

4399 OMSDK 1.11.0 接入说明

4399 OMSDK 1.11.0 接入说明

一、接入说明

- 1.运行环境要求
- 2.接入步骤

二、客户端接口文档

- 1、初始化配置（新，增加游戏协议）（必须接入）
- 2、初始化配置（废弃）（必须接入）
- 3、URL回执（必须接入）
- 4、登录（必须接入）
- 5、登出
- 6、展示个人中心页面（合规化，必须接入）
- 7、检查手机绑定状态
- 8、开启绑定手机页面
- 9、展示在线客服
- 10、展示游戏客服弹窗
- 11、预充值许可（若有充值且不使用SDK内置充值接口，必须接入）
- 12、从SDK获取IAP商品
- 13、从SDK发起内购
- 14、从SDK发起内购（废弃）
- 15、前往第三方平台
- 16、打开评分弹窗
- 17、角色创建上报
- 18、进入游戏上报
- 19、上报事件

三、服务端接口说明

- 用户充值成功通知（如有充值，且不使用13接口的必须接入）
- 用户信息校验（可选接入）

四、Q&A

- 打包**IPA processing failed**错误提示
- 未显示手机号一键登录
- 未显示Apple ID登录渠道
- 未显示QQ登录渠道
- QQ登录未跳转回游戏
- 未显示第三方登录渠道
- iOS 12.2以下的系统启动即闪退
- 上架 Appstore 因 Guideline 5.1.1 被拒绝
- 切换Xcode 15打包后，上传TestFlight后出现启动即闪退或显示与此设备不兼容
- 如何接入IAP

四、更新日志

一、接入说明

1.运行环境要求

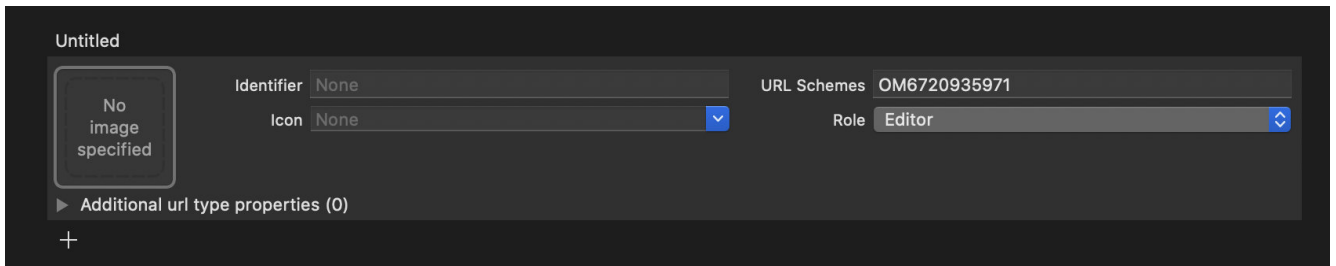
- SDK最低系统要求**iOS 8**，游戏最低系统要求应不低于iOS 8，否则将无法接入。
- 微信登录最低系统要求**iOS 11**，手机系统低于该系统时不展示该登录渠道。

- Apple ID登录最低系统要求**iOS 13**, 手机系统低于该系统时不展示该登录渠道。
- 手机号一键登录最低系统要求**iOS 9**, 手机系统低于该系统时不展示该登录渠道。

2.接入步骤

- (1) 请确保有4399运营分配的4399 GameKey用于SDK的配置和初始化
- (2) 配置URL Schemes

在Xcode -> TARGETS -> [主工程Target] -> Info -> URL Type 中点击 + 添加一个新的Item, 在 URL Schemes 中填入 OM[GameKey], 其中 [GameKey] 为步骤(1)中获取的key。例如 OM6720935971, 中间请勿加入任何分隔符。未完成本步骤, 可能造成QQ/微信登录无法使用。



