常见问题归纳 （iOS端）

1.如何启动SDK

1.1、参数如何配置,如何快速启动

SDK启动流程

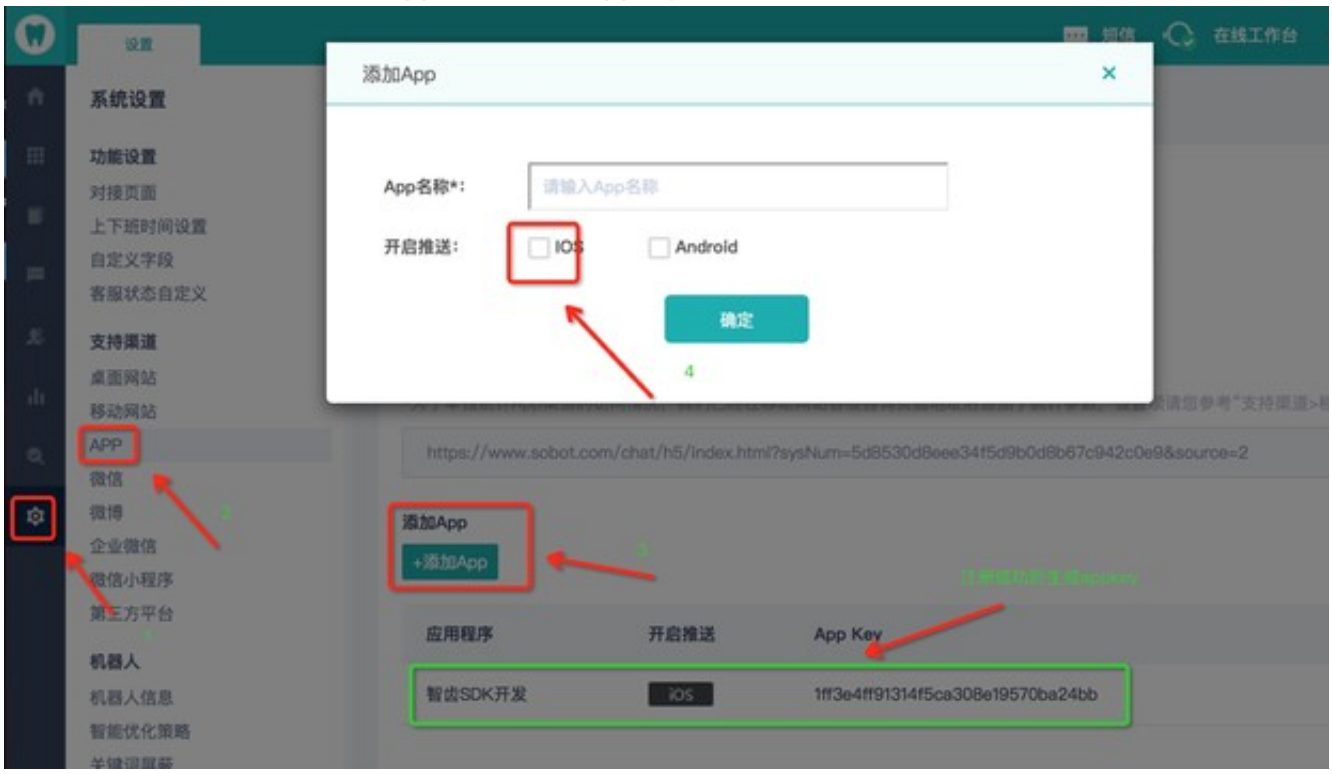
```
/* 1 初始化SDK ,注意要在启动SDK之前调用
*/
 [[ZCLibClient getZCLibClient] initSobotSDK:@"your appkey"];


/* 2 启动SDK
*/
// ZCLibInitInfo 类 配置appkey 接口环境 apiHost 用户ID userId 等参数 详情点击 ZCLibInitI
nfo.h 文件查看
  ZCLibInitInfo *initInfo = [ZCLibClient getZCLibClient].libInitInfo;
  initInfo.userId = @"用户ID";

  // 配置SDK UI相关参数 （自定义字体（可选） 自定义背景、按钮等颜色）
  ZCKitInfo *uiInfo=[ZCKitInfo new];

  [[ZCLibClient getZCLibClient] setLibInitInfo:initInfo];

  [ZCSobot startZCChatVC:uiInfo with:self target:nil pageBlock:^(ZCChatController
*object, ZCPageBlockType type) {
                // 点击返回
                if(type==ZCPageBlockGoBack){
                    NSLog(@"点击了关闭按钮");

                }
                // 页面UI初始化完成，可以获取UIView，自定义UI
                if(type==ZCPageBlockLoadFinish){
                    NSLog(@"页面加载完成");
                }
    } messageLinkClick:nil];
```

1.2、初始化失败怎么办

- 1.检查网络环境是否正常，真机测试时是否有开启VPN相关应用
- 2.检查appkey 是否添加，是否iOS端或者和安卓共同生成的appkey。

如果您还没有注册appkey

1.登录智齿官网 https://www.sobot.com
2.登录账号，在设置页面找到app选项，创建appkey 如图所示



- 3.是否有配置过域名 apiHost，检查当前环境下是否能正常调用接口,具体设置在ZCLibInitInfo.h的apiHost属性（**本地化项目必须设置此项，腾讯云环境https://ten.sobot.com,默认为智齿线上阿里云环境，**）。

- 4.检查初始化方法是否有调用 智齿iOS_SDK 从2.4.3版本开始 初始化SDK需要调用[[ZCLibClient getZCLibClient] initSobotSDK:@"your appkey"];方法

1.3 集成方式

- 直接到官网下载集成，说明文档中会有相关下载地址。

- 使用pod集成

```
// 普通版本
pod 'SobotKit'

// 电商版本
pod 'SobotPlatform'
```

2.如何获得未读消息数

2.1、SDK未读消息数获取方法，代码如下：

```
/*
* 获取未读消息
* obj：当前消息msg内容 unRead：未读消息数  object：当前消息对象ZCLibMessage
*/
[ZCLibClient getZCLibClient].receivedBlock=^(id obj,int unRead,NSDictionary *objec
t){
            NSLog(@"未读消息数量：\n%d,%@",unRead,obj);
        };
```

