

Gbet体育SDK iOS接入文档

此文档引导宿主App iOS端如何接入SDK

目录

一、环境、兼容性等公共配置

- [1、开发语言](#)
- [2、三方依赖管理](#)
- [3、开发工具](#)
- [4、系统兼容性](#)
- [5、SDK引入](#)
- [6、工程配置](#)
 - [6.1、Deployment Target设置成10.0或更高版本](#)
 - [6.2、Enable Bitcode 设置成NO](#)
- [7、info.plist\(ATS、隐私等\)配置](#)

二、Objective-C接入说明

- [具体接入](#)
 - [1、ICON版方式：启动SDK](#)
 - [2、TAB版方式：配置SDK](#)

三、Swift接入说明

- [接入准备](#)

[1、移除多window相关配置](#)

[2、添加Objective-C桥接](#)

[具体接入](#)

[四、SDK接口说明](#)

[1、配置商户信息、设置回调、启动SDK\(ICON版方式\)](#)

[2、配置SDK信息\(TAB版方式\)](#)

[3、获取SDK主页面](#)

[4、获取投注记录](#)

[5、获取页面ID列表](#)

[6、页面释放](#)

[7、查看SDK版本](#)

[8、视频暂停/播放](#)

[五、动画播放对环境要求](#)

[六、如何获取config配置信息与文档](#)

[七、config配置参数说明](#)

[八、常见问题](#)

[九、版本更新说明](#)

文档

一、环境、兼容性等公共配置

1、开发语言：Objective-C、Swift

2、三方依赖管理：CocoaPods【安装教程[1](#)、[2](#)】

3、开发工具：**Xcode**请使用最新版本

4、系统兼容性：**SDK**仅支持**iOS10+**系统，建议使用iPhone7或更高配手机，**为精简包大小现仅支持真机!!!**

5、SDK引入(**Pods**管理本地代码方式引入!!!)

1、请将iOSSDK文件夹复制到工程文件夹下

2、Podfile添加：

```
pod 'XCSDK_iOS', :path => 'iOSSDK/iOSSDK.podspec'
```

```
pod 'AFNetworking', '~> 4.0.1' # Apple已禁止有UIWebView的IPA上传，SDK中已将版本升到4.0+
```

```
pod 'MBProgressHUD', '~> 1.1.0'
```

```
pod 'SDWebImage'
```

文件结构如下：

```
├── iOSSDKSample
│   ├── iOSSDK
│   │   ├── iOSSDK.podspec
│   │   └── frameworks
│   │       ├── App.framework
│   │       ├── XCSDK.framework
│   │       └── ...
│   ├── iOSSDKSample.xcodeproj
│   ├── iOSSDKSample.xcworkspace
│   ├── Podfile
│   ├── iOSSDKSample
│   │   └── ...
│   └── ...
└── ...
```

3、执行 `pod install` (如果失败对终端翻墙然后`pod update`)

6、工程配置

1、Deployment Target设置成10.0或更高版本

General

Signing & Capabilities

Resource Tags

Info

PROJECT

iOSSDKSample

TARGETS

iOSSDKSample

version

1.0

Build

1

Deployment Info

Target	Device
iOS 9.0	<input checked="" type="checkbox"/> iPhone
	<input checked="" type="checkbox"/> iPad
	<input type="checkbox"/> Mac (requires n

2、Enable Bitcode 设置成NO

PROJECT

iOSSDKSample

TARGETS

iOSSDKSample

Basic

Customized

All

Combined

Levels

Enable Bitcode

▼ Build Options

Setting	iOSSDKSample
Enable Bitcode	No

7、info.plist(ATS与隐私设置等)

```

<key>CFBundleDevelopmentRegion</key>
<string>zh_CN</string>
<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
    <key>NSAllowsArbitraryLoadsInWebContent</key>
    <true/>
    <# 按需添加 异常域名 #>
    <key>NSExceptionDomains</key>
    <dict>
        <key>your domain eg:baidu.com</key>
        <dict>
            <key>NSExceptionAllowsInsecureHTTPLoads</key>
            <true/>
            <key>NSIncludesSubdomains</key>
            <true/>
        </dict>
    </dict>
</dict>
<key>io.flutter.embedded_views_preview</key>
<string>YES</string>

