

智齿客服SDK-iOS对接集成文档V2.2.0

一、集成准备

- [1.1 注册AppKey](#)
- [1.2 下载智齿SDK](#)
- [1.3 包文件说明](#)
- [1.4 导入SDK和添加依赖库](#)

二、快速集成

- [2.1 快速集成SDK](#)
- [2.2 使用CocoaPods集成](#)
- [2.3 初始化参数配置](#)
- [2.4 UI配置](#)
- [2.5 推送配置和消息监听](#)
- [2.6 关闭推送](#)
- [2.7 离线消息](#)
- [2.8 未读消息](#)

三、详细参数说明

- [3.1 ZCLibInitInfo类说明](#)
- [3.2 ZCKitInfo类说明\(主要UI相关配置\)](#)
- [3.3 ZCProductInfo类说明 \(商品信息\)](#)
- [3.4 ZCLibMessage \(消息对象\)](#)
- [3.5 ZCLibRich \(富媒体消息类\)](#)
- [3.6 ZCLibSkillSet类说明\(技能组\)](#)
- [3.7 请求接口返回状态字段说明](#)

四、接口说明

- [4.2 获取会话编号列表](#)
- [4.3 获取历史记录](#)
- [4.4 转人工](#)
- [4.5 发送消息](#)
- [4.6 获取机器人引导语](#)
- [4.7 提交评价](#)
- [4.8 获取技能组](#)
- [4.9 发送正在输入内容](#)
- [4.10 提交留言](#)
- [4.11 退出SDK](#)

[4.12 清除历史记录](#)

[4.13 留言上传图片](#)

[4.14 机器人顶踩](#)

[五、常见问题](#)

[5.1 启动后页面直接返回](#)

[5.2 iOS 10相机、相册、麦克风权限问题](#)

[5.3 无法正常接收推送消息](#)

[5.4 没有收到客服发送的消息](#)

一、集成准备

1.1注册AppKey

请先确保已注册智齿客服账号<https://www.sobot.com>。

1.2下载智齿SDK

点击下载最新SDK包。 [iOS_SDK](#)

1.3包文件说明

解压下载【iOS_SDK】,下载完毕为一个压缩包，包含（[SobotKit.framework](#)、[SobotKit.bundle](#)和[ZCEmojiExpression.bundle](#)）、[SobotDemo](#)和相关说明文档。

文件名	说明
SobotKit.framework	智齿SDK核心代码库
SobotKit.bundle	图片资源文件
ZCEmojiExpression.bundle	表情资源文件
ZCLibInitInfo.h	实体类：基础参数类（自定义用户配置信息）
ZCKitInfo.h	实体类：基础参数（UI自定义）
ZCSobot.h	连接客户端到智齿客服，该文件提供接入功能
SobotKit.h	该文件引用所有SDK需要的文件
ZCUIBaseController	UI界面父类，定义顶部View
ZCUIChatController	主要UI界面，如果需要自己实现跳转，可以自己引用实现
SobotLocalizable.strings	国际化语言文件
ZCLibClient.h	全局配置类

1.4 导入SDK和添加依赖库

把下载的SDK-Release文件夹中的文件拖入你的工程里。智齿iOS_SDK 的实现，依赖了一些系统的框架，在开发应用时需要在工程里加入这些框架。开发者首先点击工程右边的工程名，然后在工程名右边依次选择 TARGETS -> Build Phases -> Link Binary With Libraries，展开 Link Binary With Libraries 后点击展开后下面的 + 来添加下面的依赖项：

```
AVFoundation.framework
AssetsLibrary.framework
AudioToolbox.framework
SystemConfiguration.framework
MobileCoreServices.framework
libz.1.2.5.tbd( dylib)
```

二、快速集成

2.1 快速集成SDK

智齿SDK提供了一套开源的聊天界面，帮助开发者快速集成，并提供自定义接口，以实现定制需求。只需以下几行代码便可快速启动智齿SDK应用。

```
// 配置参数类
ZCLibInitInfo *initInfo = [ZCLibInitInfo new];
initInfo.appKey = 注册的appkey
// 将配置参数传入
[ZCLibClient getZCLibClient].libInitInfo = initInfo;
ZCKitInfo *uiInfo = [ZCKitInfo new]; // 自定义UI参数类
// 启动智齿SDK
[ZCSobot startZCChatView:uiInfo with:self
                pageBlock:^(ZCUIChatController *object, ZCPageBlockType type){}
                messageLinkClick:nil];
```