2.2、不能正确获取SDK未读消息

- 首次进入APP无法获取SDK消息数，必须至少进入一次SDK，并且人工接待成功
- 进入APP重新连接一次智齿服务，代码如下：

```
    [[ZCLibClient getZCLibClient] checkIMConnected];
```

3.自定义消息样式

3.1、自定义商品信息样式

- 1.1商品信息显示样式

- 1.2商品信息发送样式

- 1.3 如果要修改商品信息样式需要使用 SDK UI 源码集成的方式来修改 **智齿 iOS_SDK UI源码 Git下载地址：**https://github.com/ZCSDK/sobotKit_UI_iOS

- 1.4 用户想实现商品信息自定义，发送之后的商品信息样式也和商品信息展示样式一致 。
  `未实现` ，发送的商品信息样式是固定的纯文本类型的消息样式，不能修改消息类型，历史记录中也无法区分当前发送的商品信息类型。

- 1.5发送商品信息中固定文案的修改 可以修改 使用 SDK UI 源码集成的方式来修改 **智齿 iOS_SDK UI源**

码 Git下载地址：https://github.com/ZCSDK/sobotKit_UI_iOS

3.2、自定义链接事件，链接事件点击无效果等问题

```
// 自定义点击
[ZCSobot startZCChatVC:uiInfo with:self target:nil pageBlock:^(ZCChatController *o
bject, ZCPageBlockType type) {

    } messageLinkClick:^(NSString *link) {
        NSLog(@"%@",link);
    }];

// 系统默认跳转
/ **
[ZCSobot startZCChatVC:uiInfo with:self target:nil pageBlock:^(ZCChatController *o
bject, ZCPageBlockType type) {

    } messageLinkClick:nil];
    */
```

- 如果实现了这个messageLinkClick SDK超链点击事件将由直接交给block处理，内部不在处理链接的点击事件，您可以获取 `链接地址link` 自己处理点击链接事件。
- 如何显示和添加电话链接在SDK端调用拨打电话功能

解决方法： 在PC 端添加如下格式的标签样式

```
客服电话:<a href="tel:60605078">60605078</a>
```

3.3、自定义的消息格式

如果修改 SDK现有消息样式 需要使用 SDK UI 源码集成的方式来修改 **智齿 iOS_SDK UI源码 Git下载地址：**https://github.com/ZCSDK/sobotKit_UI_iOS 如果要新增消息类型，无法新增，服务端没有记录新增消息类型，**无法自定义消息类型**

4.资源文件如何修改

4.1、图片资源

SDK资源文件

4.2、国际化资源

智齿iOS*SDK* ***SobotKit.bundle****文件包含* `图片资源` `emoji表情资源文件` `国际化语言文件` *en*lproj(英文) zh-Hans_lproj(中文)

4.3、如何更换自己的图片

- 如果使用源代码集成方式，直接“查看SobotKit.bundle包内容”，替换相关资源即可。
- 如果使用pod方式对接也可以到资源文件中直接修改，每次更新pod会覆盖相关更新，可以指定pod版本或备份一份资源文件每次更新覆盖。

4.4、SDK部分按钮或者所有按钮图片未显示

- 1.检查语言文件是否集成到项目中，或者替换新版之后是否替换了图片资源文件包
- 2.之前有替换过图片资源文件包中的某些按钮的图片，替换版本后请核对图片名称是否一致（图片名大小写）
- 3.pod 集成的用户替换新版后出现上述问题，请清理Xcode缓存和更新本地pod仓库。

5.接入SDK后台挂起时间影响

- 添加智齿SDK后后台挂起时间变短了怎么办?
  由于智齿SDK后台有一个长连接监听消息，会使用到系统相关的网络、屏幕退出、进入相关监听事件，导致退出后台后APP很容易被系统杀死；如果对相关问题敏感的APP，可通过如下代码暂停智齿相关服务，进入APP后再启动即可，消息不会丢失。

详细代码如下： SDK2.6.3之后的版本 处理方法如下：

```
- (void)applicationDidEnterBackground:(UIApplication *)application {
 [[ZCLibClient getZCLibClient] closeIMConnection];
 [[ZCLibClient getZCLibClient] removeIMAllObserver];
}
```

```
- (void)applicationDidBecomeActive:(UIApplication *)application {
    [[ZCLibClient getZCLibClient] checkIMConnected];
     [[ZCLibClient getZCLibClient] checkIMObserverWithRegister];
}
```

SDK2.6.2~SDK2.4.5 的版本 处理方法如下

```
- (void)applicationDidEnterBackground:(UIApplication *)application {
[ZCLibClient closeZCServer:NO];
}
```

```
- (void)applicationDidBecomeActive:(UIApplication *)application {
[[ZCLibClient getZCLibClient] aginitIMChat];
}
```

