

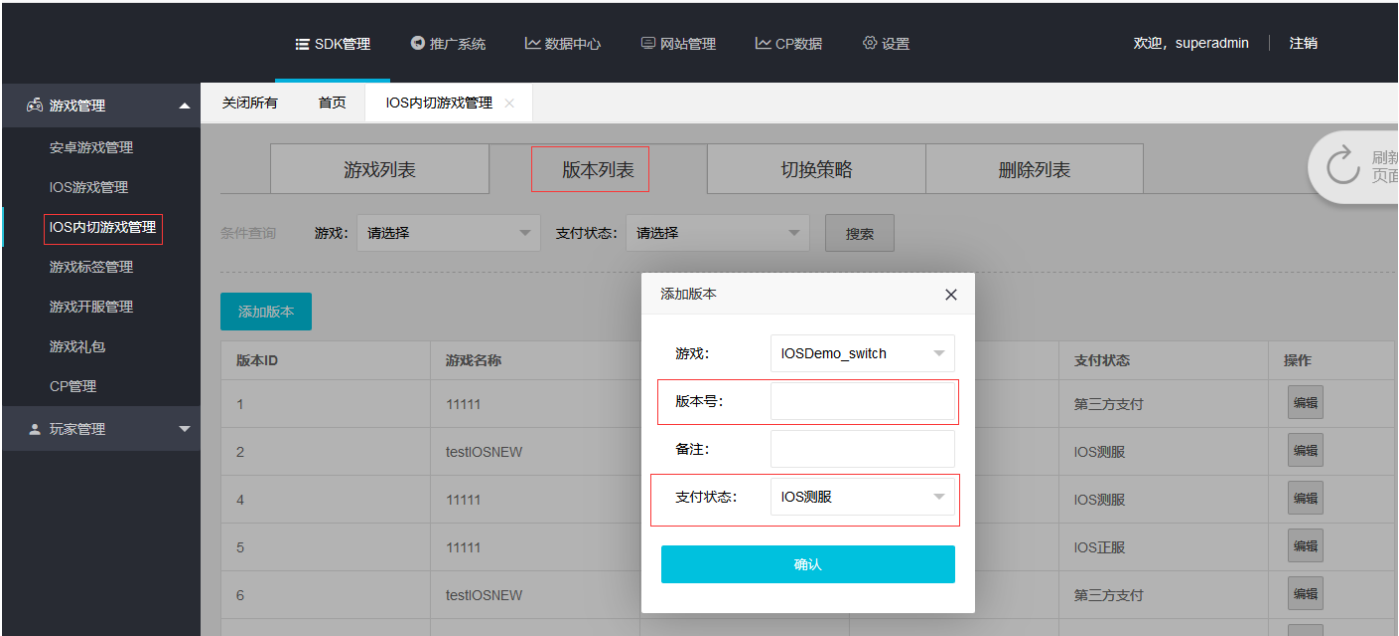
创富天地 SDK Object C 接口说明文档

目录

一、接入前的准备	2
二、SDK 接入说明	3
1、lib 和资源文件	3
2、接口头文件	4
3、添加 SDK 所依赖的系统库	4
4、修改工程 Deployment Target	4
5、工程 Other Linker Flags 设置	4
6、openurl 打开其他应用（ios 正版可以不接）	4
7、第三方登录参数填写（ios 正版可以不接）	5
8、sdk 参数的配置	5
9、URL Types 参数的配置（ios 正版可以不接）	6
三、SDK 接口说明	6
1、第三方跳转接口	6
2、初始化接口	6
3、登录接口	6
3、登陆验证（客户端处理）（ios 正版可以不接）	7
4、登出接口	7
5、充值接口	7
6、支付接口	8
7、游戏开始接口	8
8、创建角色接口	8
9、角色升级接口	9
10、实名注册接口	9
11、防沉迷查询接口	9
12、激活礼包码接口	9
13、登录回调	10
14、支付结果回调	10
15、防沉迷回调	10
16、礼包激活码回调	10
四、AppDelegate 的设置	11
五、服务器接入	11
1、登陆验证（ios 正版可以不接）	11
2、支付结果通知—回调应用服务器接口	12