- (3) 在 Xcode -> TARGETS -> [主工程Target] -> Info -> Custom iOS Target Properties 中配置隐私权限提示语。
 - **Privacy - Media Library Usage Description** 实名认证功能需要您同意, 才能访问媒体库。
 - **Privacy - Camera Usage Description** 实名认证功能需要您同意, 才能使用摄像头。
 - **Privacy - Photo Library Usage Description** 实名认证功能需要您同意, 才能访问相册。

游戏中如果已经有上述隐私描述, 则无需修改。未添加可能导致游戏闪退。
- (4) 关闭ATS, 在Xcode -> TARGETS -> [主工程Target] -> Info -> Custom iOS Target Properties, 添加 App Transport Security Settings, 并为其添加子项目 Allow Arbitrary Loads 设置为 YES
- (5) 增加URLSchemes白名单, 用于保障QQ登录正常运行。在Info.plist中添加 LSApplicationQueriesSchemes 类型Array, 添加String类型的子项 mq, sso4399, gamehotspot, weixinULAPI, wechat, weixin, 已有其他白名单的游戏, 直接在原有基础上新增即可。

也可在Info.plist中编辑完成(2)(3)(4)(5)步

```
<!--URL Schemes-->
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Editor</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>OM62019492383</string><!--请根据实际Gamekey配置本行-->
    </array>
  </dict>
</array>
<!--ATS-->
<key>NSAppTransportSecurity</key>
```

```

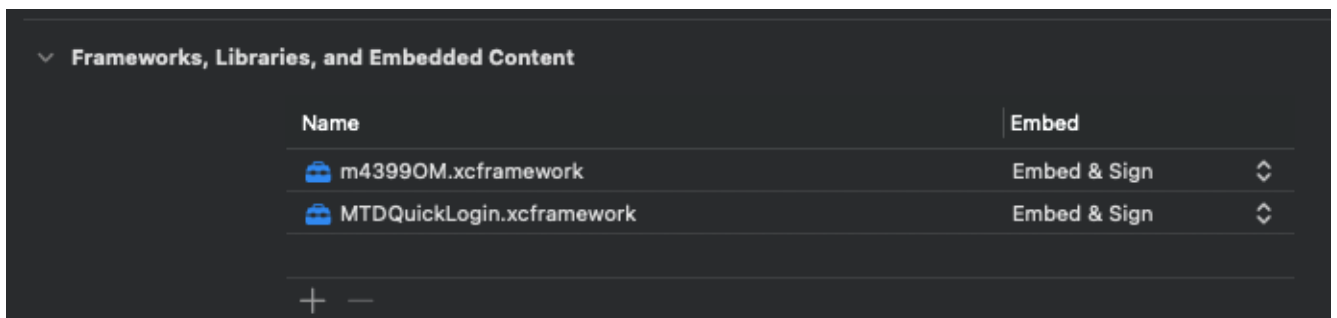
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
<!--隐私权限说明-->
<key>NSAppleMusicUsageDescription</key>
<string>实名认证功能需要您同意，才能访问媒体库。</string>
<key>NSCameraUsageDescription</key>
<string>实名认证功能需要您同意，才能访问相机。</string>
<key>NSPhotoLibraryUsageDescription</key>
<string>实名认证功能需要您同意，才能访问相册。</string>
<!--Schemes白名单-->
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>mqq</string>
  <string>sso4399</string>
  <string>gamehotspot</string>
  <string>weixinULAPI</string>
  <string>wechat</string>
  <string>weixin</string>
</array>

```

- (6) 关闭Bitcode，在 Xcode -> TARGETS -> Build Settings中，`Enable Bitcode` 设置为 `No`。Xcode 14后已弃用，如果为14以上打包请忽略
- (7) 打开Swift支持，在 Xcode -> TARGETS -> Build Settings中，`Always Embed Swift Standard Libraries` 设置为 `Yes`
- (8) 将下列SDK依赖文件引入工程
 - m4399OM.xcframework
 - MTDQuickLogin.xcframework
 - EAccountOpenPageResource.bundle

EAccountHYSDK.xcframework仅作为索引，在报'**EAccountHYSDK/EAccountOpenPageConfig.h'** file not found 错误时引入。