6.UI适配问题：如何修改背景，如何适配导航，如何适配链接颜色等

6.1、聊天区域

```
/**
 *  自定义风格颜色 主题色，导航颜色
 */
@property (nonatomic,strong) UIColor    *customBannerColor;
/**
 *  相册导航栏的颜色
 *
 */
@property (nonatomic,strong) UIColor   *imagePickerColor;

/**
 *  相册导航栏的标题颜色
 *
 */
@property (nonatomic,strong) UIColor  *imagePickerTitleColor;


/**
 *  左边聊天气泡颜色
 */
@property (nonatomic,strong) UIColor    *leftChatColor;

/**
 *  右边聊天气泡颜色
 */
@property (nonatomic,strong) UIColor    *rightChatColor;

/**
 *  左边聊天气泡复制选中的背景颜色
 */
@property (nonatomic,strong) UIColor    *leftChatSelectedColor;

/**
 *  右边聊天气泡复制选中的背景颜色
 */
@property (nonatomic,strong) UIColor    *rightChatSelectedColor;


/**
```

```
 *  底部bottom的背景颜色
 */
@property (nonatomic,strong) UIColor    *backgroundBottomColor;


/**
 *  底部bottom框边框线颜色(输入框、录音按钮、分割线)
 */
@property (nonatomic,strong) UIColor    *bottomLineColor;
/**
 *  提示气泡的背景颜色
 */
@property (nonatomic,strong) UIColor    *BgTipAirBubblesColor;

/**
 * 语音cell选中的背景色
 *
 */
@property (nonatomic,strong) UIColor    *videoCellBgSelColor;

/**
 *
 *  富文本中的线条颜色
 *
 */
@property (nonatomic,strong) UIColor    *LineRichColor;

/**
 *
 *  通告栏的背景色
 *
 */
@property (nonatomic,strong) UIColor    *notificationTopViewBgColor;

/**
 *
 *  通告栏的文字颜色
 *
 */
@property (nonatomic,strong) UIColor    *notificationTopViewLabelColor;

/**
 *
 *  通告栏的字体设置
 *
 */
@property (nonatomic,strong) UIFont    *notificationTopViewLabelFont;
/**
```

```
 *  顶部文字颜色
 */
@property (nonatomic,strong) UIColor    *topViewTextColor;

/**
 *  左边气泡文字颜色
 */
@property (nonatomic,strong) UIColor    *leftChatTextColor;


/**
 *  右边气泡文字颜色
 */
@property (nonatomic,strong) UIColor    *rightChatTextColor;

/**
 *  时间文字的颜色
 */
@property (nonatomic,strong) UIColor    *timeTextColor;

/**
 *  提示气泡文字颜色
 */
@property (nonatomic,strong) UIColor    *tipLayerTextColor;

/**
 *  客服昵称颜色
 */
@property (nonatomic,strong) UIColor    *serviceNameTextColor;

/**
 *  提示cell中客服昵称的文字颜色
 */
@property (nonatomic,strong) UIColor    *nickNameTextColor;


/**
 *  左边气泡中的链接颜色
 */
@property (nonatomic,strong) UIColor   *chatLeftLinkColor;


/**
 *  右边气泡中的链接颜色
 */
@property(nonatomic,strong) UIColor    *chatRightLinkColor;
```

```
/**
 *  商品详情cell中title的文字颜色
 *
 */
@property (nonatomic, strong) UIColor   *goodsTitleTextColor;

/**
 *  商品详情cell中摘要的文字颜色
 *
 */
@property (nonatomic, strong) UIColor   *goodsDetTextColor;

/**
 *  商品详情cell中标签的文字颜色
 *
 */
@property (nonatomic ,strong) UIColor   *goodsTipTextColor;

/**
 *  商品详情cell中发送的文字颜色
 *
 */
@property (nonatomic, strong) UIColor   *goodsSendTextColor;
/**
 *  是否设置相册背景图片
 */
@property (nonatomic ,assign) BOOL    isSetPhotoLibraryBgImage;


/**
 *  多轮会话模板四的超链颜色
 */
@property (nonatomic,strong) UIColor   *chatLeftMultColor;


/**
 *  多轮会话中 展开和收起的文字颜色
 */
@property (nonatomic,strong) UIColor * openMoreBtnTextColor;

/**
 *
 *  更多按钮默认图片
 *
 **/
@property (nonatomic,copy) NSString * moreBtnNolImg;

/**
```

```
 *
 *  更多按钮选中图片
 *
 **/
@property (nonatomic,copy) NSString * moreBtnSelImg;

/**
 *
 *  转人工按钮默认图片
 *
 **/
@property (nonatomic,copy) NSString * turnBtnNolImg;

/**
 *
 *  转人工按选中图片
 *
 **/
@property (nonatomic,copy) NSString * turnBtnSelImg;

/**
 *
 *  返回按钮默认图片
 *
 **/
@property (nonatomic,copy) NSString * topBackNolImg;

/**
 *
 *  返回按钮选中图片
 *
 **/
@property (nonatomic,copy) NSString * topBackSelImg;

/**
 *
 *  返回按钮的默认背景色
 *
 **/
@property (nonatomic,strong) UIColor * topBackNolColor;

/**
 *
 *  返回按钮的高亮背景色
 *
 **/
@property (nonatomic,strong) UIColor * topBackSelColor;
```

```
/**
 *
 *  导航栏背景色 （单独修改）
 *
 **/
@property (nonatomic,strong) UIColor * topViewBgColor;


/**
 *
 *  顶踩 文字 默认颜色
 *
 **/
@property (nonatomic,strong) UIColor * topBtnNolColor;

/**
 *
 *  顶踩 文字 选中颜色
 *
 **/
@property (nonatomic,strong) UIColor * topBtnSelColor;

/**
 *
 *  顶踩 文字 置灰颜色
 *
 **/
@property (nonatomic,strong) UIColor * topBtnGreyColor;
```

6.2、评价区域

```
/**
 *  暂不评价按钮文字颜色
 */
@property (nonatomic, strong) UIColor   *noSatisfactionTextColor;

/**
 *  评价页面中 已解决 未解决  文字的高亮状态颜色
 *
 */
@property (nonatomic, strong) UIColor   *satisfactionTextSelectedColor;


/**
 *  评价页面中 已解决 未解决 按钮的选中的背景色
 *
 */
@property (nonatomic, strong) UIColor   *satisfactionSelectedBgColor;

/**
 *  提价评价按钮的文字颜色
 */
@property (nonatomic,strong) UIColor    *submitEvaluationColor;

/**
 *  提交评价背景颜色
 */
@property (nonatomic,strong) UIColor    *commentCommitButtonColor;


//    kitInfo.satisfactionSelectedBgColor = [UIColor redColor];
//    kitInfo.satisfactionTextSelectedColor = [UIColor blueColor];
//        kitInfo.submitEvaluationColor = [UIColor blueColor];
//        kitInfo.commentCommitButtonColor = [UIColor redColor];
//     kitInfo.noSatisfactionTextColor = [UIColor blueColor];
```

