

目录

1. 文档说明	2
1.1. 概述.....	2
1.2. 阅读对象	2
2. 总体机制说明	2
2.1. 关于通知CP服务器加钻.....	2
2.2. 参数和开发证书.....	2
2.3. 应用打包	2
3. 接入流程	3
3.1. SDK环境搭建.....	3
3.2. 工程设置	3
3.3. IOS10下因为相机、相册权限闪退问题	4
3.4. 功能编码接入.....	5
3.4.1.初始化【客户端调用】（必接）	5
3.5. 登录方法接入（有登录注册页面）	6
3.6. 注册方法接入.....	6
3.7. 支付方法的接入（必接）	7
4. 注意事项	8

注意：SDK支持iOS8.0(含)及以上版本

1. 文档说明

1.1. 概述

本文档是《iOS抓娃娃SDK接口文档V10.1》附录文档，它详细说明登陆、支付等在技术接入与使用过程中需要注意的地方，以帮助商户避免风险产生。阅读后如有疑问，请联系SDK相关技术支持。

1.2. 阅读对象

供开发人员与商户平台服务方技术或业务人员参考和查询。

2. 总体机制说明

2.1. 关于通知CP服务器加钻

需要CP方提供服务器接口，我们sdk会把extend和订单信息等，在用户支付成功了以后通知该接口。CP方服务器收到通知以后客户端即可给玩家加钻。

2.2. 参数和开发证书

SDK接入所需的参数由抓娃娃团队提供，开发者证书也由抓娃娃团队配置好统一发送给CP方。

2.3. 应用打包

证书p12文件的密码统一为123456，dev描述文件用于应用本地调试，hoc描述文件用于打测试包，测试包抓娃娃这边测试通过后，会要求打提审包，提审包用appstore描述文件打包。

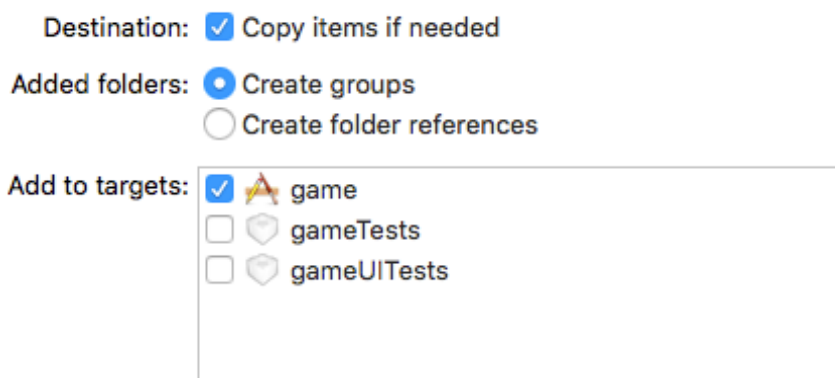
3. 接入流程

拿到sdk以后抓娃娃团队对接人会将SDK接入所需的相关参数以文档的形式发送给你们。

3.1. SDK环境搭建

1.工程文件导入

将DEMO中的Lib文件夹导入应用工程

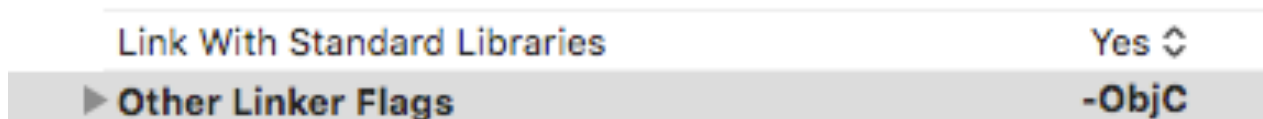


2.添加工程依赖库

在Build Phases->Link Binary With Libraries下添加如下依赖框架AVFoundation.framework、Security.framework、libstdc++.6.tbd、libsqlite3.0.tbd、libz.1.2.5.tbd、CoreMotion.framework、CoreTelephony.framework、SystemConfiguration.framework、libc++.tbd、CFNetwork.framework、libz.tbd、AdSupport.framework、StoreKit.framework。

3.2. 工程设置

1.在工程的Build Setting中找到Other Linker Flags中添加-ObjC宏



2.由于iOS9改用更安全的https，为了能够在iOS9中正常使用SDK，请在“Info.plist”中进行如下配置，否则影响SDK的使用。(右键info.plist用source code打开)

```

<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>