一、接入前的准备

- 1、在创富后台创建应用，获取参数：Appld、AppKey、AppSecret
 - 2、在创富后台创建内切应用，获取参数：Appld、AppKey、AppSecret
- 打开 IOS 内切游戏管理中的版本列表，进行版本配置。



版本号需要与 XCODE 工程中的 info.plist 中的 `com.sfchannel.versioncode` 配置一致。

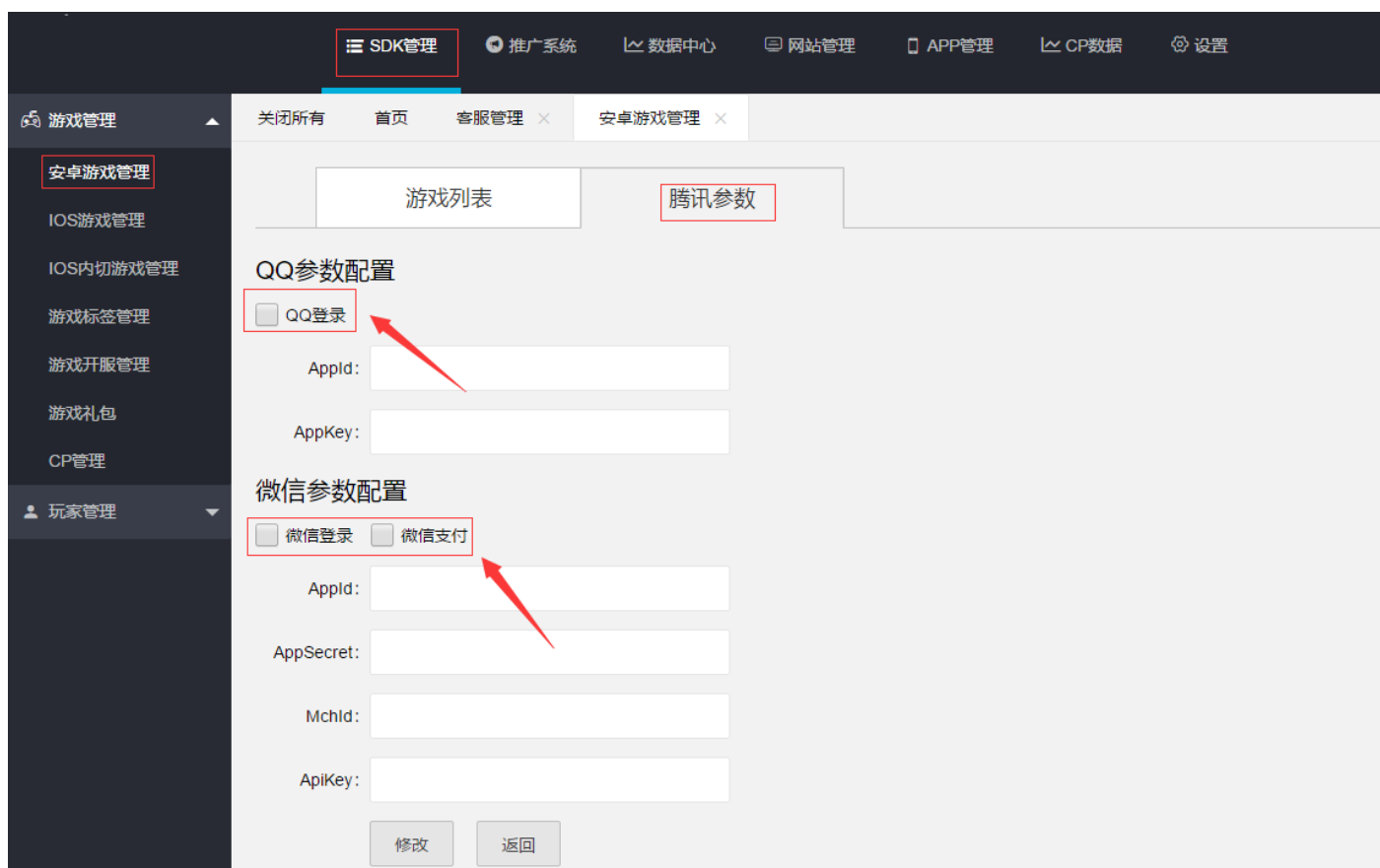
General				Capabilities				Resource Tags				Info				Build Settings				Build Phases				Build Rules			
Key				Type				Value																			
Bundle name				String				\$(PRODUCT_NAME)																			
appid				String				2																			
com.sfchannel.versioncode				String				1																			