3、导航栏适配

* 消息列表向下偏移

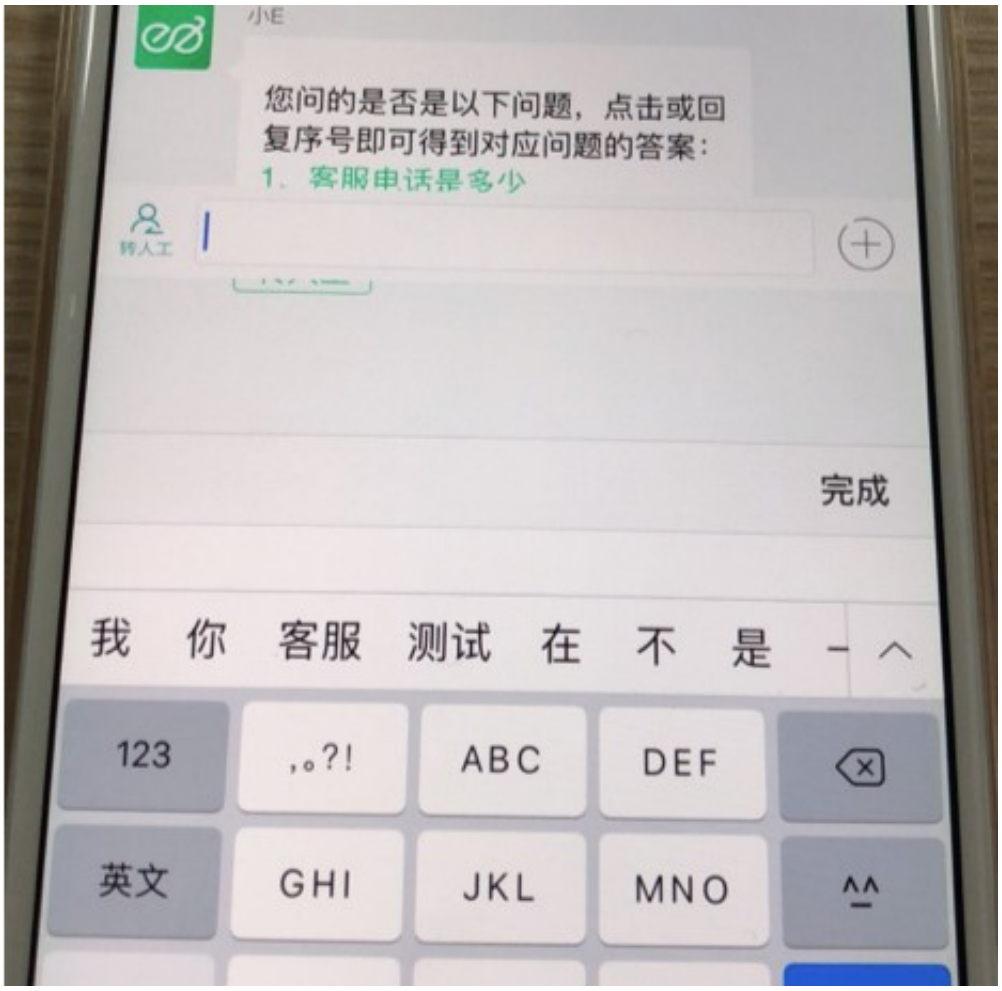

iPhone 8 - 11.3
Carrier
5:10 PM
返回
Hello
非常对不起哦，不知道怎么回答这个问题呢，我会努力学习的~您可以点击左下角转人工客服咨询哦~
16:03

- 出现两种导航栏 

  iOSSDK2.5.0之后版本集成 页面显示异常，导航栏修改方法

详细解决方法（链接里有演示demo和相关问题的处理方法）： https://github.com/ZCSDK/sobot_iOS_Demo

7.如何适配键盘



解决方法：
7.1、用户使用的是2.3.3之前的版本，请切换到2.3.3或者2.4.0的版本，2.3.3开始适配的iOS 11版本

7.2、使用第三方管理库
* 用户有使用第三方的键盘管理库，在启动SDK页面的时候关闭这个库，退出SDK的时候在开启

例如：您的项目如有使用IQKeyboardManager第三方键盘管理库，请在启动SDK的时候关闭，退出SDK页面的时候在开启。

```
IQKeyboardManager关闭方法：
[IQKeyboardManager sharedManager].enableAutoToolbar = NO;[[IQKeyboardManager share
dManager] setEnable:NO];

IQKeyboardManager开启方法：
[IQKeyboardManager sharedManager].enableAutoToolbar = YES;[[IQKeyboardManager shar
edManager] setEnable:YES];
```

8.事件监听

8.1、初始化事件监听

```
   [ZCSobot startZCChatView:uiInfo with:self target:nil pageBlock:^(ZCUIChatContro
ller *object, ZCPageBlockType type) {
       // 页面UI初始化完成，可以获取UIView，自定义UI
        if(type==ZCPageBlockLoadFinish){

        }

        // 点击返回
       if(type==ZCPageBlockGoBack){

        }

    } messageLinkClick:nil];
```

8.2、点击链接监听 * 如果实现了这个messagelinkBlock SDK超链点击事件将由直接交给block处理，内部不在处理链接的点击事件，您可以获取 `链接地址link` 自己处理点击链接事件。

```
[ZCSobot startZCChatVC:uiInfo with:self target:nil pageBlock:^(ZCChatController *o
bject, ZCPageBlockType type) {

    } messageLinkClick:^(NSString *link) {
        NSLog(@"%@",link);
    }];
```

8.3、留言事件监听

自定义留言事件处理
* 1.在启动SDK方法中的 target 传入合适的接收代理对象

```
[ZCSobot startZCChatVC:uiInfo with:self target:self pageBlock:^(ZCChatController *
object, ZCPageBlockType type) {

    } messageLinkClick:nil];
```

- 2.服从协议
- 3.实现方法 -(void)openLeaveMsgClick:(NSString*)tipMsg;

9.权限介绍

智齿SDK需要设置哪些权限以及这些权限作用和使用场景

解决方法：

```
在info.plist添加NSContactsUsageDescription的key，value中添加相应的描述
 <key>NSPhotoLibraryUsageDescription</key>
 <stringSDK会在您生成留言工单以及反馈问题发送拍照图片时需要访问您的相册权限</string>

 <key>NSCameraUsageDescription</key>
 <string>SDK会在您生成留言工单以及反馈问题发送拍照图片时需要访问您的相机权限</string>

 <key>NSMicrophoneUsageDescription</key>
 <string>SDK会在您反馈问题发送语音时需要访问麦克风权限</string>
 <key>NSPototLibraryAddUsageDescription</key>
 <string>SDK会在您聊天页保存聊天图片时需要图片添加到相册的权限</string>
```