2.2 使用CocoaPods集成

在 Podfile 文件中加入: `pod 'SobotKit', '~> 2.1.0'`

如果搜索不到最新版本, 请运行以下命令更新CocoPods仓库

```
pod repo update --verbose
```

2.3 初始化参数配置

详情见ZCLibInitInfo对象属性介绍, 事例代码如下:

```

ZCLibInitInfo *initInfo = [ZCLibInitInfo new];
initInfo.customInfo =
    @{
        @"key1":@"values1",
        @"key2":@"value2"
        ... ...
    };

// AppKey, 用于标识AppKey, 初始化必须设置, 否则初始化不成功。
initInfo.appKey    = @"你的AppKey";

// 唯一标识用户(注意: 如果不传会默认手机唯一标识, 同一个手机的用户聊天信息相同)
// 【不能写入固定值, 这样所有相同userId的聊天信息是共享的】
initInfo.userId    = @"用户userId";

// 技能组的名称
initInfo.skillSetName = @"技能组名称";
// 技能组ID
initInfo.skillSetId    = @"技能组ID";
// 对接到指定的客服
initInfo.receptionistId = @"客服ID";
// 是否必须转入指定客服
initInfo.tranReceptionistFlag = NO;
// 对接的机器人编号
initInfo.robotId = _configModel.robotId ;
// 历史记录时间范围, 单位分钟(例:100-表示从现在起前100分钟的会话)
initInfo.scopeTime = 0;

```

2.4 UI配置

根据ZCUIKitInfo对象的属性设置, 详情见ZCUIKitInfo属性介绍; 事例代码如下:

```
ZCUIKitInfo * kitInfo = [ZCUIKitInfo new];
// 左边气泡的颜色
kitInfo.leftChatColor = [UIColor redColor];
// 右边气泡的颜色
kitInfo.rightChatColor = [UIColor greenColor];
...    ...    (详见请参考demo)

//如果要展示商品信息, goodTitle和goodLink是必选字段。
ZCProductInfo *productInfo = [ZCProductInfo new];
productInfo.thumbUrl = @"缩略图url";
productInfo.title = @"标题";
productInfo.desc = @"描述";
productInfo.label = @"客服价格";
productInfo.link = @"产品链接";
// 赋值传入
uiInfo.productInfo = productInfo;
```

2.5 推送配置和消息监听

智齿iOS_SDK支持将无法通过通道送达的消息通过推送送达给APP用户, 需要在
[<https://www.sobot.com/console/settings/channelApp>]添加APP时上传推送证书, 格式为.p12.

添加App

App名称*:

请输入App名称

开启推送:

☒ IOS

☐ Android

IOS推送配置

bundle ID*:

请输入bundle ID

开发环境证书:

证书文件:

选择证书

生产环境证书:

证书文件:

选择证书

确定

配置示例:

在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 {
```

```

- (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_GREATER_THAN_OR_EQUAL_TO(@"10.0")) {
        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];
    }

    // 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(registerUserNotificationSettings)]) {
        //IOS8
        //创建UIUserNotificationSettings，并设置消息的显示类类型
        UIUserNotificationSettings *notiSettings = [UIUserNotificationSettings settingsForTypes:(UIUserNotificationTypeBadge | UIUserNotificationTypeAlert | UIRemoteNotificationTypeSound) categories:nil];

        [application registerUserNotificationSettings:notiSettings];
    } else { // ios7
        [application registerForRemoteNotificationTypes:(UIRemoteNotificationTypeBadge|UIRemoteNotificationTypeSound|UIRemoteNotificationTypeAlert)];
    }
}

```



```

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

```

APNS消息推送及提醒

```

// 是否自动提醒
[[ZCLibClient getZCLibClient] setAutoNotification:YES];
// 设置推送环境
[[ZCLibClient getZCLibClient] setIsDebugMode:NO];

```

2.6 关闭推送

```

[[ZCLibClient getZCLibClient] removePush:^(NSString *uid, NSData *token, NSError *
error) {
    if((uid==nil && token==nil) || error!=nil){
        // 移除失败，可设置uid或token(uid可不设置)后再调用
    }else{
        // 移除成功
    }
}
}];

```

2.7 消息监听

```

// 初始化智齿客服，会建立长连接通道，监听服务端消息（建议启动应用时调用，没有发起过咨询不会浪费资源，至少转一次人工才有效果）
[[ZCLibClient getZCLibClient] initWithCaht];

// 消息监听
[ZCLibClient getZCLibClient].delegate = self;
或
[ZCLibClient getZCLibClient].receivedBlock:^(id obj,int unread){
    NSLog(@"未读消息数量：\n%d,%@",unread,obj);
}];

