CRM 使用说明

目录

1.注册账号

访问 https://forward.guyongli.net，点击其中一条线路

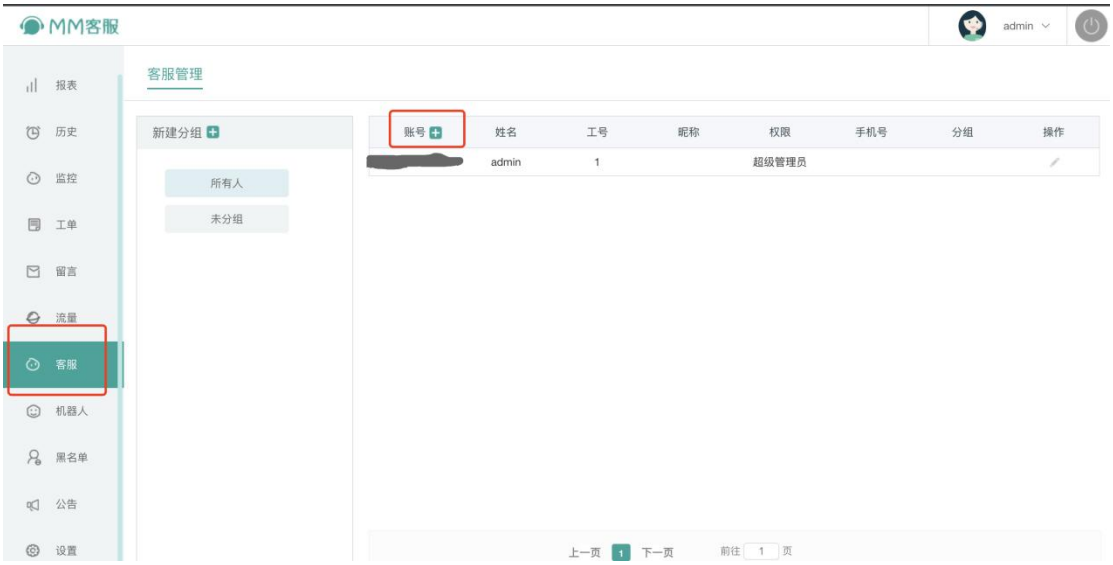
进入 crm 客服系统注册页面，填写相关信息注册公司账号，并激活账号，激活账号需要 CRM 工作人员手动激活。