10.推送介绍

10.1、本地推送

```
/**
 自动提醒消息
 说明：如果开启自动提醒消息，当没有在智齿聊天页面的时候，都会主动把消息作为本地通知展示
 */
@property (nonatomic,assign) BOOL autoNotification;

[[ZCLibClient getZCLibClient] setAutoNotification:YES];
```

10.2、远程推送

智齿服务端会根据当前用户环境判断是否发送APNS，具体情况如下： * 当前消息已经到达用户端(即智齿自建通道仍在继续运行)不会发送远端推送 * 当前消息未到达用户但是未能判断自建通道是否无法使用(尝试连接中,比如短时间的网络波动)不会发送远端推送 * 当前消息未送达，通道已经中断，发送远端推送

10.3、注册和取消

- 注册生成appkey时 上传P12格式的推送证书，注意证书密码不要为空

- AppDelegate.m 文件中注册推送。

```
导入头文件
#import <SobotKit/SobotKit.h>
#import <UserNotifications/UserNotifications.h>
#define SYSTEM_VERSION_GRATERTHAN_OR_EQUALTO(v)  ([[[UIDevice currentDevice] systemVersion] compare:v options:NSNumericSearch] != NSOrderedAscending)
服从协议
<UIApplicationDelegate,UNUserNotificationCenterDelegate>
注册

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    [[UIApplication sharedApplication] setStatusBarHidden:YES withAnimation:UIStatusBarAnimationNone];
    [[UIApplication sharedApplication] setStatusBarHidden:NO withAnimation:UIStatusBarAnimationNone];

    if (SYSTEM_VERSION_GRATERTHAN_OR_EQUALTO(@"10")) {
        UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
        center.delegate = self;
        [center requestAuthorizationWithOptions:(UNAuthorizationOptionSound | UNAuthorizationOptionAlert |UNAuthorizationOptionBadge) completionHandler:^(BOOL granted, NSError * _Nullable error) {
            if (!error) {
```

```
                [[UIApplication sharedApplication] registerForRemoteNotifications]
;
            }
        }];
    }else{
         [self registerPush:application];
    }
    [[ZCLibClient getZCLibClient].libInitInfo setAppKey:@"your appKey"];
    // 设置推送是否是测试环境，测试环境将使用开发证书
    [[ZCLibClient getZCLibClient] setIsDebugMode:YES];
    // 错误日志收集
    [ZCLibClient setZCLibUncaughtExceptionHandler];
    return YES;
}

-(void)registerPush:(UIApplication *)application{
    // ios8后，需要添加这个注册，才能得到授权
    if ([[UIApplication sharedApplication] respondsToSelector:@selector(registerUs
erNotificationSettings:)]) {
        //IOS8
        //创建UIUserNotificationSettings，并设置消息的显示类类型
        UIUserNotificationSettings *notiSettings = [UIUserNotificationSettings set
tingsForTypes:(UIUserNotificationTypeBadge | UIUserNotificationTypeAlert | UIRemot
eNotificationTypeSound) categories:nil];

        [application registerUserNotificationSettings:notiSettings];

    } else{ // ios7
        [application registerForRemoteNotificationTypes:(UIRemoteNotificationTypeB
adge|UIRemoteNotificationTypeSound|UIRemoteNotificationTypeAlert)];
    }
}

- (void)application:(UIApplication *)application didRegisterForRemoteNotifications
WithDeviceToken:(NSData *)pToken{
    NSLog(@"---Token--%@", pToken);
    // 注册token
    [[ZCLibClient getZCLibClient] setToken:pToken];

}

// 是否自动提醒
[[ZCLibClient getZCLibClient] setAutoNotification:YES];

// 设置推送环境
[[ZCLibClient getZCLibClient] setIsDebugMode:NO];
```

10.4.SDK接收不到推送消息

解决方法：

- 检查上传的证书格式是否是P12
- 检查上传的证书开发环境和线上环境是否一致
- 测试远程推送请打包成adHoc包进行测试
- 检查是否调用过SDK移除推送的方法
- 检查推送是否注册成功