// 关闭通道，清理内存，退出智齿客户
说明：调用此方法后将不能接收到离线消息，除非再次进入智齿SDK重新激活
isClosePush:YES ,是关闭push; NO离线用户，但是可以收到push推送+
(void) closeAndoutZCServer:(BOOL) isClosePush;
[ZCLibClient closeAndoutZCServer:YES];

```

2.8未读消息

```
//直接获取未读消息数
[[ZCLibClient getZCLibClient] getUnReadMessage];
```

三、详细参数说明

####3.1 ZCLibInitInfo类说明

属性名称	是否必选	数据类型	说明
customInfo	可选	NSMutableDictionary	自定义属性，以KEY-VALUE键值对存储。
tel	可选	String	用户手机号码，默认为空，用于客服工作台显示。该字段为可选择字段。
realname	可选	String	用户姓名，默认为空，用于客服工作台显示，该字段为可选字段。
email	可选	String	用户邮箱，默认为空，用于客服工作台显示，该字段为可选字段。
nickName	可选	String	用户昵称，默认为空，用于客服工作台显示，该字段为可选字段。
weiChat	可选	String	用户微信号码，默认为空，用于客服工作台显示，该字段为可选字段。
weiBo	可选	String	用户微博号码默认为空，用于客服工作台显示，该字段为可选字段。
userSex	可选	String	用户手机号码，默认为空，用于客服工作台显示。该字段为可选择字段。
userBirthday	可选	String	用户的生日默认为空，用于客服工作台显示，该字段为可选字段。传入格式：20170323 如果不是这个格式，初始化接口会给过滤掉
userRemark	可选	String	备注默认为空，用于客服工作台显示，该字段为可选字段。
face	可选	String	用户的头像Url,不设置，取默认值（SDK图片资源包中的用户默认头像）用于聊天页面中用户头像的，该字

			段为可选字段。
qqNumber	可选	String	用户的QQ号码，用于客服工作台显示，该字段为可选字段。
avatarUrl	可选	String	接入来源页Url，标记用户从哪个浏览页面进入SDK，改字段为可选字段。
sourceTitle	可选	String	接入来源页标题，标记用户浏览页面标题，该字段为可选字段。
sourceURL	可选	String	接入来源页url,该字段为可选字段。
自定义字段	可选	String	自定义字段名，以key/value的方式传入，用户可以自己命名参数名(参见示例)。
appKey	必选	String	App唯一标识，必须设置，不设置初始化不成功。
receptionistId	可选	String	指定客服ID，设置指定客服ID，客户可以在转人工的操作中直接转到指定的客服，不设置按默认的转人工流程执行。
tranReceptionistFlag	可选	int	转指定客服不在线，是否可以再转到其他客服（0 可转入其他客服 1 必须转入指定客服）。
userId	可选	String	用户标识（对接用户可靠身份，不建议为null）。
robotId	可选	String	对接机器人ID，可以对接指定机器人。
skillSetId	可选	String	对接技能组ID，设置技能组ID之后，转人工不在弹技能组弹框，直接转到该技能组中的客服，不设置此字段按默认的转人工流程执行。
skillSetNam	可选	String	对接技能组名称，和对接的技能组ID要匹配，设置技能组昵称之后，转人工不在弹技能组弹框，直接转到该技能组中的客服，不设置此字段按默认的转人工流程执行。
serviceModel	可选	String	自定义接入模式（0 默认跟随系统设置 1仅机器人 2仅人工 3机器人优先 4人工优先）。

titleType	可选	String	聊天页顶部标题 自定义方式（0 默认 1 企业名称 2.自定义字段）。
customTitle	可选	String	聊天页顶部标题，自定义字段（如果传入优先使用该字段）。
scopeTime	可选	String	历史记录时间范围，单位分钟（可设置有效范围：10分钟~48小时之间）。

3.2 ZCKitInfo 类说明(主要UI相关配置)

属性名称	是否必选	数据类型	说明
isOpenEvaluation	可选	BOOL	默认为NO 未开启,点击返回时是否开启满意度评价
isCloseAfterEvaluation	可选	BOOL	默认为NO 未开启,评价完人工是否关闭会话（人工满意度评价后释放会话）
isShowTansfer	可选	BOOL	默认为YES 显示转人工按钮,机器人优先模式，是否直接显示转人工按钮（值为NO时，会在人工无法回答时显示转人工按钮）
unWordsCount	可选	String	机器人优先模式，通过记录机器人未知说辞的次数设置是否直接显示转人工按钮
isOpenRecord	可选	BOOL	默认YES，开启。是否开启语音功能。
isShowPhoneNumber	可选	BOOL	默认NO不显示。留言页面是否显示手机号输入框
isShowNickName	可选	BOOL	默认NO不显示，留言页面是否显示昵称输入框
isAddPhoneNumber	可选	BOOL	默认NO，非必须输入，留言时是否必须输入手机号
isAddNickName	可选	BOOL	默认为NO，非必须输入，留言时是否必须输入昵称
titleFont	可选	UIFont	顶部标题的font

listTitleFont	可选	UIFont	页面返回按钮，输入框，评价提交按钮。Toast 提示语font
listDetailFont	可选	UIFont	网络提醒font
customlistDetailFont	可选	UIFont	评价页面中“是否有以下情况” font
listTimeFont	可选	UIFont	消息提醒（转人工。客服接待等） font
chatFont	可选	UIFont	聊天气泡中文字 font
voiceButtonFont	可选	UIFont	录音按钮的文字 font
goodsTitle	可选	UIFont	商品详情cell 中title的文字 font
goodsDetFont	可选	UIFont	商品详情cell中摘要的文字 font
goodSendBtnColor	可选	UIColor	商品详情cell中btn的背景色
backgroundColor	可选	UIColor	对话页面背景色
customBannerColor	可选	UIColor	自定义风格颜色：导航
imagePickerColor	可选	UIColor	相册导航栏的颜色
imagePickerTitleColor	可选	UIColor	相册导航栏的标题颜色