2.客服系统后台功能简要说明

1、管理员模式和客服模式：

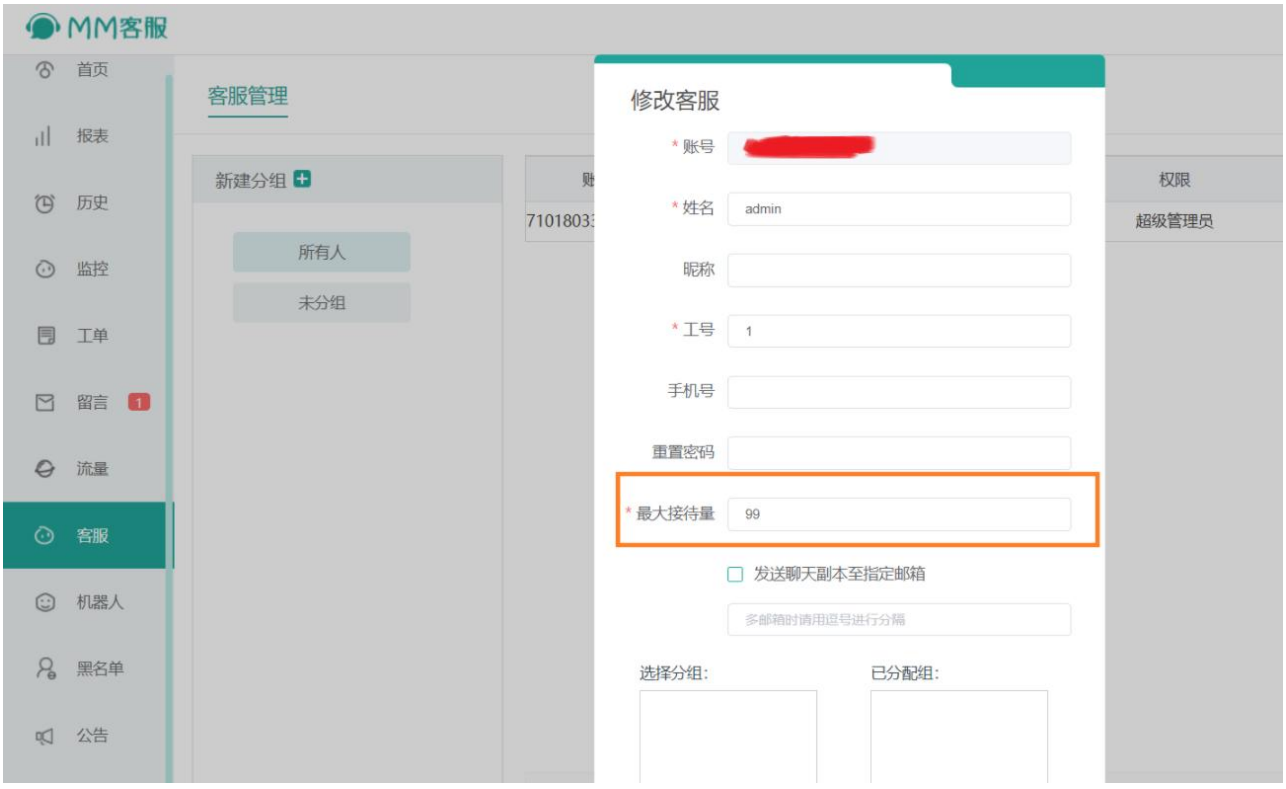
账号模式分管理员账号和客服账号，用邮箱注册的账号具有管理员权限和客服账号权限

使用管理员账号可以创建多个客服账号，如下图所示



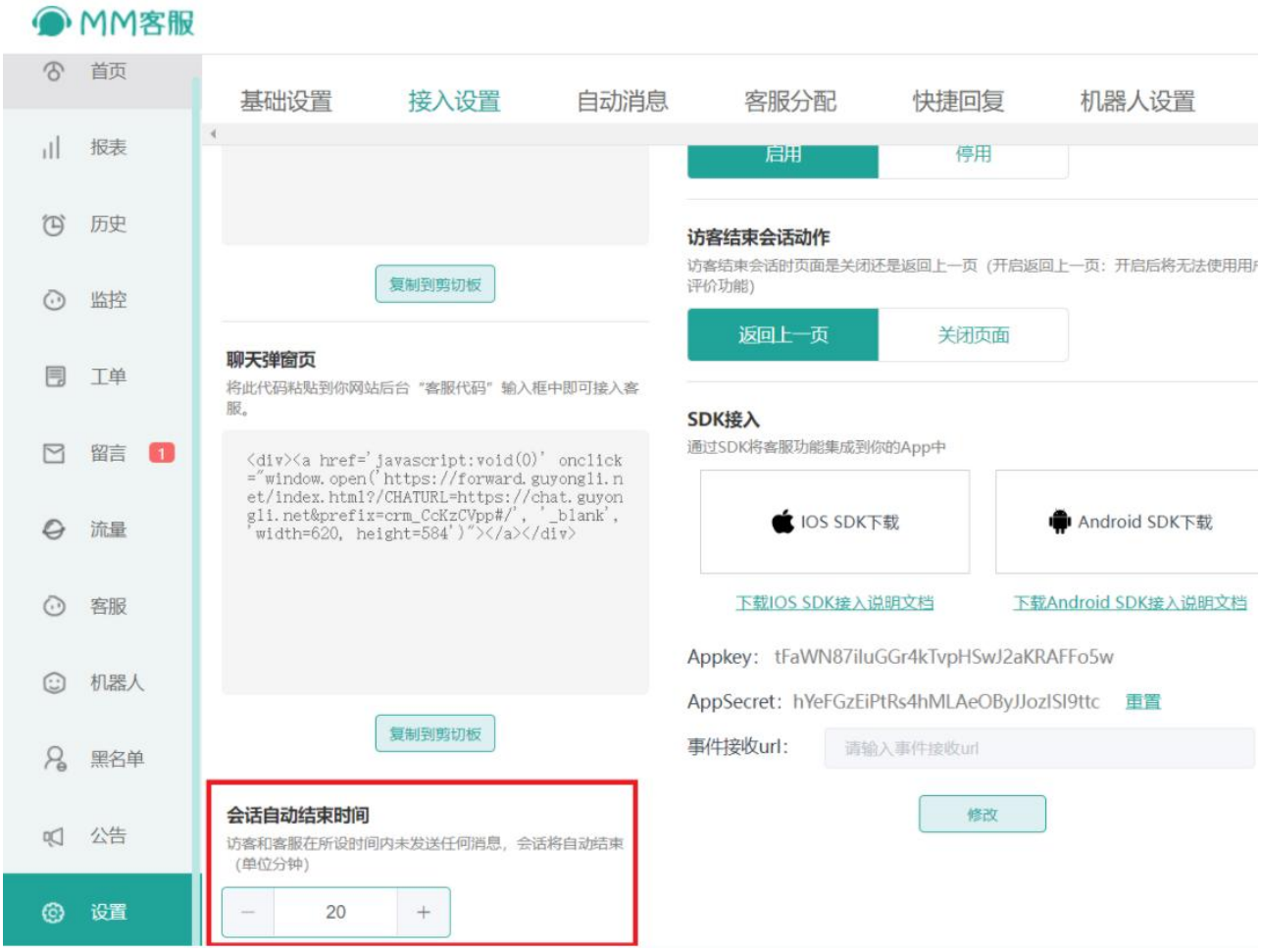
2、管理员模式下设置客服的接待会话数：

每个客服默认接待的用户数为 99，超过 99 个就不接入，这个数值可以实时调整，管理员模式下客服列表，修改客服：(最大接待量的有效数值范围建议为：1 ~ 300)