注意事项：

测试苹果支付需要注意如下三点：

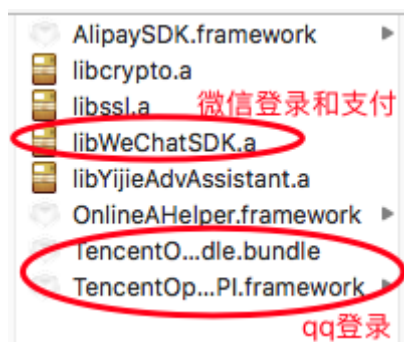
- 1) 参数配置正确。
- 2) versioncode 配置正确。
- 3) Bundle identifier，计费点，证书，与苹果后台配置一致。
- 4) 手机登陆 app store 的苹果账号必须开通测试权限。
- 5) 必须从 IOS 测服（苹果沙箱环境）开始测试。

3、如果需要第三方登陆：QQ 登陆和微信登陆，请在腾讯后台申请 QQ 和微信参数，在管理后台配置 QQ 和微信参数，操作如下图：



二、SDK 接入说明

1、lib 和资源文件



iOS 正版无需使用 AlipaySDK.framework, libWeChatSDK.a, TencentOpenAPI.framework

2、接口头文件

header\YJSDKHelper.h

3、添加 SDK 所依赖的系统库

libstdc++.dylib;libz.dylib;libiconv.dylib;libresolv.dylib;libstdc++.dylib;libsqlite3.dylib;AdSupport.framework;JavaScriptCore.framework;OpenAL.framework;CoreMotion.framework

4、修改工程 Deployment Target

如果接入 QQ 登录，请将 Deployment Target 设置为 10.3 以上； 方法：程序 Target->General->Deployment Info->Deployment Target 下。

5、工程 Other Linker Flags 设置

Target->Build Settings->Linking->Other Linker Flags 添加 -ObjC -fobjc-arc

6、openurl 打开其他应用

需要在 info.plist 添加

▼ Information Property List		Dictionary	(1 item)
▼ LSAApplicationQueriesSchemes	↕	Array	(9 items)
Item 0		String	mqq
Item 1		String	mqqOpensdkSSoLogin
Item 2		String	mqqapi
Item 3		String	mqqopensdkapiV3
Item 4		String	mqqopensdkapiV2
Item 5		String	alipay
Item 6		String	alipayauth
Item 7		String	weixin
Item 8		String	wechat

7、第三方登录参数填写（ios 正版可以不接）

如果使用第三方登录请在 info.Plist 中添加，若不使用则忽略此处
若使用第三方登录 Other Linker Flags 中添加 -ObjC -fobjc-arc

（1） 使用 QQ 登录添加一下参数，并填入对应的参数

▼ qqlogin	↕	Dictionary
qqloginAppkey		String
qqloginAppid		String

（2） 使用微信登录添加一下参数，并填入对应的参数

▼ wxlogin	↕	Dictionary
wxsecretkey		String
wxmchid		String
wxapikey		String
wxappid		String

8、sdk 参数的配置

在 info.Plist 中添加在后台申请到的参数 key 的名称为： appSecret 、 appKey 、 appid 。并在对应 Key 中填入 Value 的值 ， 参数类型为 String 。

另外需要注意：

- ① com.sfchannel.appid 对应 Key 填入 Value 的值应与 appid 的 Value 值保持一致 。
- ② com.sfchannel.channelid 对应 Key 填入 Value 的值为后台分包子渠道 ID，若无子渠道默认填 0 。