- (9) 在 Xcode -> TARGETS -> [主工程Target] -> General -> Framework, Libraries, and Embedded Content中，将 `m4399OM.xcframework`，`MTDQuickLogin.xcframework` 的Embed选项设置为 `Embed & Sign`



如有引入EAccountHYSDK.xcframework，EAccountHYSDK.xcframework配置为Do Not Embed

- (10) 配置Apple ID 登录
 - 登录Apple Developer Center, 在 Certificates, Identifiers & Profiles -> Identifiers 中找到游戏的Bundle ID, 进入配置。
 - 勾选 Sign In with Apple, 如无特殊需求, 在Configuration界面配置 Enable as a primary App ID
 - 重新生成配置描述文件.mobileprofiles, 并下载覆盖本地配置描述文件。
 - 在Xcode中 Signing & Capabilities 中按 + 添加 Sign In with Apple 能力描述

二、客户端接口文档

1、初始化配置（新，增加游戏协议）（必须接入）

```
+ (void)preConfig:(m4399OMManagerConfig *)config gameProtocolState:(void (^)(NSInteger code))gameProtocolState lastLoginUser:(void (^)(OMUser * _Nullable user))lastLoginUser switchAccount:(void (^)(void))switchAccount;
```

请务必在 AppDelegate 的 - application: didFinishLaunchingWithOptions: 中调用本接口，确保配置信息正确加载。

新旧初始化接口二选一，请勿重复调用

m4399OMManagerConfig参数说明

请使用 - (instancetype)initWithGamekey:(NSString *)gamekey 或 - (instancetype)initWithGamekey:(NSString *)gamekey customAnalysisAttributes:(NSDictionary<NSString *, NSString * _Nullable > *)customAnalysisAttributes 初始化Config, customAnalysisAttributes 用于定制的全局分析字段，与预制字段冲突时会使用指定字段。

参数名	类型	默认值	说明
gameKey	NSString		4399 OMSDK Key， 请向运营人员索取。
printLog	Bool	false	打印OMSDK日志
debugHeartbeatInterval	Double	-1	强制心跳间隔，用于开发期间调试使用，切勿在线上环境修改本参数。小于0时失效。

gameProtocolState回调说明

游戏协议回调，code定义：0 用户同意协议；160001 协议无更新通知；

lastLoginUser回调说明

该回调会返回上一次运行时用户使用的身份信息，如果用户未手动注销，并且在本次启动时校验身份后，会调用本回调返回用户信息（可能为空）。仅会在配置接口调用后最多调用一次。

若出现网络异常、防沉迷系统异常等情况，用户信息可能为空，此时可以调用登录接口进行登录。

若网络波动，此回调可能会较慢返回。

switchAccount回调说明

用户因防沉迷踢出或主动注销账号、用户在别处登录/修改密码、切换账号等情况，会调用本回调。游戏方在接收到该回调时，应注销游戏中存储的用户信息，返回登录界面，无需再调用logout接口。此时如需弹出SDK的登录界面，可在本回调中调用login接口。本回调可能在游戏运行期间随时被调用。

2、初始化配置（废弃）（必须接入）

```
+ (void)config:(m4399OMManagerConfig *)config lastLoginUser:(void (^)(OMUser *_Nullable user))lastLoginUser switchAccount:(void (^)(void))switchAccount;
```

请务必在 AppDelegate 的 - application: didFinishLaunchingWithOptions: 中调用本接口，确保配置信息正确加载。

m4399OMManagerConfig参数说明

请使用initWithGameKey初始化config

参数名	类型	默认值	说明
gameKey	NSString		4399 OMSDK Key， 请向运营人员索取。
printLog	Bool	false	打印OMSDK日志

lastLoginUser回调说明

该回调会返回上一次运行时用户使用的身份信息，如果用户未手动注销，并且在本次启动时校验身份后，会调用本回调返回用户信息（可能为空）。仅会在配置接口调用后最多调用一次。