leftChatColor	可选	UIColor	左边聊天气泡的颜色
rightChatColor	可选	UIColor	右边聊天气泡的颜色
leftChatSelectedColor	可	UIColor	左边气泡复制选中的颜色

	选		
rightChatSelectedColor	可选	UIColor	右边气泡复制选中的颜色
backgroudBottomColor	可选	UIColor	底部bottom的背景颜色
commentoTherButtonBgColor	可选	UIColor	评价普通按钮选中背景颜色和边框颜色
commentCommitButtonColor	可选	UIColor	评价（立即结束、取消）按钮文字颜色
BgTipAirBubblesColor	可选	UIColor	提示气泡的背景颜色
videoCellBgSelColor	可选	UIColor	语音cell选中的背景色
LineRichColor	可选	UIColor	富文本中线条颜色
submitEvaluationColor	可选	UIColor	提交评价按钮的文字颜色
topViewTextColor	可选	UIColor	顶部文字颜色
leftChatTextColor	可选	UIColor	左边气泡文字颜色
rightChatTextColor	可选	UIColor	右边气泡文字颜色
timeTextColor	可选	UIColor	时间文字的颜色
tipLayerTextColor	可选	UIColor	提示气泡的颜色
serviceNameTextColor	可选	UIColor	客服昵称颜色
nickNameTextColor	可选	UIColor	提示cell中客服昵称的文字颜色
	可		

chatLeftLinkColor	选	UIColor	左边气泡中的链接颜色
chatRightLinkColor	可选	UIColor	右边气泡中链接颜色
goodsTitleTextColor	可选	UIColor	商品cell中title的文字颜色
goodsTipTextColor	可选	UIColor	商品cell中标签的文字颜色
goodsDetTextColor	可选	UIColor	商品cell中摘要的文字颜色
goodsSendTextColor	可选	UIColor	商品详情cell中发送的文字颜色
isSetPhotoLibraryBgImage	可选	BOOL	是否设置相册背景图片

3.3 ZCProductInfo类说明（商品信息，本信息可选）

属性名称	数据类型	说明
thumbUrl	String	商品图片链接地址
title	String	商品标题
desc	String	商品摘要，商品的描述
label	String	商品标签，例如价格
link	String	发送链接，发送给客服的内容

3.4 ZCLibMessage（消息对象）

属性名称	数据类型	说明
date	String	会话时间
cid	String	会话id
action	String	会话类型0, 用户, 1机器人 2客服
sender	String	用户id
senderName	String	用户名称
senderType	int	发送,0, 自己, 1机器人, 2客服
senderFace	String	发送者头像
t	String	发送时间戳 -long
ts	String	发送时间
receiver	String	客服ID
receiverName	String	客服名称
offlineType	String	下线类型
receiverFace	String	客服头像
isRead	BOOL	是否已读
progress	float	上传进度
sendStatus	int	发送状态, 1 开始发送, 2发送失败, 0, 发送完成
isPlaying	BOOL	是否播放
isHistory	BOOL	是否是历史消息
richModel	ZCLibRich	富媒体消息类
sysTips	String	提醒,上下线等
tipStyle	int	提示类型

3.5 ZCLibRich（富媒体消息类）

属性名称	数据类型	说明
richmoreurl	String	富媒体更多地址,“查看更多的”url
richpicurl	String	富媒体图片地址
msg	String	消息体，当时图片和语音时，为图片、语音地址
duration	String	语音时长
msgType	int	0文本 1图片 2音频 4 富文本中有图片 5 富文本中纯文字 6 富文本中有视频
suggestions	NSArray	引导说辞不带docId
suggestionArr	NSArray	引导说辞带docId
answerType	int	1 直接回答, 2 理解回答, 3 不能回答, 4引导回答, 6互联网寒暄, 7 私有寒暄（包括第三方天气、快递接口）,8百科, 9 向导回答,10 业务接口
stripe	String	建议引导
time	String	时间

3.6 ZCLibSkillSet类说明 (技能组)

属性名称	数据类型	说明
groupId	String	技能组ID
channelType	String	来源
groupName	String	技能组名称
companyId	String	企业编号
isOnline	BOOL	客服是否在线

3.7请求接口返回状态说明

字段名	数据类型	说明
ZCNetWorkCode	NSInteger	访问成功, ZCNETWORKSUCCESS = 0 访问成功, 但是空数据, ZCNETWORKNULLDATA = 1。 连接失败, ZCNETWORK_FAIL = -1。
ZCConnectUserStatusCode	NSInteger	人工服务连接失败, 已重新连接,ZCConnectUserServerFailed = -1 连接失败,ZCConnectUserOfWaiting = 0 连接成功,ZCConnectUserSuccess = 1 没有客服在线,ZCConnectUserNoAdmin = 2 用户被拉黑,ZCConnectUserBeBlock = 3 用户已经转人工无需再次请求,ZCConnectUserBeConnected = 4
ZCMessageSendCode	NSInteger	发送消息成功,ZCSENDMessageSuccess = 0 新消息回复,ZCSENDMessageNew = 1 发送消息失败,ZCSENDMessageFAIL = -1 发送消息失败,ZC_SENDMessageFAILSTATUS = 1002 发送消息为空,ZCSENDMessageNULL = -2 文件不存在,ZCFile_NotFound = -1000
errorMessage	String	接口返回错误提醒消息

四、接口说明

4.1 初始化接口