```

二、Objective-C接入说明

具体代码接入

1、ICON版方式：启动SDK(示例代码如下)

```

1、导入头文件
#import <XCSDK/XCSDK.h>
//
2、启动SDK
- (IBAction)launchSDKButton:(UIButton *)sender {
    NSError *error = nil;
    __weak __typeof(self) weakSelf = self;
    NSMutableDictionary *config = @{
        XCSDKMainUrlKey      : self.config[XCSDKMainUrlKey] ?: @"",
        XCSDKImgUrlKey       : self.config[XCSDKImgUrlKey]  ?: @"",
        XCSDKImUrlKey        : self.config[XCSDKImUrlKey]   ?: @"",
        XCSDKTokenKey        : self.config[XCSDKTokenKey]   ?: @"",
        XCSDKAppTypeKey       : self.config[XCSDKAppTypeKey] ?: @"",
    } mutableCopy];
    NSLog(@"config=%@", config);
}

```

```

// 此方法 (openSDKWithConfig:naviVC:error:handler:) 即将废弃
[XCSDKManager openIconSdk:config naviVC:self.navigationController
    typeof(weakSelf) weakSelf = weakSelf;
    NSLog(@"method = %@", params=@"", method, params);
    NSString *showName = nil;
    if ([method isEqualToString:XAppMethodExitSDK]) {
        // @"宿主App-主动退出SDK";
        [self readUserInfo];
        [self.navigationController popViewControllerAnimated:YES
    }
    else if ([method isEqualToString:XAppMethodRetryAlert]) {
        NSUInteger index = [params[@"index"] intValue];
        if (index == 0) {} // 重试
        else if (index == 1) {} // 去检查网络
    }
    else if ([method isEqualToString:XAppMethodNeedLogin]) {
        [self showAnonymousLoginVC:params]; // 匿名登录模式进入, 触
    }
    else if ([method isEqualToString:XAppMethodSystemRepair]) {
        NSUInteger code = [params[@"code"] intValue];
        NSString *msg = params[@"msg"];
        NSString *whTime = params[@"whTime"];
        NSLog(@"code=%zd, msg=%@, whTime=%@", code, msg, whTime)
    }
}];
// 错误处理
if (!!error) {
    NSLog(@"error=%@", error);
    NSString *msg = @"其他错误";
    switch (error.code) {
        case XErrorCode_naviVCError:
        {
            msg = @"根控制器为空或类型错误";
            break;
        }
        case XErrorCode_configError:
        {
            msg = @"配置信息错误或缺失";
            break;
        }
        default:
        {
            //...
            break;
        }
    }
}
}
}

```

2、TAB版方式：配置SDK(示例代码如下)

1、导入头文件

```
#import <XCSDK/XCSDK.h>
//
#define LazyLoadingModel 1
#define LazyLoadingModel2 2
#define TabSDKMode LazyLoadingModel1 // 强烈建议使用懒加载方式2或1接入Ta
bSDK
- (void)viewDidLoad {
    [super viewDidLoad];
#if TabSDKMode == LazyLoadingModel
    // TODO:懒加载方式1: 初始化viewControllers->>需要使用SDK时配置SDK->>初
    // 2.1.1、配置SDK
    BOOL ret = [self configTabSDK]; if(!ret) return;
    // 2.1.2、初始化viewControllers
    [self initViewControllers];
    // 2.1.3、使用的时候再初始化betPage或mainPage
    if (self.selectedIndex == 2) [self configSDKPage:YES]; // 默认选中
#elif TabSDKMode == LazyLoadingModel2
    // TODO:懒加载方式2: 初始化viewControllers->>需要使用SDK时配置SDK->>初
    // 2.2.1、初始化viewControllers
    [self initViewControllers]; self.selectedIndex = 1;
#else
    // TODO:旧方式: 先配置SDK->>初始化viewControllers->>初始化betPage和ma
    /* 如需同时初始化betPage和mainPage, 请先初始化betPage然后再初始化mainPa
    包装示例比如: [[UINavigationController alloc] initWithRootViewController:
    // 1.1、配置SDK
    BOOL ret = [self configTabSDK]; if(!ret) return;
    // 1.2、初始化viewControllers
    [self initViewControllers];
    // 1.3、初始化betPage和mainPage
    self.betNavVC.viewControllers = @[[XCSDKManager betPage]];
    self.mainNavVC.viewControllers = @[[XCSDKManager mainPage]];
#endif
    [self initTabBar];
    self.delegate = self;
    self.view.backgroundColor = UIColor.orangeColor;
}
- (void)tabBarController:(UITabBarController *)tabBarController di
dSelectViewController:(UIViewController *)viewController{
#if TabSDKMode == LazyLoadingModel || TabSDKMode == LazyLoadingMod
e2
    // 2.2.2、配置SDK
    if(!self.didConfigSDK) { BOOL ret = [self configTabSDK]; if(!ret
    // 2.2.3、初始化mainPage或betPage
    if(self.selectedIndex == 2) [self configSDKPage:YES];
    else if(self.selectedIndex == 3) [self configSDKPage:NO];
```

```
#endif  
}
```