若出现网络异常、防沉迷系统异常等情况，用户信息可能为空，此时可以调用登录接口进行登录。

若网络波动，此回调可能会较慢返回。

switchAccount回调说明

用户因防沉迷踢出或主动注销账号、用户在别处登录/修改密码、切换账号等情况，会调用本回调。游戏方在接收到该回调时，应注销游戏中存储的用户信息，返回登录界面，无需再调用logout接口。此时如需弹出SDK的登录界面，可在本回调中调用login接口。本回调可能在游戏运行期间随时被调用。

3、URL回执（必须接入）

```
+ (void)handlerUrl:(NSURL *)url;
```

在 AppDelegate 的 -application:openURL:options: 方法中调用本方法，传入方法参数openURL。该方法保障QQ登录功能正常运行。

如果为 SenceDelegate 接入，在 -scene:openURLContexts: 调用本方法，保留 AppDelegate 的接入。

4、登录（必须接入）

```
+ (void)loginWithSuccess:(void (^)(OMUser *_Nullable user))success fail:(void (^)(int code, NSString *_Nonnull msg))fail;
```

在用户需要进行登录时调用本接口，会在应用最顶层window中弹出登录界面。

如果为应用启动时的调用，则需等待初始化接口中的 `lastLoginUser` 回调执行后，方可调用本接口，否则可能引起防沉迷系统异常。

success登录成功回调，在用户登录成功后调用，返回用户登录信息。

fail登录失败回调，在登录遇到错误时调用，返回错误码和错误信息。

5、登出

```
+ (void)logout;
```

在用户需要进行登出时调用本接口。

注：调用本接口时，同样会触发初始化配置接口的 `switchAccount` 回调

6、展示个人中心页面（合规化，必须接入）

```
+ (void)userCenter:(void (^)(void))didClose;
```

在需要启动用户中心时调用本接口，会在应用最顶层window中弹出个人中心。

仅限用户登录后使用本接口。

didClose窗口关闭回调。

7、检查手机绑定状态

```
+ (void)checkPhoneBindingStatus:(void (^)(BOOL isBinding))success fail:(void (^)(NSInteger code, NSString *message))fail;
```

在需要检查用户手机绑定状态时调用本接口，会返回是否绑定手机

仅限用户登录后使用本接口。

success手机绑定状态回调。

fail手机绑定检查失败回调，返回错误码和错误信息。

8、开启绑定手机页面

```
+ (void)bindPhoneNumberWithSuccess:(void (^)(void))success fail:(void (^)(NSInteger code, NSString *message))fail;
```

在需要绑定手机时调用本接口，会在应用最顶层window中弹出绑定手机页面。

仅限用户登录后使用本接口。

success手机绑定成功回调。

fail手机绑定失败回调，返回错误码和错误信息。

Fail Code	含义
201	用户已绑定手机号
901	网络错误
902	用户取消绑定手机
606/607/608	用户在其他设备登录

9、展示在线客服

```
+ (void)onlineCustomerService:(void (^)(void))didClose;
```

在需要在线客服时调用本接口，会在应用最顶层window中弹出客服界面。

10、展示游戏客服弹窗

```
+ (void)customerService:(void (^)(void))didClose;
```

在用户需要联系客服时调用本接口，会在应用最顶层window中弹出客服界面。

didClose窗口关闭回调。

11、预充值许可（若有充值且不使用SDK内置充值接口，必须接入）

```
+ (void)prepayWithMoney:(int)money mark:(NSString *)mark success:(void (^)(NSString *orderId))success fail:(void (^)(int code, NSString *message))fail;
```

在用户发起充值后，优先调用本接口获得用户充值许可，以确保用户在符合防沉迷规则的前提下进行充值行为。

仅限用户登录后使用本接口。

money 充值金额（元）最小支持1元，仅支持整数。

mark 游戏方自定义订单号，用于关联游戏方单号和4399充值中心订单号

success 允许充值回调，返回orderId用于后续服务端接口的操作。

fail 充值失败回调，返回失败code和相关信息，此时有可能会根据防沉迷系统的需要，弹出防沉迷说明弹窗。

12、从SDK获取IAP商品

```
+ (void)fetchProductWithSuccess:(void (^)(NSArray<OMProduct *> *products))success fail:(void (^)(OMPaymentError *error))fail
```