```

3.在iOS9/10下就需要增加一个可跳转的白名单，指定对应跳转App的URL Scheme，否则将在第三方平台判断是否跳转时用到的canOpenURL时返回NO，进而只进行webview授权或授权/分享失败，需要在info.plist中加入安全可跳转的白名单(右键info.plist用source code打开)

```

<key>LSApplicationQueriesSchemes</key>
<array>
  <string>weixin</string>
  <string>wechat</string>
  <string>alipays</string>
  <string>alipayauth</string>
  <string>alipay</string>
  <string>alipayqr</string>
  <string>alipays</string>
  <string>sinaweibo</string>
  <string>sinaweibosso</string>
  <string>mqqapi</string>
  <string>mqq</string>
</array>

```

3.3. iOS10下因为相机、相册权限闪退问题

在iOS10，需要在info.plist中配置关于使用用户相册和相机的说明，所以需要添加NSCameraUsageDescription、NSPhotoLibraryUsageDescription权限

Privacy - Camera Usage Description	String	App需要您的同意,才能访问相机
Privacy - Photo Library Usage Des...	String	App需要您的同意,才能访问相册

Xcode9需要打包需要另外新增NSPhotoLibraryAddUsageDescription权限

Privacy - Photo Library Additions Usa...	String	App需要您的同意,才能保存图片到相册
--	--------	---------------------

3.4. 功能编码接入

3.4.1.初始化【客户端调用】（必接）

执行全局设置，初始化SDK，在程序入口初始化SDK对象，[[YXProApi YXPro_sharedInstance] YXPro_setYXProApiWithInfo:completion:]获取设备标识符，此处参数为应用信息(应用ID、应用名称、应用APPid、推广ID、推广名称、是否测试等参数)

```
//初始化SDK,传入游戏信息
GameAppInfo *game = [[GameAppInfo alloc] init];
game.gameId = @"32";//应用ID
game.gameName = @"test";//应用名称
game.gameAppId = @"6AE25A7FACBE8BA76";//应用Appid
game.promoteId = @"14";//推广ID
game.promoteAccount = @"QWtest1";//推广名称
game.is_test = @"1";//是否测试, 0 正式(提审)版 | 1 测试版

[[YXProApi YXPro_sharedInstance] YXPro_setYXProApiWithInfo:game completion:^(NSDictionary *resultDic) {
    NSLog(@"[init] resultDic:%@",resultDic);
    NSNumber *statusCode = [resultDic objectForKey:@"statusCode"];
    if ([statusCode intValue] == YXProINITSuccess) {
        //初始化成功
    }
}];
```

2) 初始化回调码说明

```
//初始化
typedef enum {
    YXProINITSuccess                = 2000,    //初始化成功
    YXProINITServerError            = 6001,    //服务器正在维护
    YXProINITNetError               = 7,       //网络异常

    YXProINITOtherError            = 0,       //其他错误
} YXPro_INIT_ERROR;
```

3.5. 登录方法接入（有登录注册页面）

1) 因为初始化时传入了应用相关信息，此处可以直接调用登录方法。

```
[[YXProApi YXPro_sharedInstance] YXPro_addLoginView:self.view completion:^(NSDictionary *resultDic) {
    NSLog(@"[showLogin] resultDic = %@", resultDic);

    NSNumber *loginresult = [resultDic objectForKey:@"loginresult"];
    if([loginresult intValue] == YXProLOGINSuccess){
        NSLog(@"登录成功");
        //-----如需二次验证可向服务端验证uid和token-----
        NSString *SDKuid = [resultDic objectForKey:@"userId"];
        NSString *SDKtoken = [resultDic objectForKey:@"token"];
    }
}];
```

2) 登录回调码说明

```
//登录
typedef enum {
    YXProLOGINFail                = 0,    //登录失败
    YXProLOGINSuccess             = 1,    //登录成功
    YXProLOGINServerError         = 2,    //服务器正在维护
    YXProLOGINRETURN              = 3,    //返回游戏
    YXProLOGINNetError            = 7,    //网络异常

    YXProLOGINOtherError          = 0,    //其他错误
} YXPro_Login_ERROR;
```