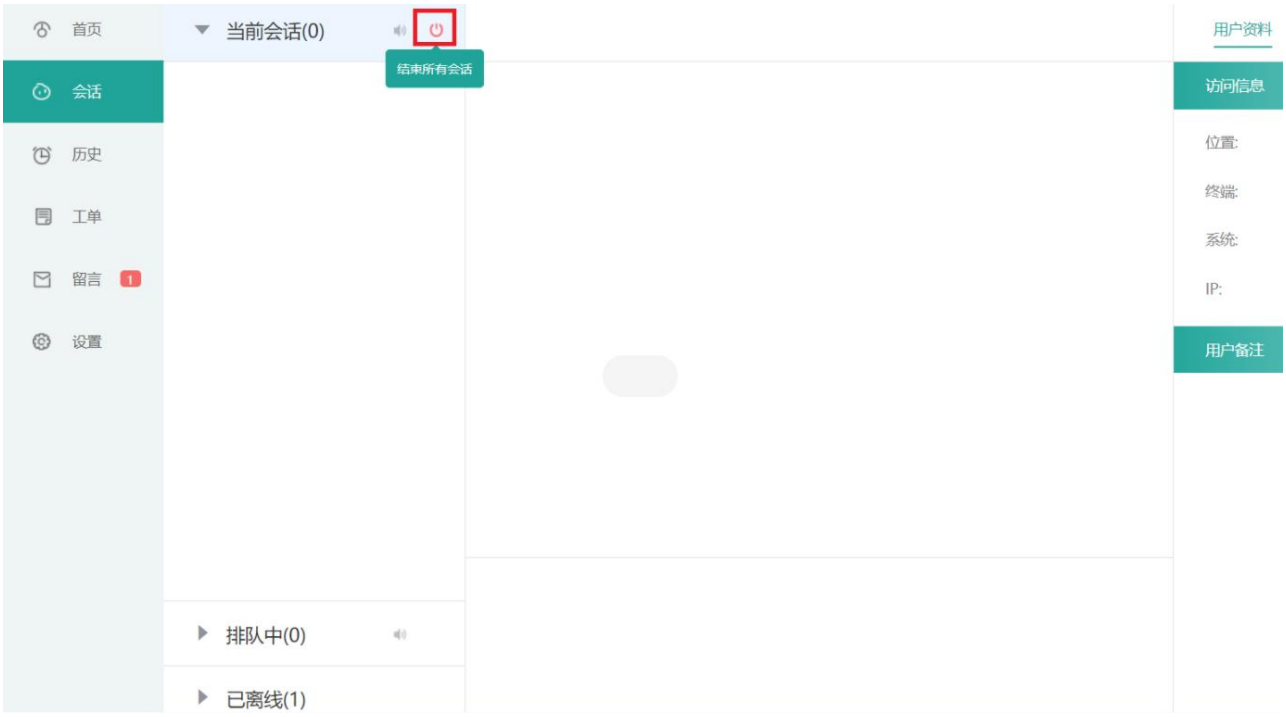
3、管理员模式下设置客服每个会话的空闲存活时间：

当客服和用户没有消息收发时，系统默认 24 小时（1440 分钟)才会结束该会话（用户在这个时间内退出再接入客服且传入同样的 uid 会自动接入该会话），管理员可以调整这个数据，数值的有效范围为 1 ~ 1440，(实时修改该数值，对修改后新接入的会话有作用效果，已经存在的会话还是原来旧数值的存活时间)。



4、客服清理会话：

会话的结束取决于会话空闲存活时间的设置,客服也可以实时手动关闭和结束某个会话(比如实时释放没消息的会话）。　当客服想一键清理结束所有的“当前会话”的时侯，可以使用一键结束当前会话的功能，（比如在客服想清掉当前所有会话）：



3.Web 版接入

3.1 基本使用方法

管理员账号点击设置，接入设置，可以看到聊天超链接和聊天弹窗页两种接入方式。

默认都是访客模式不关联用户信息，如果需要在客服会话中展示用户信息，需要在接入 url 中注入用户信息(customers=用户 id,用户名,语言,vip,vip_level,更多展示)。

