSBuild	◇	String	21A559	
DTPlatformBuild	◇	String	19A339	
Bundle OS Type code	◇	String	APPL	
DTXcodeBuild	◇	String	13A1030d	
Localization native development region	◇	String	English	
MinimumOSVersion	◇	String	9.0	
Bundle version	◇	String	3	
Icon already includes gloss effects	◇	Boolean	YES	
Status bar is initially hidden	◇	Boolean	YES	
FacebookAppID	◇	String	{{FacebookAppID}}	替换成游戏的 fbappid
> UIDeviceFamily	◇	Array	(2 items)	
Launch screen interface file base name	◇	String	LaunchScreen	
Bundle identifier	◇	String	com.firefantasyxx.ios	
FacebookDisplayName	◇	String	{{FacebookDisplayName}}	游戏的 fb 应用名
DTXcode	◇	String	1310	
FacebookClientToken	◇	String	{{FacebookClientToken}}	游戏的 fbclienttoken

OpenUrl 回调



接口调用

引入 REDeLoginKit.h

```
#import <JYouLoginKit/REDeLoginKit.h>
```

Facebook 点赞和分享

函数：

```
/**
 * 服务端配置的 FB 分享
 */
+ (void)shareWith:(NSString *)serverId roleId:(NSString *)roleId;
```

功能：进入用户中心页面，显示关联其他平台账号的信息和操作入口。现在 FaceBook 已经不支持点赞了。

函数：

```
/**
 * 自定义 FB 活动分享
 * urlString:游戏自定义的活动分享地址
 */
+ (void)fbshareUrl:(NSString *)urlString;
```

功能：打开游戏自行开发的 Facebook 活动网址。

Facebook 原生图片分享

使用 FB 原生图片分享，需要在工程 info.plist 里面配置参数，具体配置参照 Facebook 登录部分。

1.1 直接分享图片

函数：

```

/**
 * FB 原生图片分享
 * image  分享的图片
 * caption 分享的照片创建标题,如果传 nil 则没有标题, FB 平台不允许预填写分享内容
这里填写了也不会 FB 里展示
 * tag 话题标签,可为空, 会显示在分享对话框中, 因此用户在发布之前可决定是否将它删除
 * completeResults 分享回调
    分享成功 status=0, error = nil
    分享失败 status=1, error != nil
    取消分享 status=2, error = nil
 * tip:
 1.照片大小必须小于 12MB
 2.用户需要安装版本 7.0 或以上的原生 iOS 版 Facebook 应用
 */
+ (void)fbSharePhoto:(UIImage *)image caption:(NSString *)caption
tag:(NSString *)tag completeWithResults:(void (^)(NSError
*error,NSInteger status))completeResults;

```

功能: 分享图片到 FB 动态里面

代码示例:

```

[REDeLoginKit fbSharePhoto:[UIImage imageNamed:@"1080-
1920.jpg"] caption:@"133" completeWithResults:^(NSError *error,
NSInteger status) {
    if (status == 0) {
        NSLog(@"分享成功");
    }else if(status == 1){
        NSLog(@"分享失败%@",error.description);
    }else{
        NSLog(@"用户取消了分享");
    }
}

```

tip:

- 1.照片大小必须小于 12MB 且不能中文命名
- 2.用户需要安装版本 7.0 或以上的原生 iOS 版 Facebook 应用

1.2 分享本地图片

```

/**
 * FB 原生链接图片分享
 * imageUrl 分享的图片本地链接, 如需分享线上图片, 需要先下载图片到本地
 * caption 分享的照片创建标题,如果传 nil 则没有标题, FB 平台不允许预填写分享内容
这里填写了也不会 FB 里展示
 * completeResults 分享回调
    分享成功 status=0, error = nil
    分享失败 status=1, error != nil
    取消分享 status=2, error = nil
 * tip:
 1.用户需要安装版本 7.0 或以上的原生 iOS 版 Facebook 应用

```

```

*/
+ (void)fbShareUrlPhoto:(NSString *)imageUrl caption:(NSString
*)caption tag:(NSString *)tag completeWithResults:(void (^)(NSError
*error, NSInteger status))completeResults;

```

功能：分享链接图片到 FB 动态里面

tip:

- 1.用户需要安装版本 7.0 或以上的原生 iOS 版 Facebook 应用

Facebook 链接分享

使用 FB 链接分享，需要在工程 info.plist 里面配置参数，具体配置参照 Facebook 登录部分。

函数：

```

/**
 * FB 链接分享
 * urlString 分享链接地址
 * quote 高亮文本，与链接一同显示，如文章中的醒目引文,可为空
 * tag 话题标签,可为空，会显示在分享对话框中，因此用户在发布之前可决定是否将它删除
 * completeResults 分享回调
    分享成功 status=0, error = nil
    分享失败 status=1, error != nil
    取消分享 status=2, error = nil
 */
+ (void)fbShareUrl:(NSString *)urlString quote:(NSString *)quote
tag:(NSString *)tag completeWithResults:(void (^)(NSError
*error, NSInteger status))completeResults;

```

功能：分享链接到 FB 动态里面

Facebook 视频分享

使用 FB 视频分享，需要在工程 info.plist 里面配置参数，具体配置参照 Facebook 登录部分。

```

/**
 * FB 分享照片库视频--只能分享照片库视频
 * videoAsset 分享视频 PHAsset
 * tag 话题标签,可为空，会显示在分享对话框中，因此用户在发布之前可决定是否将它删除
 * completeResults 分享回调
    分享成功 status=0, error = nil
    分享失败 status=1, error != nil
    取消分享 status=2, error = nil
 */

```

```

+ (void)fbSharevideoAsset:(PHAsset *)videoAsset
tag:(NSString *)tag completeWithResults:(void (^)(NSError
*error,NSInteger status))completeResults;
/**
 * FB 分享本地视频需要先保存视频到相册再分享--需要添加相册权限
 * videoPath 分享视频地址
 * previewPhoto 视频封面图片
 * completeResults 分享回调
    分享成功 status=0, error = nil
    分享失败 status=1, error != nil
    取消分享 status=2, error = nil
 */
+ (void)fbSharevideoWithVideoPath:(NSString *)videoPath
tag:(NSString *)tag completeWithResults:(void (^)(NSError
*error,NSInteger status))completeResults;

```

应用跳转回调(必接)

类: REDeLoginKit

函数: +(void)application:(UIApplication *)application openURL:(NSURL *)url
options:(NSDictionary *)options;

功能: 处理第 3 方应用回调结果

在- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> *)options; 中调用。

类: REDeLoginKit

函数: +(void)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)source application:(id)annotation;

功能: 处理第 3 方应用回调结果

在- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)source application:(id)annotation; 中调用。

示例:

```

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

```

```
- (BOOL)application:(UIApplication *)application
openURL:(NSURL *)url
options:(NSDictionary<UIApplicationOpenURLOptionsKey,id>
*)options {
    [REDeLoginKit application:application openURL:url
options:options];
    return YES;
}
```