获取IAP商品。

支持iOS12+。

使用IAP需在 Xcode工程->Target->Frameworks, Libraries, and Embedded Content中引入StoreKit.framework。
然后在Signing & Capabilities->左上角的 + Capability ->增加In-App Purchase 能力。

success 商品获取成功回调

OMProduct参数说明

参数名	类型	默认值	说明
name	NSString		商品名称
productID	NSString		商品productID
price	NSString		仅用于UI展示，请勿依赖price字段校验订单，在某些情况下price可能会显示错误地区的价格。由于float存在精度问题，计算时请注意
currency	NSString		货币，仅用于UI展示，在某些情况下currency可能会返回错误地区的货币
currencySymbol	NSString		货币符号，仅用于UI展示，在某些情况下currencySymbo可能会返回错误地区的货币符号

fail 失败回调

OMPaymentError

参数名	类型	默认值	说明
code	NSString		错误码
message	NSString		错误信息
info	NSString		错误补充信息，可为空
originalCode	NSString		原始错误码
orderId	NSString		订单id，可为空

13、从SDK发起内购

```
+ (void)purchaseWithProductID:(NSString *)productID mark:(NSString *)mark roleInfo:
(OMRoleInfoModel *_Nullable)roleInfo purchasingStatu:(void (^)(OMPurchasingStatu
statu))purchasingStatu success:(void (^)(OMPurchaseResultInfo *info))success userCancel:
(void (^)(OMPaymentError *error))userCancel other:(void (^)(OMPaymentError *error))other
```

发起应用内购买。请在调用获取IAP商品接口，并得到商品列表后调用此接口发起内购。

支持iOS12+。

仅限用户登录后使用本接口。

productID 内购商品id，请从「获取IAP商品」接口的商品列表中获取。

mark 游戏方自定义订单号，用于关联游戏方单号和4399充值中心订单号

roleInfo 充值角色信息，可以为nil，详细字段说明参「角色创建上报」接口附带的参数说明

purchasingStatu 支付中状态回调，OMPurchasingStatuPlaceOrder 下单中、OMPurchasingStatuVerifying 验单中、OMPurchasingStatuRequestIAP 正在拉起IAP。

success 成功回调，请在此向服务端验证内购状态，请勿依据本回调发放商品。

OMPurchaseResultInfo参数说明

参数名	类型	默认值	说明
orderId	NSString		订单id，可能为空
mark	NSString		游戏方自定义订单号，用于关联游戏方单号和4399充值中心订单号

userCancel 用户取消

other 失败/未知状态回调，参考商品获取说明的Fail code，即使SDK返回此回调，仍有可能通过补单找回购买记录，请勿依据本回调发放商品，请在此向服务端验证内购状态。

OMPaymentError

参数名	类型	默认值	说明
code	NSString		错误码
message	NSString		错误信息
info	NSString		错误补充信息，可为空
originalCode	NSString		原始错误码
orderId	NSString		订单id，可为空

Fail Code

Fail Code	含义
10001	未登录
10002	找不到Product
10003	系统不支持
1001	失败,用户禁止应用内付费购买

1002	无法获取产品信息
1003	ProductID为空, 请检查后台配置
1004	下单失败
1005	验单请求失败
1006	用户凭证失效
1007	空SKProduct实体
1100	产生除非消耗、自动续期Product的重复购买
1101	恢复购买错误
1102	无需恢复购买
1103	账单刷新错误
1104	账单校验失败
1105	用户取消或超时
1106	设备被Apple禁止充值
1107	请求变量、签名发生错误
1108	促销信息有误
1109	商品不可用, 检查苹果后台
1110	未同意隐私Apple的协议
1111	需要等待批准
1112	Apple返回未知错误
1113	读取Receipt错误
1114	Apple云服务错误
1115	不支持的平台
2001	防沉迷限制充值
3001	网络错误
4001	未知错误