用户名如果有“，”特殊字符需要使用\进行转义。

例如: ke,bo 要写为 ke\,bo

语言列表：

语言代码	说明
zh-CN	中文简体
en-US	美式英语
id	印尼
pt-BR	巴西葡萄牙

vip_level：需是正整数，范围是(0 ~ 65535)

更多展示：用户可以传一个任意字符串，长度控制在 500 个字符内，这个字符串需要单独 url_encode 编码一次。

加入 customers 参数效果图 1：（带用户 id 和用户名）

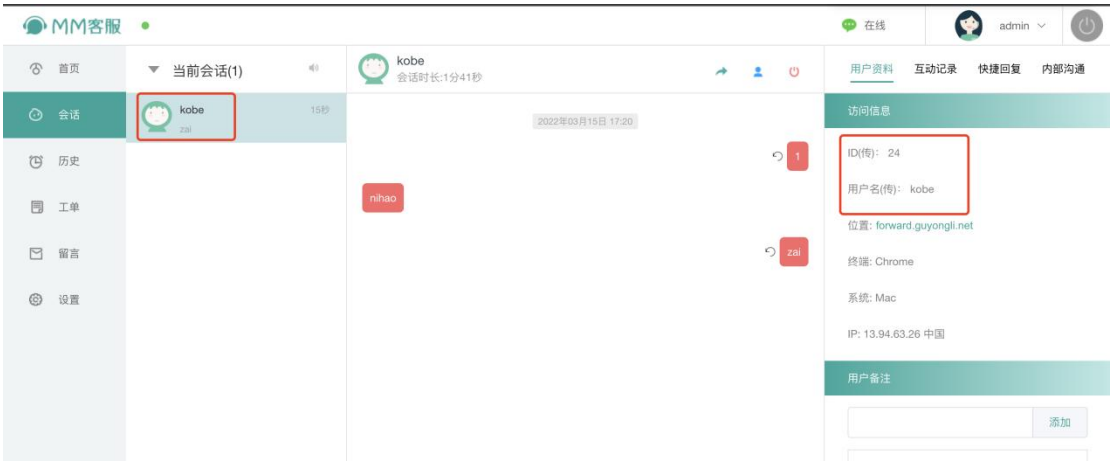
请求 url 形式 1：

https://forward.guyongli.net/index.html?/CHATURL=https://chat.guyongli.net&prefix=crm_T7x9qXwN&customers=24,kobe#/

请求 url 形式 2：（减小跳转次数，现在后台呈现的是这种形式）

https://chat.guyongli.net/index.html?&fileUpload=https://file.guyongli.net&prefix=crm_T7x9qXwN&customers=24,kobe#/chat/crm_T7x9qXwN

这里要注意，不管使用哪一种 url 的形式访问，#及#之后的部分是属于 hash 路由部分，一定要放在整个 url 的最后。（形式 1 中”#/“，形式 2 中”#/chat/crm_T7x9qXwN”部分是 hash 路由）



加入 customers 参数效果图 2：（带用户 id、用户名、语言、vip、vip_level、更多展示）

如，传更多展示的字符为：

test
你好
Hello

url_encode 为：test%0A%E4%BD%A0%E5%A5%BD%0AHello，那最终访问的 url 样式为：（以 url 的形式 1 做示例说明）

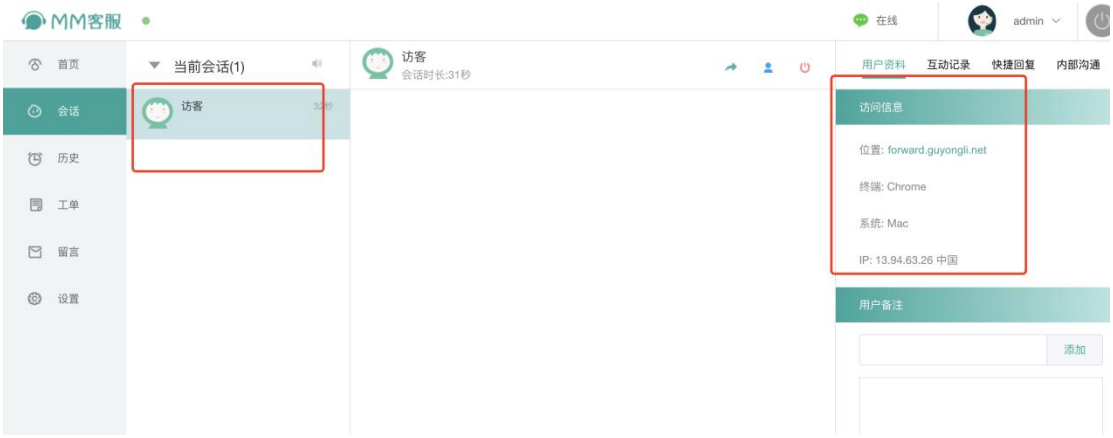
https://forward.guyongli.net/index.html?/CHATURL=https://chat.guyongli.net&prefix=crm_T7x9qXwN&customers=12,21,zh-CN,0,0,test%0A%E4%BD%A0%E5%A5%BD%0AHello#/

