资讯通手游iOS SDK接入手册

v1.0

技术支持：资讯通手游

目录

[资讯通手游iOS SDK接入手册 1](#_Toc526066992)

[1. 申请资讯通手游参数gameCode和platformId 3](#_Toc526066993)

[2. 集成SDK到xcode工程中 3](#_Toc526066994)

[2.1 添加SDK目录到工程 3](#_Toc526066995)

[2.2 设置Bitcode选项 4](#_Toc526066996)

[2.3 设置Other Linker Flags 4](#_Toc526066997)

[2.4 适配HTTPS网络控制 5](#_Toc526066998)

[2.5 选择支持第三方支付 5](#_Toc526066999)

[3. SDK接口说明 5](#_Toc526067000)

[3.1 初始化接口 5](#_Toc526067001)

[3.2 登录接口 5](#_Toc526067002)

[3.3 支付接口 6](#_Toc526067003)

[3.4 注销接口 6](#_Toc526067004)

[3.5 上报角色信息接口 6](#_Toc526067005)

[3.6 获取SDK版本号接口 7](#_Toc526067006)

[4. SDK回调接口说明 7](#_Toc526067007)

[4.1 初始化成功回调 7](#_Toc526067008)

[4.2 登录成功回调 7](#_Toc526067009)

[4.3 支付成功回调 8](#_Toc526067010)

[4.4 支付失败回调 8](#_Toc526067011)

[4.5 支付取消回调 9](#_Toc526067012)

[4.6 注销回调 9](#_Toc526067013)

[5. 调用周期函数 9](#_Toc526067014)

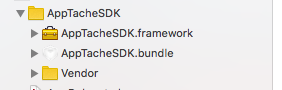
\*注意: sdk只⽀支持真机上运⾏，第三方库文件需要在ARC环境运行

# 申请资讯通手游参数gameCode和platformId

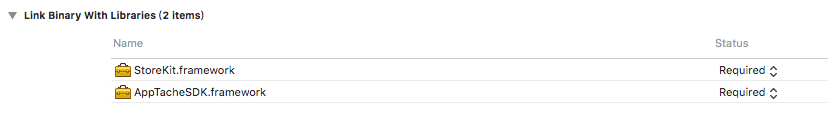
# 集成SDK到xcode工程中

## 2.1 添加SDK目录到工程

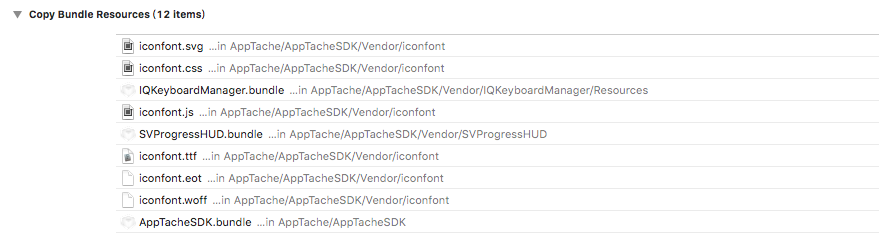
将SDK包中所有文件加入到工程中。



加入所需依赖库：

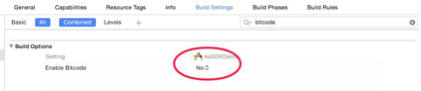


导入第三方Vendor中的资源文件：



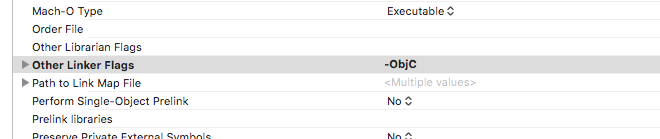
## 2.2 设置Bitcode选项

Build Settings->Builid options->Enable Bitcode->设置为 No。如图所示:

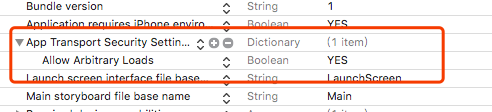


## 2.3 设置Other Linker Flags

在Build Settings->Builid setting-> Other Linker Flags 添加 -ObjC。如图所示:

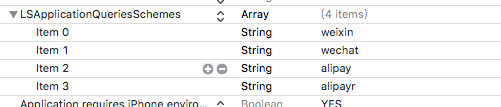


## 2.4 适配HTTPS网络控制



## 2.5 选择支持第三方支付

如果需要支持第三方支付需要设置微信和支付宝白名单，由服务端控制



# SDK接口说明

## 3.1 初始化接口

[[AppTacheSDK sharedInstance] initSdkWithGameCode: @"xxxxxx" platformId:@"xxxx"];