3.6. 注册方法接入

注册回调结果和登录回调的保持一致，注册完成后自动登录，所以需要在注册回调里面处理登录逻辑。

```
[[YXProApi YXPro_sharedInstance] YXPro_addRegisterView:self.view completion:^(NSDictionary *resultDic) {
    NSLog(@"[showLogin] resultDic = %@", resultDic);
    NSNumber *loginresult = [resultDic objectForKey:@"loginresult"];
    if([loginresult intValue] == YXProLOGINSuccess){
        NSLog(@"登录成功");
        NSString *uid = [resultDic objectForKey:@"userId"];
        NSString *token = [resultDic objectForKey:@"token"];
    }
}];
```

3.7. 支付方法的接入（必接）

注：CP方下发道具应以服务器异步通知为准，SDK客户端同步通知仅做参考

1) 发起支付

注：为了减少漏单几率，不要在这个回调里面处理加钻逻辑，支付成功我们服务器会通知cp服务器，cp服务器收到通知以后客户端再进行加钻。

```
OrderInfo *orderInfo = [[OrderInfo alloc] init];
orderInfo.goodsName = @"金币";
orderInfo.goodsPrice = 1*100;//单位为分
orderInfo.goodsDesc = @"有了金币就可以买买买了";//商品描述
orderInfo.extendInfo = @"1234567890";//此字段会透传到游戏服务器，可拼接订单信息和其它信息等
orderInfo.productId = @"com.anheidiguo.dark.6";//虚拟商品在APP Store中的ID
//-----注意：此处需要传入的是区服的名称，而不是区服编号-----
orderInfo.player_server = @"素月流天";//玩家所在区服名称（跟游戏内显示的区服保持一致）
orderInfo.player_role = @"小明";// 玩家角色名称
orderInfo.cp_trade_no = @"201603101021";//CP订单号

[[YXProApi YXPro_sharedInstance] YXPro_pay:orderInfo completionBlock:^(NSDictionary *resultDic) {
    NSLog(@"[pay] resultDic = %@", resultDic);
    NSNumber *payresult = [resultDic objectForKey:@"payresult"];
    if ([payresult intValue] == YXProTreatedOrderSuccess) {
        NSLog(@"支付成功");
        //*****为了减少漏单几率，不要在这个回调里面处理加钻逻辑，支付成功我们服务*****
        //*****器会通知cp服务器，cp服务器收到通知以后客户端再进行加钻。*****
    }
}];
```

2) 掉单结果处理回调

注：建议在-(BOOL)application:(UIApplication*)application didFinishLaunchingWithOptions:(NSDictionary*)launchOptions SDK 中调用，放在SDK初始化之后，方便启动后及时进行掉单处理

```
//检查漏单的处理回调
[YXProApi YXPro_sharedInstance].YXPro_paySuccess = ^(NSDictionary *resultDic){
    NSLog(@"NSDictionaryResultDic = %@", resultDic);

    NSNumber *payresult = [resultDic objectForKey:@"payresult"];
    if ([payresult intValue] == YXProTreatedOrderSuccess) {
        NSLog(@"支付成功，漏单了");
        //*****为了减少漏单几率，不要在这个回调里面处理加钻逻辑，支付成功我们服务*****
        //*****器会通知cp服务器，cp服务器收到通知以后客户端再进行加钻。*****
    }
};
```

3) 支付回调码说明

```
//支付
typedef enum {
    YXProTreatedOrderSuccess      = 1,      //支付成功
    YXProTreatedOrderFail         = 2,      //支付失败
    YXProTreatedOrderPaying       = 3,      //正在支付中
    YXProTreatedOrderCancel       = 4,      //取消支付
    YXProTreatedOrderCloseMenu    = 5,      //关闭支付界面
    YXProTreatedOrderNoModelist   = 6,      //服务端支付信息列表为空
    YXProTreatedOrderNetError     = 7,      //网络异常

    YXProTreatedOrderInvalidProduct = 11,   //内购，苹果后台没有对应的产品ID或产品ID无效
    YXProTreatedOrderPriceError    = 12,   //内购，订单对应的金额和苹果后台配置的金额不一致
    YXProTreatedOrderReceiptFail   = 13,   //内购，凭证验证失败或者没有通知到游戏服务器
    YXProTreatedOrderUnKnow        = 14,   //支付结果未知
    YXProTreatedOrderOtherError    = 0,     //其他错误
} YXPro_TreatedOrder_ERROR;
```

4. 注意事项

- (1) 支付的时候商品金额的单位都是分，最小的单位是1分，不能有小数出现的