```
- (void)initSobotSDK:(void (^)(ZCLibConfig *config))successBlock
    error:(void (^)(ZCNetWorkCode status))errorBlock
    appIdIncorrect:(void (^)(NSString *appId))appIdIncorrectBlock;
```

连接到智齿客服,成功调用一次就可以, 主要初始化一些必要参数

请求参数:

有效参数从[ZCLibClient getZCLibClient].libInitInfo中获取, 调用初始化参数之前必须设置此对象的相关属性

字段名	数据类型	说明
successBlock	Block	初始化成功，返回初始化对象
errorBlock	Block	初始化失败，返回失败状态
sysNumIncorrectBlock	Block	企业编号错误

返回参数(ZCLibConfig):

字段名	数据类型	说明
adminHelloWord	String	客服欢迎语
adminNonelineTitle	String	客服不在线提示语
adminTipTime	String	客服无应答提示时间
cid	String	会话ID
companyId	String	用于留言工单的ID
color	String	颜色
companyName	String	企业名称
companyStatus	String	公司付费状态
face	String	用户头像
groupflag	BOOL	是否显示技能组
guideFlag	int	是否显示机器人引导语 1.开启 0关闭
inputTime	int	发送正在输入的频率
isblack	BOOL	是否是黑名单
manualCommentTitle	String	评价标签以逗号隔开
msgFlag	int	留言开关 0关闭 1开启
msgTmp	String	留言内容模板
msgTxt	String	留言引导文案
onORoff	int	1.开启语音 2关闭语音工能 3只选择机器人
robotCommentTitle	String	机器人评论标签已逗号隔开

robotHelloWord	String	机器人欢迎语
robotLogo	String	机器人头像
robotName	String	机器人昵称
robotUnknownWord	String	机器人未知问题说辞
type	int	接入方式, 1只有机器人 2仅人工 3智能客服-机器人优先 4智能客服-人工客服优先
uid	String	用户ID
uname	String	用户的昵称
userOutTime	int	用户超时下线时间
userOutWord	String	用户超时下线说辞
userTipTime	int	用户无应答时间
userTipWord	String	用户无应答文案
ustatus	int	用户当前状态 -2排队中 -1机器人 0 离线 1在线
manualType	String	机器人引导转人工 勾选为1, 默认为0 (直接回答勾选, 理解回答勾选, 引导回答勾选, 未知回答勾选)
realuateFlag	BOOL	顶踩开关 0关闭 1开启
telShowFlag	BOOL	电话是否显示
telFlag	BOOL	电话是否必填
enclosureShowFlag	BOOL	附件是否显示
enclosureFlag	BOOL	附件是否必填 0 可选 1 必填
emailFlag	BOOL	邮箱是否必填
emailShowFlag	BOOL	邮箱是否显示
ticketStartWay	BOOL	工单发起方式 1邮箱, 2手机

4.2 获取会话编号列表

