

iOS SDK开发文档

| 序号 | 版本 | 更新内容 | 作者 | 日期 |
|----|-----|------------|-------|------------|
| 1 | 2.0 | 登录模块、支付模块等 | Candy | 2022/05/11 |

注意：

要玩SDK资源文件均以BGG开头，所以希望用户在接入平台时自己工程的其他资源不要以BGG开头，以免造成资源冲突。

1. SDK构成

当你解压SDK,里面包括

| 文件名 | 说明 |
|------------------|-------|
| BGGSDK.framework | SDK框架 |
| BGGSDK.bundle | 图片资源 |
| BGGConfig.plist | 配置相关 |

2. SDK接入工程环境搭建

添加依赖库

添加SDK依赖的系统库：

StoreKit.framework

NetworkExtension.framework

BGGSDK.framework

添加SDK依赖的dylib：

libc++.tbd

libsqlite3.0.tbd

libz.tbd

添加资源文件：

BGGSDK.bundle

添加配置文件：

BGGConfig.plist

添加第三方库

设置URLScheme

点击工程target,进入Info选项,找到URL Types栏目,点击“+”按钮添加URL type。添加bundleid

Info.plist配置

1、

```
<key>NSLocationAlwaysAndWhenInUseUsageDescription</key>
<string>我们需要通过位置信息进行组队周边匹配等获取相关数据，详情见隐私协议</string>
<key>NSLocationAlwaysUsageDescription</key>
<string>我们需要通过位置信息进行组队周边匹配等获取相关数据，详情见隐私协议</string>
<key>NSLocationWhenInUseUsageDescription</key>
<string>我们需要通过位置信息进行组队周边匹配等获取相关数据，详情见隐私协议</string>
<key>NSPhotoLibraryAddUsageDescription</key>
<string>我们需要访问相册权限用于保存账号的截图信息</string>
<key>NSPhotoLibraryUsageDescription</key>
<string>我们需要访问相册权限用于保存账号的截图信息</string>
<key>NSUserTrackingUsageDescription</key>
<string>此标识符将用于向您推荐个性化广告</string>
```

2、如果你的app基于9.0编译，那么为了适配iOS9.0中的App Transport Security(ATS)对http的限制，在app对应的info.list中添加如下配置：

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

3、为了能够启动微信、QQ客户端，需要在app对应的info.list中添加如下配置：

```
<key>LSApplicationQueriesSchemes</key>
<array>
    <string>mqqopensdkapiV2</string>
    <string>weixin</string>
</array>
```

5、设置bitcode 点击工程target,进入Build Settings选项,搜索bitcode，将Enable Bitcode 设置为No。

更改其 Other Linker Flags 为： *-allload* 或 *-forceload*

3. 开发与测试

1.导入头文件

导入SDK框架的头文件：

```
#import <BGGSdk/BGGSdk.h>
```

2.初始化SDK

```
本地自定义实现初始化回调
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(
name:BGGInitNotify object:nil];

-(void)initClick{
    [[BGGAPIO sharedAPIManager] BGGInit];
}
#pragma mark - ==== 初始化回调 ====
-(void)BGGInitCallback:(NSNotification *)notify{
    if (notify.object == BGGSUCCESS_RESULT) {
        NSLog(@"初始化成功");
    }else{
        NSLog(@"初始化失败");
        [[BGGAPIO sharedAPIManager] BGGInit];
    }
}
```

3.登录接口

本地自定义实现登录回调

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(BGGLoginCallback:) name:BGGLoginNotify object:nil  
[[BGGAPI sharedAPIManeger] BGGLogin];
```

4.创建角色

```
-(void)roleButton{  
    BGGRoleData *roleData = [[BGGRoleData alloc] init];  
    roleData.serverId = @"1";  
    roleData.serverName = @"lsServerName";  
    roleData.roleId = @"12345";  
    roleData.roleName = @"lsls";  
    roleData.roleLevel = 1;  
    roleData.roleBalance = 100;  
    roleData.roleVip = @"3";  
    roleData.dCountry = @"ee";  
    roleData.dParty = @"rr";  
    roleData.roleCreateTime = @"12356984211";  
    roleData.roleLevelUpTime = @"12658945469";  
    roleData.eventType = ROLEEVENT_CREATE_ROLE;  
    roleData.dext = @"";  
    [[BGGAPI sharedAPIManeger] BGGUploadRoleData:roleData];  
}
```

5.充值接口

支付：

本地自定义实现支付回调

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(BGGPMCallback:) name:BGGPMNotify object:nil];

BGGPMDData *PMDData = [[BGGPMDData alloc] init];
PMDData.CPOrderId = [NSString stringWithFormat:@"%123456789%", [self g
PMDData.serverId = @"1";
PMDData.serverName = @"ls";
PMDData.roleId = @"12345";
PMDData.roleName = @"lsls";
PMDData.roleLevel = 1;
PMDData.dext = @"dext";
PMDData.dradio = @"10";
PMDData.dunit = @"元宝";
PMDData.pm = [self.PMTextField.text integerValue];
PMDData.appStoreProductId = @"bingo.10";
PMDData.appStoreProductId = @"com.dzzml.wzjh2.60";
[[BGGAPI sharedAPIManeger] BGGPM:PMDData];
```

6.退出登录接口

本地自定义实现退出登录s回调

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@se
[[BGGAPI sharedAPIManeger] BGGSDKLogout];
```

7.重写AppDelegate的openURL方法,处理外部回调

```

-(BOOL)application:(UIApplication *)application openURL:(NSURL *)url s
ourceApplication:(NSString *)sourceApplication annotation:(id)annotati
on {

    return  [[BGGAPI sharedAPIManeger] handleApplication:application ope

}

// NOTE: 9.0以后使用新API接口
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url options:
(NSDictionary<NSString*, id> *)options
{
    return  [[BGGAPI sharedAPIManeger] handleApplication:app openURL:url
        sourceApplication:[options valueForKey:@"UIApplication
        annotation:[options valueForKey:@"UIApplication
}

```

至此，iOS SDK已经接入完毕，可以进行测试。也可参照SDKDemo工程。