14、从SDK发起内购（废弃）

```
+ (void)purchaseWithProductID:(NSString *)productID mark:(NSString *)mark roleInfo:
(OMRoleInfoModel *_Nullable)roleInfo purchasingStatu:(void (^)(OMPurchasingStatu
statu))purchasingStatu success:(void (^)(OMPurchaseResultInfo *info))success userCancel:
(void (^)(OMPaymentError *error))userCancel fail:(void (^)(OMPaymentError *error))fail
```

获取IAP商品。

支持iOS12+。

使用IAP需在工程->Target->Frameworks, Libraries, and Embedded Content中引入StoreKit.framework。

success 商品获取成功回调

fail 失败回调

15、前往第三方平台

```
+ (void)leadToPlatformWithKey:(NSString *_Nonnull)key success:(void (^)(void))success fail:
(void (^)(NSInteger, NSString *_Nonnull))fail
```

根据后台配置的key跳转指定第三方平台。

需在Info.plist内，配置需要跳转的平台的scheme到LSApplicationQueriesSchemes下，详细可参考接入步骤（9）

例如 跳转微博，在后台配置key为 `weibo`，url配置为 `sinaweibo://userinfo?uid=4399`

- Info.plist的LSApplicationQueriesSchemes项内需要包含 `sinaweibo`
- 在需要跳转此url的位置使用 `weibo` 作为key调用此API

key 对应社交平台的key。

success 成功回调。

fail 失败回调，返回失败code和相关信息。

16、打开评分弹窗

```
+ (void)reviewRequestWithCommentStatuCallback:(void (^)(NSInteger code, NSString *_Nonnull
msg, NSInteger level, NSString *_Nonnull comment))callback
```

打开评分弹窗

仅限用户登录后使用本接口。

测试说明：<https://note.4399doc.com/prd?sid=ffbb5d38453cbc90f54daf97#/share>

callback

参数名	类型	默认值	说明
code	NSInteger		1000, 未评论; 1001 已在弹窗内评论, 1002 已跳转苹果评论
msg	NSString		已评价; 未评价
level	NSInteger		1~5
comment	NSString		评论内容, 仅返回SDK弹窗评价内容

17、角色创建上报

```
+ (void)reportCreateWithRoleInfo:(OMRoleInfoModel *)roleInfo
                                completion:(void(^)(BOOL isSuccess))completion;
```

在完成角色创建后调用, 用于上报新角色信息。

仅限用户登录后使用本接口。

参数说明

参数名	类型	说明
roleInfo	OMRoleInfoModel	角色信息模型, 见下文 OMRoleInfoModel
completion	void(^)(BOOL isSuccess)	完成回调, <code>isSuccess=YES</code> 表示上报成功

OMRoleInfoModel参数说明

字段	类型	说明
roleID	NSString	角色唯一标识, 可为nil
roleLevel	NSString	角色等级, 建议传当前可见等级, 如“1”、“15”等, 可为nil
serverID	NSString	区服标识, 可为nil
extensionValues	NSDictionary<NSString *, NSString *>	扩展字段, 若与上述字段同名将覆盖原有字段, 可为空

- 使用 `-initWithRoleID:roleLevel:serverID:extensionValues:` 进行初始化, 允许传入 `nil` 的字段 (SDK会做空值处理)。

18、进入游戏上报

```
+ (void)reportGameEntryWithRoleInfo:(OMRoleInfoModel *)roleInfo
                                completion:(void(^)(BOOL isSuccess))completion;
```

在进入游戏 (进入主城/登录成功进入服务器) 时调用, 用于上报当前角色进入游戏行为。

仅限用户登录后使用本接口。

参数说明

参数名	类型	说明
roleInfo	OMRoleInfoModel	角色信息模型，见下文 OMRoleInfoModel
completion	void(^)(BOOL isSuccess)	完成回调， <code>isSuccess=YES</code> 表示上报成功