三、Swift接入说明

接入准备

1、移除多window相关配置

1.1、AppDelegate.swift文件中AppDelegate

1.1.1: 添加window属性: var window: UIWindow?

1.1.2: 注释或移除多屏代理方法

```
//  
func application(_ application: UIApplication, configurationForConnecting:  
    return UISceneConfiguration(name: "Default Configuration"  
}  
//  
func application(_ application: UIApplication, didDiscardSceneSessions:  
//
```

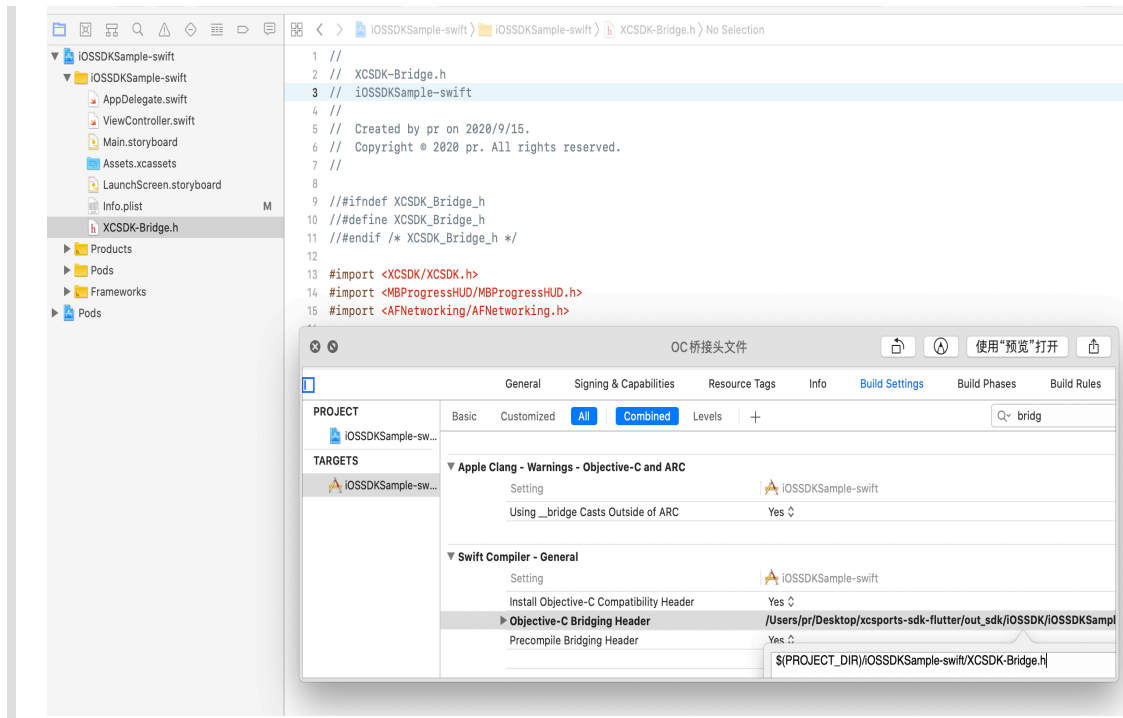
1.2、SceneDelegate.swift文件解除引用

```
//
```

1.3、info.plist移除多屏配置

```
<key>UIApplicationSceneManifest</key>  
<dict>  
    <key>UIApplicationSupportsMultipleScenes</key>  
    <false/> <# ----- 此处确保为false ----- #>  
    <# ----- 删除以下内容 ----- #>  
    <key>UISceneConfigurations</key>  
    <dict>  
        <key>UIWindowSceneSessionRoleApplication</key>  
        <array>  
            <dict>  
                <key>UISceneConfigurationName</key>  
                <string>Default Configuration</string>  
                <key>UISceneDelegateClassName</key>  
                <string>$(PRODUCT_MODULE_NAME).SceneDelegate</string>  
                <key>UISceneStoryboardFile</key>  
                <string>Main</string>  
            </dict>  
        </array>  
    </dict>  
    <# ----- 删除以上内容 ----- #>  
</dict>
```