```
-(void)getChatUserCids:(int) time
    config:(ZCLibConfig *) config
    start:(void(^)( )) startBlock
    success:(void(^)(NSDictionary *dict,ZCNetWorkCode sendCode)) successBlock
    failed:(void(^)(NSString *errorMessage,ZCNetWorkCode errorCode)) failedBlock;
```

传入参数：

字段名	数据类型	说明
time	int	历史记录的时间范围，单位为分钟，默认0（有效值为：10分钟到48小时之内）
config	ZCLibConfig	ZCLibConfig初始化成功后可以在ZCIMChat中获取，当前接口需要ZCLibConfig中uid
startBlock	Block	开始请求
successBlock	Block	请求成功
failedBlock	Block	请求失败

返回参数和示例：

字段名	数据类型	说明
dict	NSDictionary	cid 列表数据，用于历史记录查询
ZCNetWorkCode	NSInteger	网络请求状态码

4.3 获取历史记录

```
-(void)getHistoryMessages:(NSString *) cid
    withUid:(NSString *) uid
    start:(void(^)( )) startBlock
    success:(void(^)(NSMutableArray *messages,ZCNetWorkCode sendCode
)) successBlock
    failed:(void(^)(NSString *errorMessage,ZCNetWorkCode errorCode)
) failedBlock;
```

传入参数示例：

字段名	数据类型	说明
cid	String	当前要查询的cid
uid	String	初始化接口返回的用户ID
startBlock	Block	开始请求的回调
successBlock	Block	请求成功的回调
failedBlock	Block	请求失败的回调

返回消息对象(ZCLibMessage)列表：

字段名	数据类型	说明
tempArr	NSMutableArray	聊天记录的数据
ZCNetWorkCode	NSInteger	网络状态码

4.4 转人工

```
-(void)connectOnlineCustomer:(NSString *) groupId
    groupName:(NSString *) groupName
    config:(ZCLibConfig *) _config
    Aid:(NSString *) aid
    TranFlag:(int) tranFlag
    start:(void (^)())startBlock
result:(void (^)(NSDictionary *dict, ZCConnectUserStatusCode status)) resultBlock;
```

传入参数：

字段名	数据类型	说明
groupId	String	技能组ID，技能组ID的来源（1 初始化用户配置的技能组ID 2点击技能组弹框中某个技能组ID）如果没有设置请传""。
groupName	String	技能组名称，技能组ID的来源（1 初始化用户配置的技能组昵称 2点击技能组弹框中某个技能组名称）如果没有设置请传""
config	ZCLibConfig	当前接口主要使用到ZCLibConfig类中uid,cid, aid String 客服ID，初始化配置参数中设置，可为空
tranFlag	int	当前接口主要使用到ZCLibConfig类中uid,cid, aid String 是否强转到当前指定客服，默认为0，不强转（此参数只在设置客服ID时有效）
aid	String	客服ID，初始化配置参数中设置，可为空
startBlock	block	开始请求的回调
resultBlock	block	请求结果 ZCConnectUserStatusCode（智齿链接业务状态码 -1 人工链接失败，已重新连接 0 连接失败 1 连接成功 2没有客服在线 3 用户被拉黑 4用户已经转人工无需再次请求）

返回参数：

字段名	数据类型	说明
aface	String	客服头像
aid	int	客服ID
aname	String	客服昵称
status	int	转人工状态码 -1 人工服务连接失败，已重新连接 0连接失败 1连接成功 2没有客服在线 3用户被拉黑 4用户已经转人工无需再次请求 5用户超时下线
wslink.bak	String	连接通道IP
ZCConnectUserStatusCode	NSInteger	智齿连接业务状态码

返回参数示例

```

{
    code = 1;
    data = {
        aface = "https://img.sobf.jpg";
        aid = 4005;
        aname = @"智齿客服";
        pu = "https://ws.sobot.com/webchat6/webchat";
        puid = b280aca787c54b3abd898136a12696f5;
        status = 1;
        "wslink.bak" = (
            "60.205.91.1:6000"
        );
        "wslink.default" = "xxx.xxx.xxx.xxx:xxxx";
    };
    msg = "<null>";
}

```

4.5 发送消息