19、上报事件

```
+ (void)reportEventWithEventName:(NSString *)eventName
        attribute:(NSDictionary<NSString *, NSString *> *)attribute
        completion:(void(^)(BOOL isSuccess))completion;
```

用于上报自定义事件，便于后续统计与分析。

参数说明

参数名	类型	说明
eventName	NSString	事件名，例如： <code>"tutorial_finish"</code> 、 <code>"level_up"</code>
attribute	NSDictionary<NSString *, NSString *> *	事件属性键值对，键与值均为字符串（可为空或传 <code>@{}</code> ）
completion	void(^)(BOOL isSuccess)	完成回调， <code>isSuccess=YES</code> 表示上报成功

三、服务端接口说明

用户充值成功通知（如有充值，且不使用13接口的必须接入）

请访问 <https://payov.5054399.com/doc/index.html>

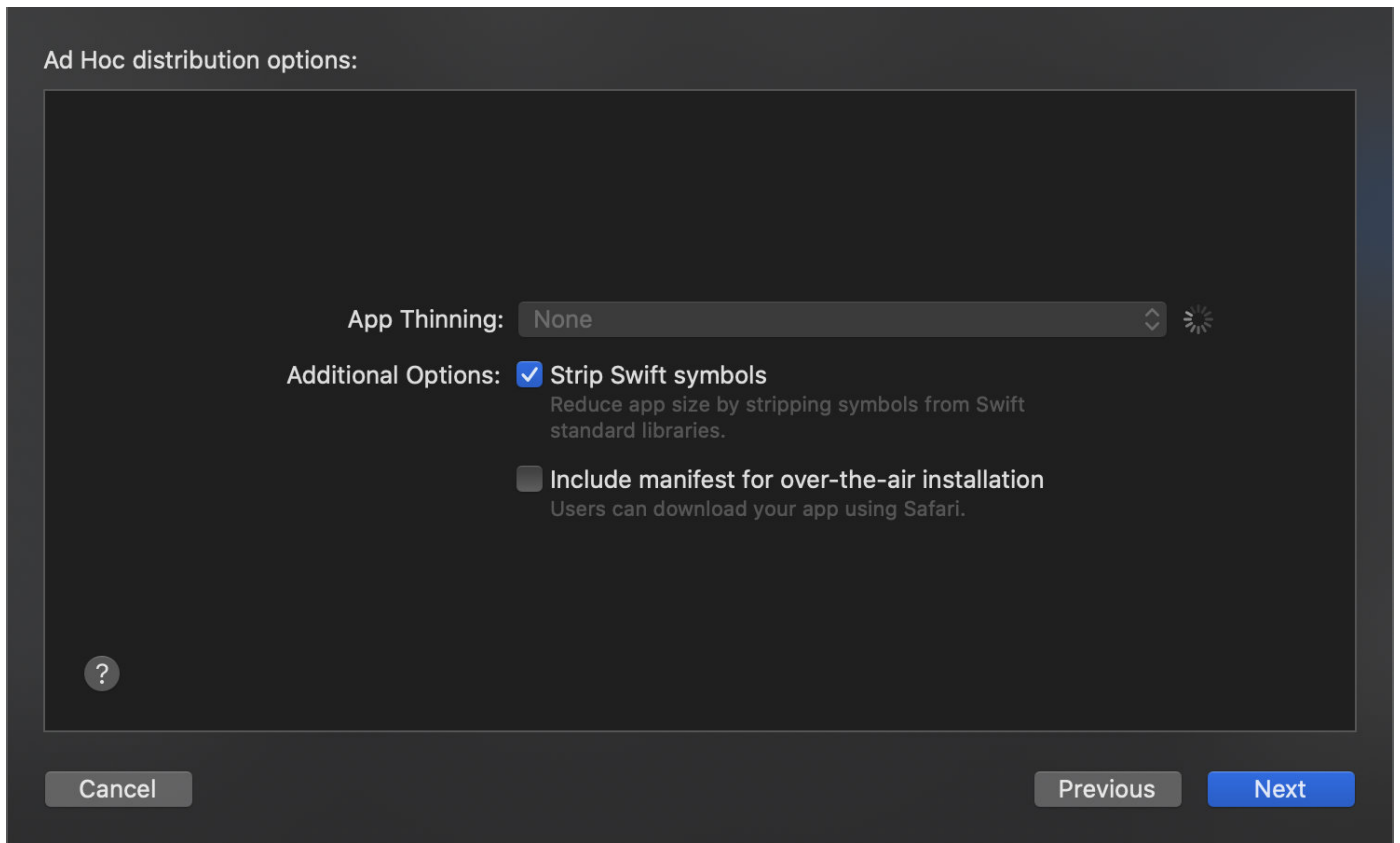
用户信息校验（可选接入）

请访问 <https://open.4399.cn/console/doc#/mobilegame/online/server>

四、Q&A

打包IPA processing failed错误提示

打包发布时，若出现**IPA processing failed**提示，可以尝试回退到上一步，再重试，加载到打包流程第一个界面进行App Thinning设置时，无需等待，快速点击Next进入下一个设置界面即可顺利进入后续流程。（该现象和Xcode打包分析流程有关，如App Thinning界面原先无特殊设置，将不影响发包）



未显示手机号一键登录

- 确认SIM卡已正确插入，蜂窝网络已打开
- 确认运行系统 $\geq$ iOS 9.0
- 和4399运营同学确认，一键登录功能在当前使用的运营商已激活并成功配置
- 确保在应用启动时在 `AppDelegate` 的 `- application: didFinishLaunchingWithOptions:` 中调用了初始化接口。
- 尝试开启手机飞行模式后关闭，再重启应用观察。
- 若日志提示『初始化失败，请核对chientId是否正确。』请和运营同学核对一键登录后台登记游戏包名和实际运行游戏包名知否一致。