11.用户ID说明

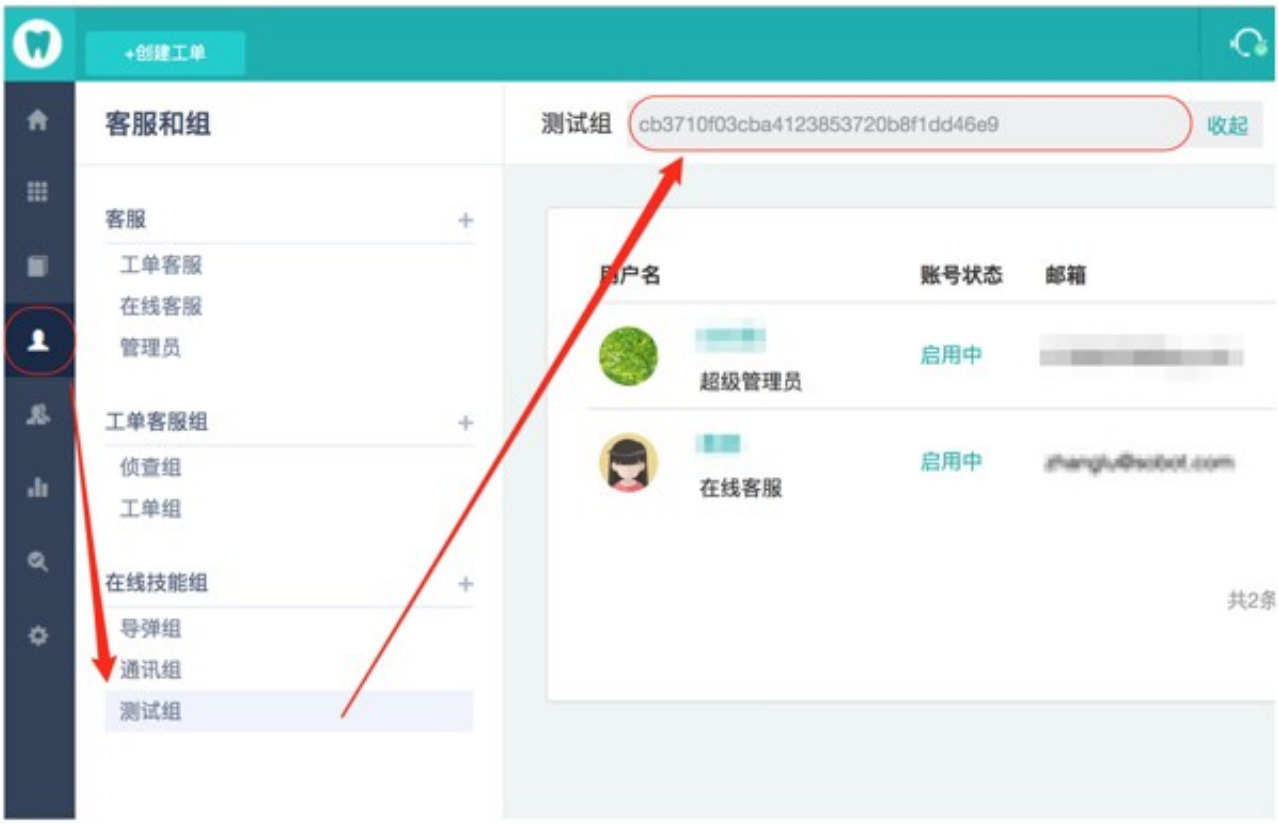
智齿保证多平台用户唯一性的标准,在使用中会保持一个单例的 `ZCLibInitInfo.h` 对象，通过ZCLibCient.h对象调用;

- 用户ID是记录用户的唯一标识，建议填写，不能传null，如果不传智齿会以设备ID自动生成一个做为标识
- 注意一点不要写死,比如0，如果用户ID写死，那么所有的用户都会成为同一用户，历史记录也是相同的。

```
/**
 *  用户唯一标识（对接用户可靠身份，不建议为null）
 *  null
 *  将自动备注到客户资料
 */
@property (nonatomic,strong) NSString   *userId;
```

12.常用功能说明

12.1、如何配置技能组
在PC端查看技能组ID



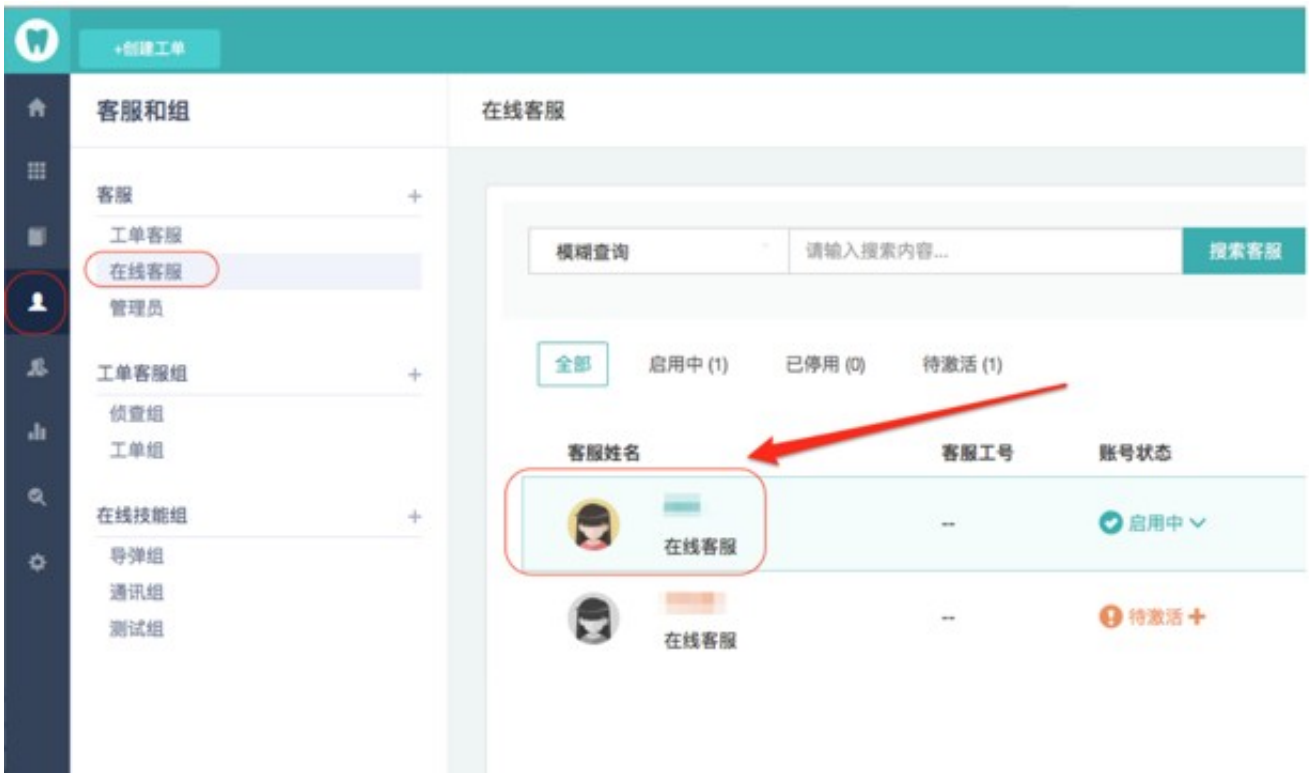

```
// 外部直接定义技能组（可选，如果传入技能组ID那么SDK内部转人工之后不在弹技能组的选择框，直接跳转到传入ID所对应的技能组中。）
// 如果选择传入技能组，根据传递的值转接到对应的技能组,不传不起作用

ZCLibInitInfo *initInfo = [ZCLibClient getZCLibClient].libInitInfo;
initInfo.skillSetId = @"技能组ID";
initInfo.skillSetName = @"技能名称";
```