2、添加Objective-C桥接



具体接入

```
// 获取token
func queryConfig() {
    // 模拟获取登录token
    XCSDKUtils.simulateLogin(XCSDKConfigUAT, config: ["loginName": "admin", "password": "123456"])
    self.data = config ?? [:];
    //NSLog("000-config=\(String(describing: config))")
}

// 启动SDK
@IBAction func launchSDK(_ sender: UIButton) {
    // 组装配置
    let config = [
        XCSDKMainUrlKey: self.data[XCSDKMainUrlKey],
        XCSDKImgUrlKey: self.data[XCSDKImgUrlKey],
        XCSDKImUrlKey: self.data[XCSDKImUrlKey],
        XCSDKTokenKey: self.data[XCSDKTokenKey],
        XCSDKAppTypeKey: self.data[XCSDKAppTypeKey],
    ]
    //NSLog("111-config=\(String(describing: config))")
    // 启动SDK
    XCSDKManager.openSDK(withConfig: config as? [String : String])
    NSLog("method\(method)")
}
}
```

四、SDK接口说明

1、配置商户信息、设置回调、启动SDK(ICON版方式)

```
/// 启动SDK(配置商户信息、设置回调)
/// @param iconSdkConfig SDK相关配置
/// @param naviVC 宿主App导航控制器, SDK导航根控制器(SDK内部仅弱引用)
/// @param configError 配置错误信息
/// @param handler SDK事件回调
+ (void)openIconSdk:(NSDictionary<NSString *, id> *_Nonnull)iconSdkConfig
                  naviVC:(UINavigationController *_Nonnull)naviVC
      configError:(NSError *__autoreleasing *)configError
          handler:(XCSDKHandler)handler;
```

2、配置SDK信息(TAB版方式)

```
/// 配置SDK信息(返回值: NO-配置失败, YES-配置成功)
/// @param sdkConfig SDK配置配置信息
/// @param tabBarVC 宿主App tabBarVC(SDK内部仅弱引用)
/// @param configError 配置错误信息
/// @param handler SDK回调
+ (BOOL)configTabSdk:(NSDictionary<NSString *, id> *_Nonnull)sdkConfig
                  tabBarVC:(UITabBarController *_Nonnull)tabBarVC
      configError:(NSError *__autoreleasing *)configError
          handler:(XCSDKHandler)handler;
```

3、获取SDK主页面

```
/// 单独获取SDK主页面
+ (UIViewController *_Nullable)mainPage;
```

4、获取投注记录页面

```
/// 单独获取SDK投注记录页面
+ (UIViewController *_Nullable)betPage;
```

5、获取页面ID列表

```
/// 获取页面ID列表
+ (NSArray * _Nullable)pageIds;
```

6、页面释放

```
/// 页面释放
/// @param pageIds nil-释放所有页面, 非nil-释放指定页面如: @"mainPage"/
/// @"memberBettingPage"
+ (void)destroyByPageIds:(NSArray<NSString *> * _Nullable)pageIds;
```

7、查看SDK版本

```
+ (NSString * _Nonnull)sdkVersion;
```

9、视频播放/暂停

```
/// 暂停视频
+ (void)pauseVideo;
/// 播放视频
+ (void)startVideo;
```

```
64 - (void)tabBarController:(UITabBarController *)tabBarController didSelectViewController:(UIViewController *)viewController{
65     NSLog(@"Native-%@", self.selectedIndex == 2 ? @"主页" : @"其他页面");
66     if(self.selectedIndex == 2) {
67         [XCSDKManager startVideo]; // 切回主页播放视频
68     }
69     else {
70         [XCSDKManager pauseVideo]; // 切非主页暂停视频
71     }
72
73     #if TabSDKMode == LazyLoadingMode1 || TabSDKMode == LazyLoadingMode2
74         // 2.2.2、配置SDK
75         if(!self.didConfigSDK) { BOOL ret = [self configTabSDK]; if(!ret) return; }
76         // 2.2.3、初始化mainPage或betPage
77         if(self.selectedIndex == 2) [self configSDKPage:YES];
78         else if(self.selectedIndex == 3) [self configSDKPage:NO];
79     #endif
80 }
```