```

-(void)sendMessage:(NSString *)message
    questionId:(NSString*)docId
    msgType:(ZCMessageType ) type
    duration:(NSString *)duration
    config:(ZCLibConfig *)config
    robotFlag:(NSString *)roboflag
    start:(void(^)(ZCLibMessage *message)) startBlock
    success:(void(^)(ZCLibMessage *message,ZCMessageSendCode sendCode)) successBlock
    progress:(void(^)(ZCLibMessage *message)) progressBlock
    fail:(void(^)(ZCLibMessage *message,ZCMessageSendCode errorCode)) failBlock;

```

SDK中将发送消息给机器人和人工客服的接口以及上传图片语音的调用方法合并，根据ZCLibConfig中isArtificial（当前是否是人工）判定发送给机器人或者人工客服。如果发送的是图片，上传的进度由ZCLibConfig中progress获取。发送成功或者失败的状态由ZCLibConfig中sendStatus获取（1 开始发送 2发送失败 0发送完成）。successBlock会调用多次，根据ZCMessageSendCode判断具体操作

传入参数：

字段名	数据类型	说明
docId	String	引导说辞的编号
type	ZCMessageType	消息类型 (0 文本 1图片 2声音 3富文本 4正在录音 5取消正在闪烁的语音cell)
duration	String	声音长度, 例如00: 20 字符串类型, 直接用于显示,如果发送的消息类型不是语音可为空
robotFlag	String	机器人ID , 可为空
config	ZCLibConfig	当前接口需要传入ZCLibConfig类中uid和cid
startBlock	Block	开始发送, 返回用于显示的消息模型
progressBlock	Block	图片上传进度主要用于图片和语音
successBlock	Block	发送成功, 返回用于显示状态的消息模型
failBlock	Block	发送失败, 包括内容为空、文件不纯在、网络错误等, 返回用于显示状态的消息模型, 如果消息未发送, 没有模型返回

发送给机器人返回参数:

progressBlock:上传音频、图片进度
 ZCLibMessage:发送对象
 ZCMessageSendCode:
 ZC_SENDMessage_Success:发送成功, 修改接口发送对象的sendStatus=0,
 ZC_SENDMessage_New: 机器人回复对象, 创建了一次新对象(第二次调用successBlock)
 ZC_SENDMessage_FAIL:本次发送消息失败, 对象为空, 可标记发送的对象失败状态

如果ustatus = 5 说明当前用户已超时下线, 当前页面回收键盘, 显示新会话键盘样式

4.6 获取机器人引导语

```
- (void)getRobotGuide:(ZCLibConfig *) config
    robotFlag:(NSString *) robotId
    start:(void(^)(ZCLibMessage *message)) startBlock
    success:(void(^)(ZCLibMessage *message,ZCMessageSendCode sendCode))
successBlock
    fail:(void(^)(ZCLibMessage *message,ZCMessageSendCode errorCode))
failBlock;
```

传入参数示例:

字段名	数据类型	说明
config	ZCLibConfig	ZCLibConfig由初始化接口返回
robotId	String	对接机器人ID，可为空
startBlock	Block	开始请求的回调
successBlock	Block	请求成功的回调
failBlock	Block	请求失败的回调

返回参数：

字段名	数据类型	说明
model	ZCLibMessage	消息对象
ZCMessageSendCode	NSInteger	智齿消息业务状态码

4.7 提交评价

```
-(void)doComment:(NSMutableDictionary *) params;
```

传入参数：

字段名	数据类型	说明
params	NSMutableDictionary	请求参数，其中包含以下字段： cid 会话编号 isresolve 是否解决问题 1没有解决 0 解决 problem 问题编号，没有传空 机器人： 1答非所问 2理解能力差 3 问题不能回答 4不礼貌 人工： 5服务态度差 6回答不及时 7没有解决问题 8不礼貌 source 分数 suggest 描述 type 0 评价机器人 1评价人工 userId 用户Id commentType 评价类型 主动评价 1 邀请评价0

返回参数：

字段名	数据类型	说明
code	int	1请求成功 0 请求失败
status	int	1 发送成功 0 发送失败
msg	String	提示消息

4.8 获取技能组

```
-(void)getSkillSet:(ZCLibConfig *) config
    start:(void (^)(void))startBlock
    success:(void (^)(NSMutableArray *messages,ZCNetWorkCode sendCode)) successBlock
    failed:(void (^)(NSString *errorMessage,ZCNetWorkCode errorCode)) failedBlock;
```

传入参数：

字段名	数据类型	说明
config	ZCLibConfig	当前接口需要ZCLibConfig中uid,appkey
startBlock	Block	开始请求的回调
successBlock	Block	请求成功的回调
failedBlock	Block	请求失败的回调

返回参数：

字段名	数据类型	说明
skillArray	NSMutableArray	技能组数组 存放（ZCLibSkillSet）对象
ZCNetWorkCode	NSInteger	网络请求状态码

4.9 发送正在输入内容