注意：自定义字段里想传展示更多的字符串，前面的语言、vip、vip_level 这些都要传。

语言如果不清楚，可以传个 0，vip, vip_level 传到后台，目前也不会显示。*(如果想展示 vip,和 vip_level 字段，可以把值加到更多展示字段里，vip、vip_level 字段随便填写一个 0 值)*

不加入 customers 参数效果图： （以 url 的形式 1 做示例说明）

https://forward.guyongli.net/index.html?/CHATURL=https://chat.guyongli.net&prefix=crm_T7x9qXwN#/



3.2 移动 app 使用 webview(iOS 为 wkwebview)使用注意事项

3.2.1 webview 全屏所引起的问题说明

a. Android 注意事项：

Android 使用 WebView 接入 web_sdk 时可能出现的软键盘弹出相关增加文档：

1. 没有全屏模式的情况下 WebView 装载的 View 会响应虚拟键盘变化。同时也可通过设置 android:windowSoftInputMode 中对应的参数控制界面在虚拟键盘变化时的处理。

2. 全屏模式下 WebView 及其装载的 View 可能无法响应虚拟键盘变化。此时有几种方案可供参考：

a. 关闭全屏及沉浸式(由客户端开发人员设置系统全屏的)模式；(推荐考虑)

手动设置 statusBar 颜色(为了使状态栏的颜色和客服页面上面的颜色一致，可以在后台查看聊天框一的边框一颜色值)。

b. 可以阅读并参考《https://blog.csdn.net/yuyuanhuang/article/details/108324256》提供的解决思路。

b. IOS 注意事项：

ios 的 wkwebview 设成全屏时，系统状态栏会和 webview 分开，可能颜色不太协调。 如果想颜色一致，不要设成全屏，把上面的手机状态栏高度的空间留出来。 可以在后台查看聊天框一的边框一颜色值，然后 app 里把留出来的状态栏高度的空间设为同样的颜色。

3.2.2 关闭问题

a. iOS 和 Android 在关闭 wxwebview 或 webview 的时侯，不能去清理 wxwebview 或 webview 的缓存。如果一定要清理，务必要保留 localstorage 里的缓存数据。不然下次打开客服服务端无法确定是同一个用户，会当一个新用户来看待。
b. iOS 在使用 web 客服系统时，要通过注入一个约定方法（“sendMessageToCocos”），当点击页面关闭时 h5 回调给原生(不然 iOS 会无法关闭页面），方法大致如下：

1.1 配置 WKWebViewConfiguration 中注入” sendMessageToCocos”,具体实现如下

```
WKWebViewConfiguration *config = [[WKWebViewConfiguration alloc] init];
WKUserContentController *wkUController = [[WKUserContentController
alloc] init];
[wkUController addScriptMessageHandler:self name:@"sendMessageToCocos"];
config.userContentController = wkUController;
```

// 在初始化 webView 时设置 WKWebView 的 config 信息为上述 config,实现如下:

```
WKWebView *webView = [[WKWebView alloc] initWithFrame:CGRectMake(0, 20,
self.view.frame.size.width, self.view.frame.size.height - 20) configuration:config];
```

1.2 实现 WKWebView 的 navigationDelegate 代理方法

```
- (void)userContentController:(WKUserContentController *)userContentController

didReceiveScriptMessage:(WKScriptMessage *)message

{

    //h5 点击按钮执行的方法

     if([message.body isEqualToString:@"h5_closeWeb"  ]){

          [self dismissViewControllerAnimated:YES completion:nil];

    }

}
```

c. Android 在使用 web 客服系统时，同 iOS 类似，只是要设一个 windows 的属性名称（“AndroidApp”），方法大致如下：

```
WebView webView = ...init();       // 初始化一个 WebView

WebSettings settings = webView.getSettings();

settings.setJavaScriptEnabled(true);   // 添加 Web 端 JS 可操作方法

// 客户端定义的 JS 方法类(类的定义方式请参考 Android WebView 官方文档注释),例如这里叫 Test, 另

   设一个挂载到 Web js 的 window 上的属性名称" AndroidApp"

webView.addJavascriptInterface(new Test(), "AndroidApp");

// new Test()类中被注解标记的方法将会被 js 调用。 例如命名为: "sendMessageToCocos(string

argument)"。 js 中将会使用 window['AndroidApp'].sendMessageToCocos('h5_closeWeb');来回

调
```

d. H5 端在关闭时回调的代码示例如下：

```
// ios 调用原生关闭

if (window['webkit'] && window['webkit'].messageHandlers) {
```

```
    window['webkit'].messageHandlers.sendMessageToCocos.postMessage('h5_closeWeb');
}
// 安卓调用原生关闭
if (window['AndroidApp'] && window['AndroidApp'].sendMessageToCocos) {
    window['AndroidApp'].sendMessageToCocos('h5_closeWeb');
}
```