五、动画播放对环境要求

- (1) 支持js
- (2) 支持http 及 https
- (3) 支持websocket
- (4) 当打开关闭遮罩通知功能时, WebView 需拦截 hybrid:协议的请求。否则会报错。若没进行拦截, 请勿填写参数 hybrid 或设 hybrid 为 0。
- (5) Iframe, 当前没使用 iframe, 但为了将来的扩展, 建议在 webview 中支持该特性。
- (6) 是部分 android 机型设置的字体过大会导致版面错乱的情况, android 客户端应设置 webview 的默认字体大小为 100。实现代码: webView.getSettings().setTextZoom(100)

六、如何获取config配置信息与文档

1、在[开放平台](#)申请接入并审核通过后[方获取config配置](#)

2、接入文档/SDK/Demo[下载链接](#)

备注：如文档或SDK下载链接失效请联系客服人员

七、config配置参数说明：

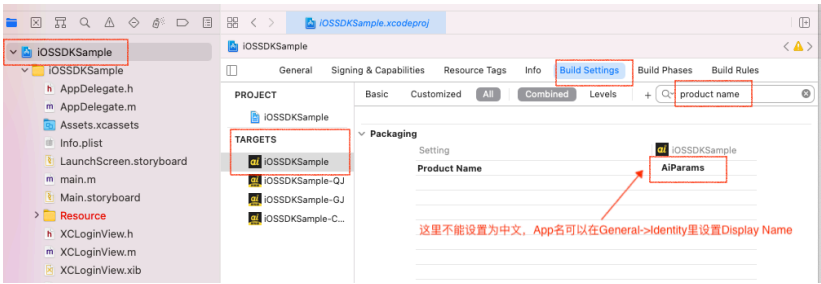
key	说明	是否必传	备注
XCSDKMainUrlKey	基础url	必传	默认值：无
XCSDKImgUrlKey	资源url	必传	默认值：无
XCSDKImUrlKey	push服务url	必传	默认值：无
XCSDKTokenKey	登录token	非必传	默认值：无，不传或传空即为匿名登录
XCSDKTypeKey	SDK类型 (TAB/ICON版)	必传	默认值：iOS-内部已隐式传入无需再传，安卓-需显式传入
XCSDKAppTypeKey	App类型	必传	请传Gbet 默认值：无
XCSDKLaunchTitleKey	SDK启动标题	必传	默认值：不传显示Gbet
XCSDKLaunchSubTitlesKey	SDK启动子标题列表	非必传	默认值：见"XCSDKManager.h"

八、常见问题

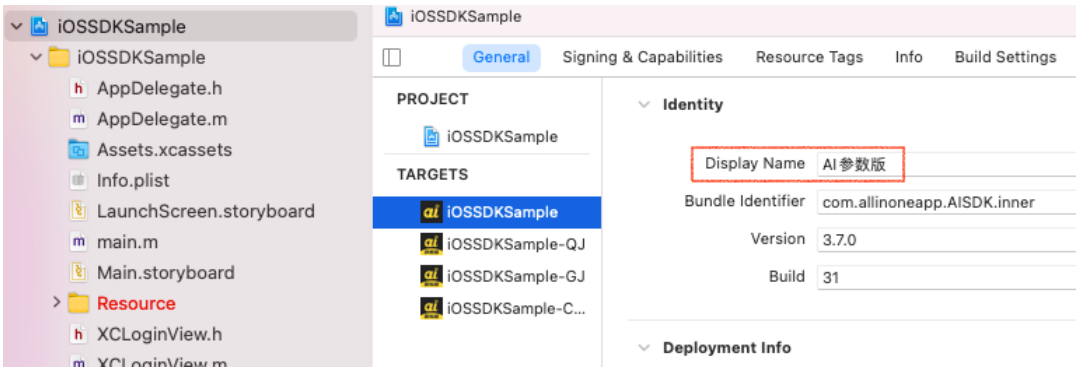
1、iOS14兼容性：因SDK内部使用FlutterSDK的版本是1.22.6，此版本**Product Name不支持中文(非UTF8)**，配置中文(非UTF8)会导致crash，如宿主App需要设