```
-(void)sendInputContent:(ZCLibConfig *) config
    content:(NSString *) content
    success:(void (^)(ZCNetWorkCode sendCode)) successBlock
    failed:(void (^)(NSString *errorMessage,ZCNetWorkCode errorCode)) failedBlock;
```

传入参数：

字段名	数据类型	说明
config	ZCLibConfig	当前接口需要传入ZCLibConfig中uid
content	NSString	用户正在输入的内容
successBlock	Block	请求成功的回调
failedBlock	Block	请求失败的回调

4.10 提交留言

```
-(void)sendMessage:(NSMutableDictionary *) params
            config:(ZCLibConfig *) config
            success:(void (^)(ZCNetWorkCode code,int status,NSString *msg))successBlock
            failed:(void (^)(NSString *errorMessage, ZCNetWorkCode errorCode))failedBlock;

// params 请将邮箱、手机号、用户昵称、留言内容以以下字段放入
NSMutableDictionary * params = [NSMutableDictionary dictionary];
    [params setValue:_emailTf.text forKey:@"customerEmail"];
    [params setValue:_megTextView.text forKey:@"ticketContent"];
    [params setValue:_phoneTf.text forKey:@"customerPhone"];
    [params setValue:_nickNameTf.text forKey:@"customerNick"];
```

传入参数：

字段名	数据类型	说明
params	NSMutableDictionary	customerEmail 用户邮箱 ticketContent 留言内容 customerPhone 用户电话 ncustomerNick 用户昵称
config	ZCLibConfig	需要传入ZCLibConfig中uid 用户邮箱
successBlock	Block	请求成功的回调
failedBlock	Block	请求失败的回调

返回参数：

字段名	数据类型	说明
ZCNetWorkCode	NSInteger	网络请求状态码
status	int	1 发送成功 0 发送失败
msg	String	提示消息

4.11 使用户主动离线

```
-(void)logOut:(ZCLibConfig *) config;
```

传入参数：

字段名	数据类型	说明
config	ZCLibConfig	需要传入ZCLibConfig中uid,cid

4.12 清除历史记录

```
-(void)cleanHistoryMessage:(NSString *) uid
    success:(void(^)(NSData *data)) successBlock
    fail:(void(^)(ZCNetWorkCode errorCode)) failBlock;
```

传入参数：

字段名	数据类型	说明
uid	String	用户ID
successBlock	Block	请求成功的回调
failBlock	Block	请求失败的回调

4.13 留言上传图片

```
-(void)fileUploadForLeave:(NSString *) filePath
    commanyId:(NSString *) commanyId
    start:(void(^)( ))startBlock
    success:(void(^)(NSString *fileURL,ZCNetWorkCode code)) successBlock
    fail:(void(^)(ZCNetWorkCode errorCode)) failBlock;
```

传入参数：

字段名	数据类型	说明
filePath	String	文件路径
commanyld	String	企业编号

4.14 机器人顶踩

```
-(void)rbAnswerComment:(ZCLibConfig *) config
    message:(ZCLibMessage *) commentMessage
    status:(int) status
    start:(void(^)())startBlock
    success:(void(^)(ZCNetWorkCode code)) successBlock
    fail:(void(^)(ZCNetWorkCode errorCode)) failBlock;
```

传入参数：

字段名	数据类型	说明
config	ZCLibConfig	需要获取到ZCLibConfig 的uid、cid、robotFlag
commentMessage	ZCLibMessage	获取ZCLibMessage中 docId 词条ID， docName 词条名称
status	int	反馈结果-顶/踩 1 顶 -1 踩

五、常见问题

5.1启动后页面直接返回

2.0.0的版本将原sysNumber替换成AppKey， 请查看是否传入AppKey正确， 如果没有问题， 请直接联系我们技术人员

5.2 iOS 10相机、相册、麦克风权限问题

```
在info.plist添加NSContactsUsageDescription的key， value中添加相应的描述
<key>NSPhotoLibraryUsageDescription</key>
<string>访问相册</string>
<key>NSCameraUsageDescription</key>
<string>访问相机</string>
<key>NSMicrophoneUsageDescription</key>
<string>访问麦克风</string>
```

5.3 无法正常接收推送消息

- 1.检查PC工作台配置的推送证书P12文件是否和项目bundle id 匹配。
- 2.检查证书线上和测试环境是否和PC工作台配置的相同。
- 3.推送证书的格式是否为P12格式。
- 4.调试是否可以获取token。

5.4 没有收到客服发送的消息

- 1.检查当前手机网络是否有网络。
- 2.是否为离线消息