参数： gameCode 和 platformId 由资讯通分配。

## 3.2 登录接口

[[AppTacheSDK sharedInstance] login];

## 3.3 支付接口

[[AppTacheSDK sharedInstance] requestPay:params];

参数：params – 支付参数列表，NSDictionary，对应key值：

|  |  |
| --- | --- |
| 参数名 | 说明 |
| itemId | 商品id，类型：NSString |
| itemName | 商品名称，类型：NSString |
| roleId | 角色id，类型：NSString |
| roleName | 角色名，类型：NSString |
| serverId | 服务器id，类型：NSString |
| serverName | 服务器名称，类型：NSString |
| cpOrderNo | 订单号，类型：NSString |
| price | 支付金额，类型：NSString ，单位分，如@”100”表示1元 |
| currency | 货币类型，类型：NSString @”CNY” @”USD” |
| extInfo | 透传参数，类型：NSString |

## 3.4 注销接口

[[AppTacheSDK sharedInstance] logout];

## 3.5 上报角色信息接口

[[AppTacheSDK sharedInstance] addDataType:type params:dict];

① 参数：type – 上报角色类型，ATDataType，对应值：

ATDataTypeCreateRole, // 角色创建

ATDataTypeEnterGame, // 角色进入游戏

ATDataTypeLevelUp, // 角色升级

② 参数：dict – 上报角色参数列表，NSDictionary，对应key值：

|  |  |
| --- | --- |
| 参数名 | 说明 |
| cpRoleId | 角色id，类型：NSString |
| cpRoleName | 角色名，类型：NSString |
| cpServerId | 服务器id，类型：NSString |
| cpServerName | 服务器名称，类型：NSString |
| level | 角色等级，类型：NSNumber，如：12级传@12 |

## 3.6 获取SDK版本号接口

[[AppTacheSDK sharedInstance] getSdkVersion];

获取SDK版本号，返回字符串，如: @"1.0.0"

# SDK回调接口说明

遵循代理 – AppTacheSDKDelegate

[[AppTacheSDK sharedInstance] setDelegate:self];

实现代理接口：

## 4.1 初始化成功回调

- (void)appTacheSdkDidInitSuccess;

## 4.2 登录成功回调

- (void) appTacheSdkDidLoginSuccess: (NSDictionary\*) result;

登录信息result: 字典类型，key值说明：

success: 返回码, 0-登录失败，1-登录成功

msg: 返回消息，NSString

fid: 登录账号唯一标识，NSString

account: 登录账号，NSString

token: token信息，NSSting，只有当code为 1 时有值

## 4.3 支付成功回调

- (void)appTacheSdkDidPaySuccess:(NSDictionary \*)result;

支付信息result: 字典类型，key值说明：

success: 返回码, NSNumber 1-支付成功

msg: 返回消息，NSString

cpOrderId: cp支付时传入的订单号，NSString

orderId: 资讯通订单号，NSString

extInfo: 透传参数，NSSting

price: 支付金额，NSNumber, 单位：分

payMethodCode: 支付方式，NSString， wxh5pay – 微信支付，alipay – 支付宝支付，applepay – 苹果内购

## 4.4 支付失败回调

- (void)appTacheSdkDidPayFailed: (NSDictionary \*) result;

支付信息result: 字典类型，key值说明：

success: 返回码, NSNumber 1-支付成功

msg: 返回消息，NSString

cpOrderId: cp支付时传入的订单号，NSString

orderId: 资讯通订单号，NSString

extInfo: 透传参数，NSSting

price: 支付金额，NSNumber, 单位：分

payMethodCode: 支付方式，NSString， wxh5pay – 微信支付，alipay – 支付宝支付，applepay – 苹果内购

## 4.5 支付取消回调

- (void)appTacheSdkDidPayCancel: (NSDictionary \*) result;

支付信息result: 字典类型，key值说明：

success: 返回码, NSNumber 1-支付成功

msg: 返回消息，NSString

cpOrderId: cp支付时传入的订单号，NSString

orderId: 资讯通订单号，NSString

extInfo: 透传参数，NSSting

price: 支付金额，NSNumber, 单位：分

payMethodCode: 支付方式，NSString， wxh5pay – 微信支付，alipay – 支付宝支付，applepay – 苹果内购

## 4.6 注销回调

- (void)appTacheSdkDidLogoutSuccess;

# 调用周期函数

[[AppTacheSDK sharedInstance] application:application didFinishLaunchingWithOptions:launchOptions];

[[AppTacheSDK sharedInstance] application:application openURL:url options:options];

[[AppTacheSDK sharedInstance] application:application openURL:url sourceApplication:sourceApplication annotation:annotation];

[[AppTacheSDK sharedInstance] application:application handleOpenURL:url];