4.机器人使用说明

4.1 打开机器人方法

在后台 设置=>机器人设置=>机器人开关 => 启用

机器人启用后，访客第一次进入首先会跟机器人进行会话。

图 4.1 后台打开机器人方法

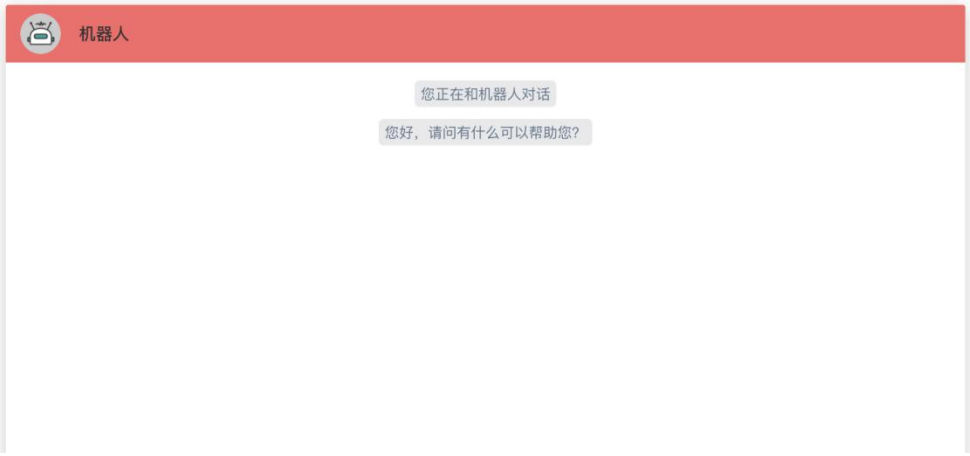


图 4.2 访客端显示机器人会话

4.2 设置机器人知识库

开启了机器人后需要设置机器人知识库，机器人才可以根据相关问题来回答问题。下面是设置机器人知识库的方法：

在机器人 => 知识库 => 添加问题

相似问题可以添加多个，主要的作用是提高问题的匹配度

访客端如果匹配到问题的答案那么就会直接显示对应的答案(图 4.4)，如果没有匹配到答案那么可以转接到人工客服（图 4.5）

图 4.3 后台机器人增加问题

图 4.4 访客端根据提问自动匹配到答案

图 4.5 访客端机器人无法匹配到答案

4.3 机器人问题学习

机器人问题学习的作用主要是统计经常提问的且未匹配到答案的问题。方便我们手动在后台增加对应的答案或者关联已经存在的答案，如图 4.6，注册问题经常被提问，点击右边的关联可以关联到已经存在的问题(如何注册账号)

图 4.6 机器人关联问题学习

5.原生 SDK 接入说明

原生 SDK 下载地址：（以这里的下载链接为准）

iOS SDK 下载地址

release 和 debug 合在一起(可以调试)

https://resource-epository.oss-cn-shanghai.aliyuncs.com/MMSDK.framework_3.zip

纯 release，上 appstore 苹果商店：

https://resource-epository.oss-cn-shanghai.aliyuncs.com/MMSDK.framework_release_3.zip