- 包名不一致(后台配置的包名和运行的包名不同)

未显示Apple ID登录渠道

- 确认运行系统 $\geq$ iOS 13.0
- 设备当前已登录iCloud账户
- 未配置 Sign In with Apple 能力

未显示QQ登录渠道

- 请检查接入步骤(5)，是否正确配置白名单
- 请检查是否已安装腾讯QQ客户端

QQ登录未跳转回游戏

- 确认接入步骤中**Url Schemes**正确配置
- 确认**URL**回执

未显示第三方登录渠道

- 若登录窗口有关闭按钮，依旧未显示第三方登录渠道，请联系运营同学，确认后台登录渠道是否开启。

iOS 12.2以下的系统启动即闪退

- 检查 `Always Embed Swift Standard Libraries` 是否为 `Yes`。

上架 Appstore 因 Guideline 5.1.1 被拒绝

- 可能为无账号注销功能。请在游戏内增加个人中心入口并接入，此功能包含注销账号功能。API参考 [6、展示个人中心页面](#)
- 详细参考 Apple支持：在您的 App 中提供帐户删除选项
<https://developer.apple.com/cn/support/offering-account-deletion-in-your-app/>

切换Xcode 15打包后，上传TestFlight后出现启动即闪退或显示与此设备不兼容

- 在 `target -> Build Setting -> Other Linker Flags` 增加 `-ld_classic`项并重新打包。

如何接入IAP

- 如果游戏自行接入IAP：请接入二 -> 11接口进行充值前检查，充值完成后，使用三 -> 用户充值成功通知上报充值订单。
- 如果使用SDK的IAP能力：使用 二 -> 12,13接口。
- 充值内购配置说明：<https://note.4399doc.com/prd?sid=af3512fd4bd4abe85e496ba3#/share>

四、更新日志

版本	更新说明	备注
1.9.8	更新一键登录SDK，优化游戏评论	请替换m4399QuickLogin.framework或m4399QuickLogin.xcframework为MTDQuickLogin.xcframework
1.9.9	更新一键登录SDK	务必更新EAccountOpenPageResource.bundle
1.10.0	增加上报角色创建、进入游戏接口，优化内购稳定性	
1.11.0	优化内购稳定性	建议使用新内购接口