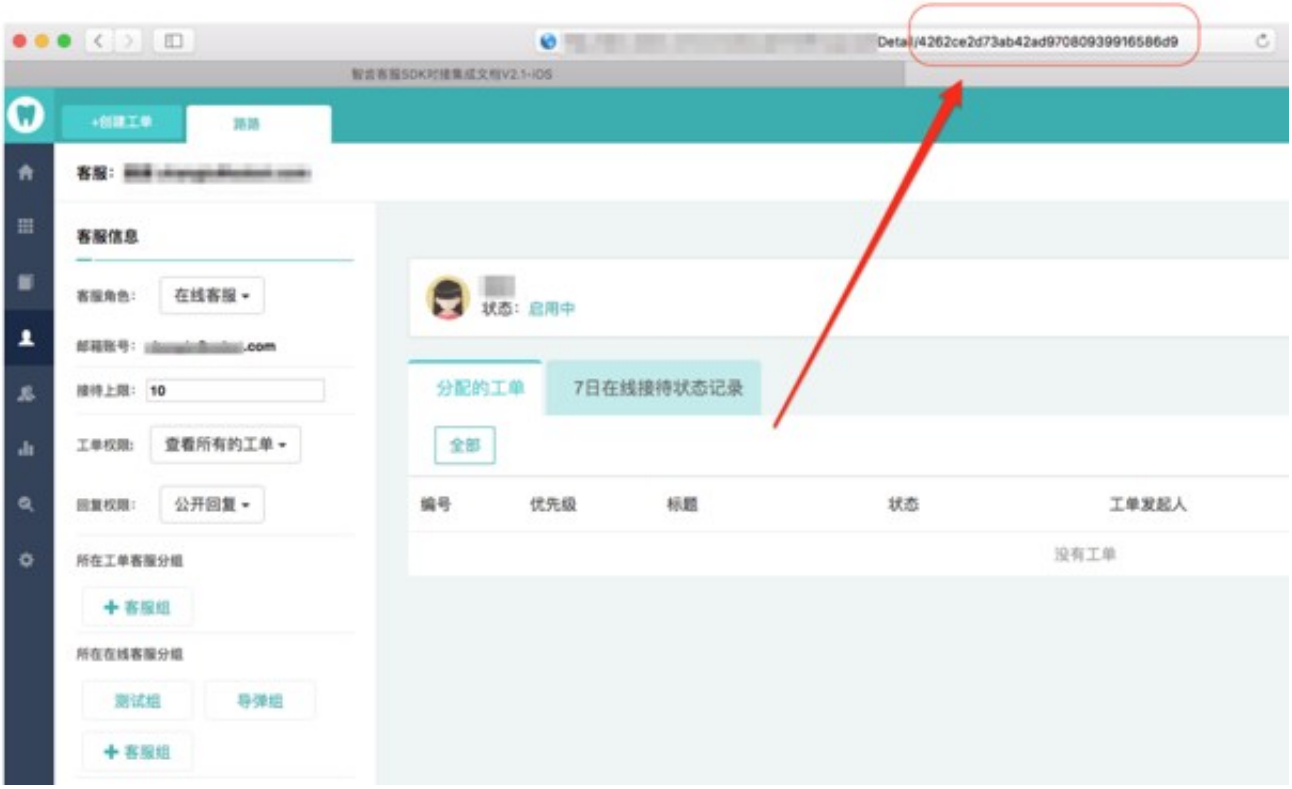
12.2、如何配置自动发送商品信息

```
/**
 *
 *  自定发送商品订单信息类型
 *  0 不发 1 给机器人发送 2 给人工发送  3 机器人和人工都发送
 **/
@property (nonatomic,assign) int goodMsgType;

/**
 *
 *   自动发送商品订单信息内容
 *   例如：“商品订单编号1403388282”
 **/
@property (nonatomic,copy) NSString * goodMsg;


//    initInfo.goodMsg = @"订购的商品订单号：124345890 该商品正在发货";
//    initInfo.goodMsgType = 3;
```

12.3、如何指定客服接待

获取指定客服ID






```
// 指定客服ID （指定对接的客服，如果不设置，取默认）
initInfo.receptionistId =  @"客服ID";
// 设置指定客服之后是否必须转入指定客服 （0 可转入其他客服 1 必须转入指定客服  注意：如果设置为1 ，当指定的客服不在线，不能再转接到其他客服）
initInfo.tranReceptionistFlag =  @"0";
```

12.4、如何开启语音转文字 （机器人接待模式ZCKitInfo.h）

```
/**
 是否开启机器人语音，（付费，否则语音无法识别）
 默认NO
 */
@property (nonatomic,assign) BOOL    isOpenRobotVoice;

kitInfo.isOpenRobotVoice = YES;
```

12.5、如何自定义底部+号菜单

```
/**
 * 自定义输入框下方更多(+号图标)按钮下面内容(不会替换原有内容，会在原有基础上追加)
 * 填充内容为：ZCLibCusMenu.h
 *  title:按钮名称
 *  url：点击链接(点击后会调用初始化linkBock)
 *  imgName:本地图片名称，如xxx@2x.png,icon=xxx
 */
@property (nonatomic,strong) NSMutableArray * cusMoreArray;


// 添加
NSMutableArray *arr = [[NSMutableArray alloc] init];
    for (int i = 0; i<20; i++) {
        ZCLibCusMenu *menu1 = [[ZCLibCusMenu alloc] init];
        menu1.title = [NSString stringWithFormat:@"测试%d",i+1];
        menu1.url = [NSString stringWithFormat:@"https://www.baidu.com/%d",i+1];;
        menu1.imgName = @"zcicon_sendpictures";
        [arr addObject:menu1];
    }

    uiInfo.cusMoreArray = arr;
```