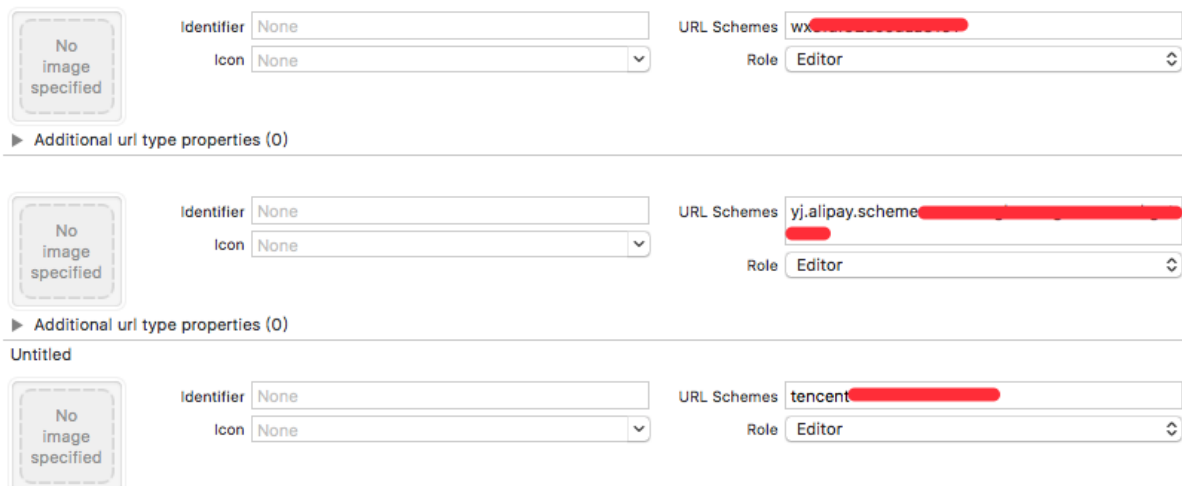
内切相关：

- ③ com.sfchannel.versioncode 对应 Key 填入 Value 的值为后台内切版本号 。

9、URL Types 参数的配置 (ios 正版可以不接)

在 URL Types 中 URL Schemes 中添加参数

如果使用 QQ 或者 微信登录则需要添加申请的 wxappid 、 qqloginappid



Additional url type properties (0)

Additional url type properties (0)

Untitled

其中使用支付宝支付需要添加 **yj.alipay.scheme.+包名**。

三、SDK 接口说明

1、第三方跳转接口

/*第三方跳转接口*/

```
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation;
+(BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url;
```

2、初始化接口

/*初始化接口*/

```
+(void) initSDK(id) initDelegate;
```

/*初始化回调接口*/

```
-(void) onResponse:(NSString*) tag : (NSString*) value ;
```

3、登录接口

/*登录接口*/

```
+(void) login: (id) loginDelegate;
```

具体使用方法请参考 DEMO 。

3、登陆验证（客户端处理）（ios 正版可以不接）

客户端需要将参数上传游戏服务器，游戏服务器与渠道服务器进行登陆验证，具体方法参考[下方文档](#)。

客户端参数获取参考：

参数	获取参数	类型
uin	[user channelUserId]	NSString
token	[user token]	NSString
app	NSDictionary *dict = [[NSBundle mainBundle] infoDictionary]; [dict objectForKey:@"appid"]	NSString
appkey	NSDictionary *dict = [[NSBundle mainBundle] infoDictionary]; [dict objectForKey:@" appKey"]	NSString