Android SDK 下载地址（解压包含单独的 debug 版和 release 版）：

https://resource-epository.oss-cn-shanghai.aliyuncs.com/MMSDK_3.zip

5.1 iOS SDK 接入

5.1.1 导入 SDK

MMChat.framework 目前提供一种集成方式，下载见上面，下载后手动将 SDK 集成到您的项目中。

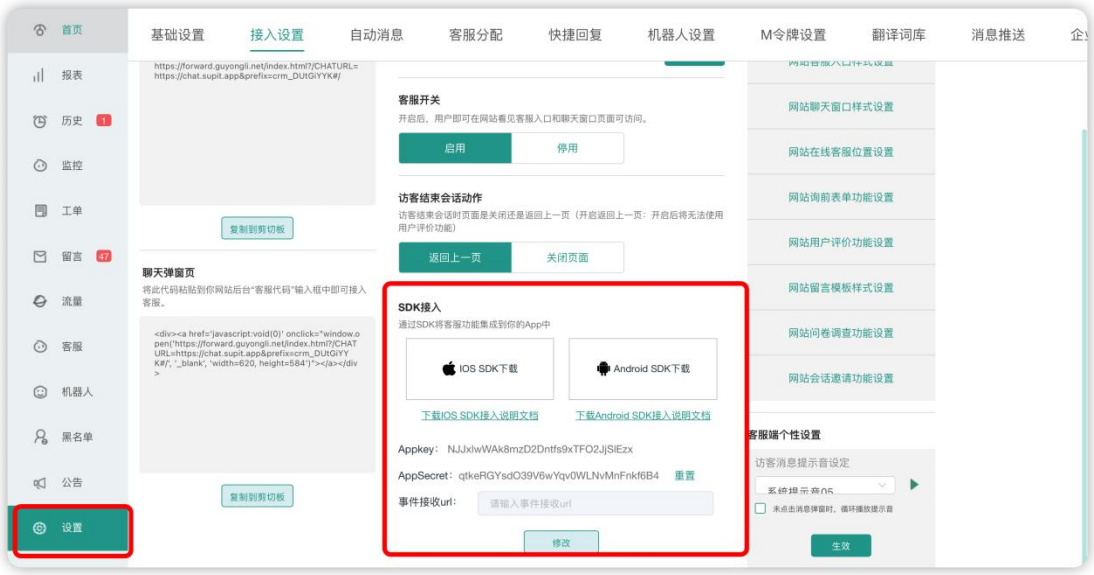


图 5.1 SDK 接入说明文档下载

5.1.2 手动集成 SDK

- 集成 MMChat.framework：将 MMChat.framework 拖入工程，确保文件 copy 至工程而非引用，并确保 General —> Frameworks, Libraries, and Embedded Content (>=Xcode11) 选项中包含导入的库，同时 Embed 选择 **Embed & Sign** 即可。

- 添加 SDK 依赖库（CocoaPods 下载安装）

```
pod 'SKPhotoBrowser', '~> 6.1.0'
pod 'Toast-Swift', '~> 5.0.1'
pod 'SwiftPhoenixClient', '~> 1.3.0'
pod 'Moya', '~> 14.0.0'
pod 'CryptoSwift', '~> 1.3.2'
pod 'EZSwiftExtensions', '~> 2.0'
pod 'Kingfisher', '~> 5.15.6'
```

支持 iOS 系统最低版本 10.0

注意：升级 XCode12.3 报错 Building for , but the linked and embedded framework was built for iOS + iOS Simulator 解决方案 启用 Validate WorkSpace ，让 XCode 对 frameworks 进行自动化管理 打开项目路径 - Build Setting > Build Options > Validate WorkSpace Validate WorkSpace 设置为 true。

5.1.3 权限设置

在 info.plist 中加入权限申请描述：

```
<key>NSPhotoLibraryUsageDescription</key>
<string>需要读取相册权限</string>
<key>NSPhotoLibraryAddUsageDescription</key>
<string>需要写入相册权限</string>
<key>NSCameraUsageDescription</key>
<string>需要相机权限</string>
<key>NSMicrophoneUsageDescription</key>
<string>需要麦克风权限</string>
```

5.1.4 https 相关

在 info.plist 中加入以下内容：

```
<key>NSMicrophoneUsageDescription</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
        <key>NSAllowsArbitraryLoadsInWebContent</key>
    <true/>
</dict>
```

5.1.5 类库说明

SDK 主要提供以下类/协议/方法：

MMChatManager-会话管理类：负责会话流程及消息监听、消息事件处理。

AppInfo-接入的客服 app 信息类:用于 SDK 接入,对 app 验证授权。

User-接入的客服 app 的用户信息类:用户信息，包含 ID 及详细信息。

5.1.6 初始化 SDK

1、在使用 SDK 任何方法之前，都应该先调用初始化方法。正常业务情况下，初始化方法有且只应调用一次，请勿重复注册。

注册方法：

```
    /// 注册 app
    /// \param app appInfo
    /// \param completion BOOL 是否注册成功，服务器维护通知，失败 error
    - (void)registerAppWithInfo:(MMAppInfo * _Nonnull)info :(void (^ _Nullable)(BOOL,
NSDictionary<NSString *, id> * _Nullable, NSError * _Nullable))completion;
```

建议在 app 启动时进行初始化操作：

```
    - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary
*)launchOptions {
        // 注册 MMChat SDK
        MMAppInfo *appinfo = [MMAppInfo.alloc init:@{@"prefix":@"your prefix ",@"app_key":@"your
app_key", @"auth_token":@"your app auth_token", @"app_id":@"your app_id", @"app_secret":@"your
app_secret"}];
        MMChatManager *manager = MMChatManager.sharedInstance;
        [manager registerAppWithInfo:appinfo :^(BOOL isSuccess, NSDictionary<NSString *,id>
*maintainInfo, NSError *error) {

        }];

        return YES;
    }
```