12.6、发送位置

```
发送位置信息，此方法，只能是收到初始化的链接监听到url=sobot://sendlocation时发起定位调用，否则可能发送失败，详细说明见ZCKitInfo.h中的canSendLocation属性说明
// 发送位置信息
1.设置 canSendLocation 为true 开启发送位置
2.在初始化方法中 实现发送位置的传参和点击定位cell的跳转
[注意:初始化方法从2.7.0开始messageLinkClick带有Bool返回值，true自己处理，false表示不处理]
// target  如果传入的参数不为nil 需要实现 -(void)openLeaveMsgClick:(NSString*)tipMsg;代理方法 留言跳转到用户自定义的留言页面

[ZCSobot startZCChatVC:uiInfo with:self target:nil pageBlock:^(id object, ZCPageBlockType type) {
                // 点击返回
                if(type==ZCPageBlockGoBack){
//                    NSLog(@"点击了关闭按钮");
                }

                // 页面UI初始化完成，可以获取UIView，自定义UI
                if(type==ZCPageBlockLoadFinish){
//                    NSLog(@"页面加载完成");
                }
    } messageLinkClick:^(NSString *link) {
        NSLog(@"%@",link);
//        当收到link = sobot://sendlocation 调用智齿接口发送位置信息
```

```
//        当收到link = sobot://openlocation?latitude=xx&longitude=xxx&address=xxx 可根据自己情况处理相关业务
        if( [link hasPrefix:@"sobot://sendlocation"]){
            //发送坐标点
            NSString *fullPath = GetDocumentsFilePath(fname);
            [imageData writeToFile:fullPath atomically:YES];
            // 发送位置信息
            [ZCSobot sendLocation:@{
                                    @"lat":@"40.001693",
                                    @"lng":@"116.353276",
                                    @"localLabel":@"北京市海淀区学清路38号金码大厦A座23层金码大酒店",
                                    @"localName":@"云景四季餐厅",
                                    @"file":fullPath}];
            return YES;
        }else if([link hasPrefix:@"sobot://openlocation"]){
            // 解析经度、纬度、地址：latitude=xx&longitude=xxx&address=xxx
            // 跳转到地图的位置
            NSLog(link);
            // 测试打开地图 高德网页版
            NSString * urlString = @"";
            urlString = [[NSString stringWithFormat:@"http://uri.amap.com/marker?position=%f,%f&name=%@&coordinate=gaode&src=%@&callnative=0",@116.353276,@40.001693,@"北京市海淀区学清路38号金码大厦A座23层金码大酒店",@"智齿SDK"] stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
            [[UIApplication sharedApplication]openURL:[NSURL URLWithString:urlString]];
          return YES;
        }

        return NO;
    }];
```

13.电商版本配置

13.1、区别

> 智齿SDK电商版针对电商平台公司下有多商家的场景使用，每个商家都是单独的appkey

13.2、配置

```
/**
 平台标识
 */
@property (nonatomic,strong) NSString     *platformUnionCode;

[ZCLibClient getZCLibClient].platformUnionCode = @“您注册的平台ID”;


/**
 *
 * 私钥
 *
 **/
@property (nonatomic,copy) NSString * platformKey;

ZCLibInitInfo * initInfo = [ZCLibClient getZCLibClient].libInitInfo;
initInfo.platformKey = @"您注册时生成的私钥";
```

13.3、功能使用，消息溢出、消息中心

- 电商版启动

```
[ZCLibClient getZCLibClient].platformUnionCode = @“您注册的平台ID”;
ZCLibInitInfo *info = [ZCLibClient getZCLibClient].libInitInfo;

// 获取AppKey
initInfo.appKey = @“appkey”;
initInfo.userId = @“用户ID”;
initInfo.platformKey = @"私钥";
[[ZCLibClient getZCLibClient] setLibInitInfo:initInfo];

ZCKitInfo *uiInfo=[ZCKitInfo new];

[ZCSobot startZCChatVC:uiInfo
                     with:self
                   target:nil
                pageBlock:^(ZCChatController *object,ZCPageBlockType type){

                } messageLinkClick:nil];
```

- 溢出功能配置

电商版转人工溢出功能（可选），需要配置已下参数

```
// 1 是开启，默认0不开启
@property (nonatomic,assign)  int   flowType;
// 转接到的公司ID
@property (nonatomic,strong)  NSString  * flowCompanyId;
// 转接到的公司技能组
@property (nonatomic,strong)  NSString  * flowGroupId;


ZCLibInitInfo *info = [ZCLibClient getZCLibClient].libInitInfo;

initInfo.flowType = 1;

initInfo.flowCompanyId = @"转接到的公司ID";

initInfo.flowGroupId = @"转接到的公司技能组";

[[ZCLibClient getZCLibClient] setLibInitInfo:initInfo];
```

- 消息中心

```
[ZCLibClient getZCLibClient].platformUnionCode = @“您注册的平台ID”;
ZCLibInitInfo *info = [ZCLibClient getZCLibClient].libInitInfo;

    info.userId = @“用户id”
    info.appKey = @“appkey”;
     info.platformKey = @"私钥";
    [[ZCLibClient getZCLibClient] setLibInitInfo:info];

    ZCKitInfo *uiInfo=[ZCKitInfo new];

// 进入消息列表页面
[ZCSobot startZCChatListView:uiInfo with:self onItemClick:^(ZCUIChatListController
 *object, ZCPlatformInfo *info) {
// 点击消息列表 进入SDK聊天页面
        [ZCSobot startZCChatVC:uiInfo with:object target:nil pageBlock:^(ZCChatCon
troller *object, ZCPageBlockType type) {

        } messageLinkClick:nil];
    }];


     /**
    删除指定商户 (先从商户列表接口获取 listId)
    ZCPlatformInfo *model = [_listArray objectAtIndex:indexPath.row];
            [[ZCPlatformTools sharedInstance] deletePlatformByAppKey:zcLibConvertT
oString(model.appkey) user:zcLibConvertToString(model.userId)];

        [[ZCLibServer getLibServer] delPlatformMemberByUser:model.listId start:^{
        } success:^(NSDictionary *dictionary, ZCNetWorkCode sendCode) {
            if (dictionary && [dictionary[@"code"] intValue] ==1) {
                [_listArray removeObject:model];
                [_listTable reloadData];
            }
        } failed:^(NSString *errorMessage, ZCNetWorkCode errorCode) {

        }];
     */
```