4、登出接口

```
/*登出接口*/
+(void) logout;
```

5、充值接口

```
/*充值监听设置*/
+(void) setPayDelegate: (id) payDelegate;
/*充值接口
 * @param payinfo      支付信息
 * NSString* orderId; //订单号
 * NSString* productName; //商品名称
 * int32_t productPrice; //商品价格
 * int32_t productCount; //商品数量
 * NSString* callbackURL; // 回调地址
 * NSString* callbackInfo; // 透传参数
 * NSString* productId; //商品 id （在 iOS 内切支付中 productId 为苹果后台申请的商品 ID）
 * int orderType; // 交易类型 0 普通消费 1 充值货币 2 充值钱包
 */
+(void) charge : (YJSDKHelperPayInfo*)payinfo;
具体使用方法请参考 DEMO 。
```

callbackURL 对应 服务器端的 cbi

callbackURL 可以不传，联系创富后台配置回调地址（推荐）

6、支付接口

/*支付监听设置*/

+(void) setPayDelegate: (id) payDelegate;

/**支付接口

* @param payinfo 定额支付信息

* NSString* orderId; //订单号

* NSString* productName; //商品名称

* int32_t productPrice; //商品价格

* int32_t productCount; //商品数量

* NSString* callbackURL; // 回调地址

* NSString* callbackInfo; // 透传参数

* NSString* productID; //商品 id (在 iOS 内切支付中 product ID 为苹果后台申请的商品 ID)

* int orderType; // 交易类型 0 普通消费 1 充值货币 2 充值钱包

*/

+(void) pay : (YJSDKHelperPayInfo*)payinfo ;

具体使用方法请参考 DEMO 。

callbackURL 对应 服务器端的 cbi

callbackURL 可以不传，联系创富后台配置回调地址（推荐）

7、游戏开始接口

/**游戏开始接口

* @param roleData 角色信息: json 字符串

* roleId: 角色 id

* roleName:角色名称

* roleLevel:角色等级

* zoneId:区服 id

* zoneName:区服名称

*/

+(void) gamestart : (NSString*) roleData ;

具体使用方法请参考 DEMO 。

8、创建角色接口

/**创建角色接口

* @param roleData 角色信息


```
* roleId: 角色 id
* roleName:角色名称
* roleLevel:角色等级
* zoneId:区服 id
* zoneName:区服名称
*/
+ (void) createRole : (NSString*) roleData ;
```

9、角色升级接口

```
/**角色升级接口
 * @param roleData    角色信息
 * roleId: 角色 id
 * roleName:角色名称
 * roleLevel:角色等级
 * zoneId:区服 id
 * zoneName:区服名称
 */
+ (void) levelup : (NSString*) roleData ;
```

10、实名注册接口

```
/*实名注册接口*/
+ (void) realNameReg;
```

11、防沉迷查询接口

```
/*防沉迷监听设置*/
+ (void) setAntiDelegate: (id) antiDelegateDelegate;

/*防沉迷查询接口*/
+ (void) antiAddictionQuery;
```

12、激活礼包码接口

```
/*激活礼包码监听设置*/
+ (void) setGiftcodeDelegate: (id) giftcodeDelegate;
/**激活礼包码接口
 * @param code    礼包码
 */
```

+(void) checkgiftcode:(NSString*)code;

13、登录回调

```
/*登录回调*/
@protocol YJSDKHelperLoginDelegate <NSObject>
@required
-(void) onYJLogout : (NSString *) remain;
-(void) onYJLoginSuccess : (YJSDKHelperUser*) user ;
-(void) onYJLoginFailed : (NSString*) reason;
@end
```

14、支付结果回调

```
/*支付结果回调
 *result: "success": 成功 "failed": 失败
 * msg: 失败原因
 */
@protocol YJSDKHelperPayResultDelegate <NSObject>
@required
-(void) onPayResponse : (NSString*) result: (NSString*) msg;
@end
```

15、防沉迷回调

```
/*防沉迷回调
 *resp: json 字符串: code: 0:成功 其它: 失败码
          msg: 失败原因
 */
@protocol YJSDKHelperAntiDelegate <NSObject>
@required
-(void) onAntiResponse : (NSString*) resp;
@end
具体使用方法请参考 DEMO 。
```

16、礼包激活码回调

```
/*礼包激活码回调
```

```
*resp: json 字符串: result: 0:成功 其它: 失败码
msg: 失败原因
*/
```

```
@protocol YJSDKHelperGiftcodeDelegate <NSObject>
@required
-(void) onGiftcodeResponse : (NSString*) result;
@end
```