置中文(非UTF8)App名请使用**Display Name**来替代

Product Name配置： **Product Name配置中文(非UTF8)**



Display Name配置



九、版本更新说明

更改日期	版本	更改人	版本说明	审核人
2019-11-01	1.0.0	eric	创建	hank
2019-11-15	1.1.0	eric	1.1需求相关	hank
2019-11-22	1.1.1	eric	修改接入方式、Bug fixed	hank
2019-12-06	1.1.2	eric	新增配置信息、Bug fixed	abraham
2019-12-13	1.1.3	eric	优化接入流程、Bug fixed	abraham
2019-				

12-20	1.2.0	eric	1.2需求相关、Bug fixed	alice/tab/abraham
2019-12-27	1.2.1	eric	1.2需求相关、Bug fixed	isco/abraham
2020-01-03	1.3.0V1	eric	新增1.3版本功能与相关配置 Bug fixed	isco/abraham
2020-01-04	1.3.0V2	eric	新增示例代码	isco/abraham
2020-01-09	1.3.0V3	eric	去掉相关配置，优化流程	isco/abraham
2020-01-16	1.3.4	eric	快捷投注	isco/abraham
2020-03-01	1.5.1V1	eric	1.5.1新需求	isco/abraham
2020-03-09	1.5.1V2	eric	边沿侧滑返回宿主App可配置	isco/abraham
2020-04-03	1.6.4	eric	1.6.4需求	isco/abraham
2020-05-06	2.0.0	eric	2.0.0需求	isco/abraham
2020-05-25	2.1.0	eric	【优化】全新的2.1界面升级，加入了动效，优化了交互细节，等你来细细发现	isco/abraham
2020-05-25	2.2.0V1	eric	2.2.0需求	isco/abraham
2020-06-12	2.2.0V2	eric	SDK新增启动图配置项，优化内存	isco/abraham
2020-06-23	2.2.0V3	eric	SDK减包	isco/abraham
2020-07-03	2.2.1	eric	2.2.1 new feature	isco/abraham
2020-07-17	2.2.2	eric	2.2.2 new feature	isco/abraham

2020-07-24	2.2.3	eric	2.2.3 new feature	isco/abraham
2020-08-27	3.0.0	eric	新增页面方式集成方式	isco/abraham
2020-09-15	2.2.3V2 3.0.0V1	eric	支持Swift	isco/abraham
2020-09-15	3.4.0	eric	1.TabSdk启动方式API变更, 简化接入流程 2.修复已知bug	isco/abraham
2020-11-16	3.5.0	eric	1.新增TAB版token失效回调 2.新增系统维护回调 3.新增视图控制器生命周期方法回调	isco/abraham
2020-12-02	3.6.0	eric	1.优化投注错误提示 2.修改串关投注最低限额为2元 3.修复已知bug	isco/abraham
2021-01-13	3.7.0	eric	1.优化比赛详情页性能 2.优化赔率点水交互效果 3.优化列表盘口开关盘及时性 4.修复已知bug	isco/abraham
2021-02-05	3.8.0	eric	1.修复已知bug, 增强系统稳定性	isco/abraham
2021-03-10	3.8.5	eric	1.优化提前结算功能, 提升结算成功率 2.修复比分遮罩层无法关闭bug 3.修复系统维护时, 页面显示错误问题	isco/abraham
2021-04-09	3.9.0	eric	1.优化视频播放速度, 修复视频偶现变形问题 2.新增注单延期结算标记 3.新增保留上一笔投注注单 4.取消震动效果 5.优化ios内存问题	isco/abraham
2021-04-25	3.10.0	eric	3.10.0需求	ricky/abraham

2021-05-27	3.10.2	eric	3.10.2需求	ricky/abraham
2021-06-09	3.12.0	eric	3.12.0需求	ricky/abraham
2021-06-10	3.12.1	eric	3.12.1需求	ricky/abraham
2021-06-12	3.12.2	eric	3.12.2需求	ricky/abraham
2021-06-13	3.12.3	eric	3.12.3需求	ricky/abraham
2021-06-23	3.12.4	eric	3.12.4需求	ricky/abraham
2021-08-18	3.13.0	eric	3.13.0需求	ricky/abraham
2021-09-07	3.13.1	eric	3.13.1需求	ricky/abraham
2021-10-18	3.14.6	eric	3.14.6需求	ricky/abraham
2021-11-29	3.14.6	eric	添加匿名登陆	ricky/abraham