四、AppDelegate 的设置

在 AppDelegate 中需要进行如下的设置

引入#import "YJAppPlatform.h" 在对应的 application 中添加如下代码

```
- (BOOL) application:(UIApplication *)application handleOpenURL:(NSURL *)url{

    return [[YJSDKHelper Instance] application:application handleOpenURL:url];

}

-(BOOL) application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication annotation:(id)annotation NS_AVAILABLE_IOS(4_2){

    return [[YJSDKHelper Instance] application:application openURL:url sourceApplication:
        sourceApplication annotation:annotation];

}
```

五、服务器接入

content-type: application/x-www-form-urlencoded

编码 : UTF-8

加密方式: MD5

1、登陆验证 (ios 正版可以不接)

- 接口: http://cfsdk2.cftdcm.com:8080/payserver/account/check_login
- 提交方式: post
- 参数:

app	Integer	应用 id
-----	---------	-------

token	string	token
ts	Long	时间戳
uin	Integer	userid
sign	String	签名

非空参数,除去 sign, 按照传递的参数的字母按照从小到大排序并使用&拼接,拼接完后拼接上生成对应应用的 app_key (游戏 KEY)

请求参数事例

app=xxx&token=xxx&ts=xxx&uin=xxx&sign=xxx

签名事例

```
StringBuilder sb = new StringBuilder();
sb.append("app=").append(req.appId);
sb.append("&token=").append(req.token);
sb.append("&ts=").append(req.timestamp);
sb.append("&uin=").append(req.uin);
String sign = MD5.encrypt(sb.toString() + app.getAppKey());
```

返回码

0	成功
1	失败
2	token 失效
3	签名错误

2、支付结果通知—回调应用服务器接口

- 接口：游戏提供
- 提交方式：post
- 参数

app	Integer	应用 id
at	Long	订单创建时间(系统时间毫秒数)
cbi	String	透传参数
cn	Integer	渠道 Id
fee	Integer	支付金额(分)
kr	Long	随机数

pt	Integer	支付方式（ 2-支付宝网页, 3-微信 web, 4-支付宝 app, 5-微信插件, 6-银联, 7-快捷支付, 102-易接支付宝, 103-易接微信, 10-苹果测服, 11-苹果正服, 16-平台币/钱包余额
res	Integer	支付结果 0 支付成功,其他失败
st	Long	当前服务器（系统时间毫秒数）
tid	Long	SDK 流水
ud	String	Cp 订单号
uid	Integer	userId
ver	Integer	版本号
sign	String	签名

除去 sign，按照传递的参数的字母按照从小到大排序并使用&拼接,拼接完后拼接上生成对应应用的 app_Secret（游戏密钥）

- 返回说明

返回值：SUCCESS-成功，其他均为失败，失败后重复发起回调

回调间隔：0 秒；30 秒；1 分钟；3 分钟；10 分钟；30 分钟；1 小时；2 小时；4 小时；8 小时；12 小时；18 小时；24 小时

签名代码事例

```
Map<String, String> params = new HashMap<String, String>();
params.put("app", notifyReq.app);
params.put("at", String.valueOf(notifyReq.atTime));
params.put("cbi", notifyReq.callbackInfo);
params.put("cn", notifyReq.channelName);
params.put("fee", String.valueOf(notifyReq.fee));
params.put("kr", notifyReq.kr);
params.put("pt", notifyReq.payType);
params.put("res", notifyReq.result);
params.put("st", notifyReq.serverTime);
params.put("tid", notifyReq.tId);
params.put("ud", notifyReq.userData);
params.put("uid", notifyReq.userId);
params.put("ver", notifyReq.version);
```

```
List<String> keys = new ArrayList<String>(params.keySet());
Collections.sort(keys);

String preStr = new String();
for (String key : keys) {
    preStr += String.format("%s=%s&", key, params.get(key));
}
preStr = preStr.substring(0, preStr.length() - 1);
String sign = MD5.encrypt(preStr + appSecret);
```