添加消息推送 token:

```
    - (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken
    {
        NSString *pushToken = [[[[deviceToken description]
                                 stringByReplacingOccurrencesOfString:@"<" withString:@""]
                                stringByReplacingOccurrencesOfString:@">" withString:@""]
                               stringByReplacingOccurrencesOfString:@" " withString:@""];

        if ([UIDevice currentDevice].systemVersion.floatValue >= 13.0) {
            pushToken = [AppDelegate getHex:deviceToken];
            MMAppInfo *info = MMChatManager.sharedInstance.appInfo;
            info.device_token = pushToken;
            MMChatManager.sharedInstance.appInfo = info;
            NSLog(@"device_token=%@", pushToken);
        }
    }
    +(NSString *)getHex:(NSData *)data
```

```
    {
        NSUInteger len = data.length;
        char *chars = (char *)[data bytes];
        NSMutableString *hexString = [NSMutableString new];
        for (int i = 0; i < len; i++) {
            NSString *str = [NSString stringWithFormat:@"%0.2hhx", chars[i]];
            [hexString appendString:str];
        }
        return hexString;
    }
```

接入客服聊天界面:

```
    - (IBAction)goToMMChat:(id)sender {
        MMChatManager *manager = MMChatManager.sharedInstance;
        // 连接的用户信息，详细的 user  根据 MMUser 属性进行赋值即可。
        MMUser *user = [MMUser.alloc init:@{@"uid":@"iOS008", @"uname":@"iOS008_uname",
@"name":@"iOS008_name"}];

        [manager readyConnect:user :^(BOOL isReady, NSError * _Nullable readyError) {
            if (isReady) {
                [manager connectToChat:^(BOOL connnected, NSError * _Nullable connnectError) {
                    if (connnected) {
                        // 成功接入 MM 客服

                    } else {




                    }
                }];
            } else {

            }
        }];
    }
```

每次的聊天接入必须在确保 readyConnect 返回 true 后才可 connect。

5.1.7 SDK 强制竖屏

IOS SDK 的横竖屏默认是跟随应用 APP 的设置，如需强制设置 SDK 竖屏,请在 AppDelegate.h 中声明 allowRotation, @property (nonatomic, assign) BOOL allowRotation (默认值为 false)，在 AppDelegate.m 中实现下面方法:

```
-(UIInterfaceOrientationMask)application:(UIApplication *)application
supportedInterfaceOrientationsForWindow:(UIWindow *)window
{
    if (self.allowRotation) {
        return UIInterfaceOrientationMaskPortrait;
    }else {
        // APP 需要使用的显示方向，是横屏
        return UIInterfaceOrientationMaskLandscapeRight;
    }
}
```

设定 MMChatManager 中的属性 mastUsePortrait 的值为 True，并在 manager.connectToChat 成功后，设置 AppDelegate 中的 allowRotation 值为 True，同时实现 MMChatManager 中 closeCallBack 闭包，在闭包中将 Appdelegate 中 allowRotation 设值为 False。

5.1.8 关闭 SDK 界面平滑效果时间(延时)设置

关闭 SDK 前先回调 closeCallBack，请调用 setAfterDelayDismissWithValue 方法，此方法传入参数是提前多少秒调用 closeCallBack，默认 0.5 秒，如不需要提前回调，请设定值为 0

5.1.9 提交苹果商店审核注意

1.在打包时替换 frameWork 包为纯 Release 版，否则会出现 contains unsupported architectures

‘[x86_ 64, i386]错误

2.删除第三方库 EZSwiftExtensions 中包含的 UIWebView,删除方式一，直接删除

EZSwiftExtensions 库下面的 BlockWebView.swift(每次执行 PodInstall 后都需要手动删除)，删除

方式二，在 Podfile 中添加以下脚本:

```
pre_install do |installer|
  puts 'pre_install begin....'
  dir_ezswift = File.join(installer.sandbox.pod_dir('EZSwiftExtensions'), 'Sources')
  Dir.foreach(dir_ezswift) {|x|
      real_path = File.join(dir_ezswift,x)
      if (!File.directory?(real_path) && File.exists?(real_path))
        if(x == 'BlockWebView.swift')
          File.delete(real_path)
          puts 'delete:'+ x
        end
      end
  }
  puts 'end pre_install.'
end
```

5.2 Android SDK 接入

5.2.1 客服系统账号

1、获得 公司 ID、 Appkey 、AppSecret

公司 ID、Appkey 、AppSecret SDK 连接服务端的身份凭证，可登录 MM 客服管理后台在 "设置" -> "接入设置" 配置界面获取。

(可参考图 5.1)

2、填写对应平台的推送信息(非必需)

为了使您的 APP 在集成本 SDK 后具有离线消息推送，建议在设置-> 消息推送->添加 APP，填写您的 App 名称、包名称、项目编号、密钥等信息。

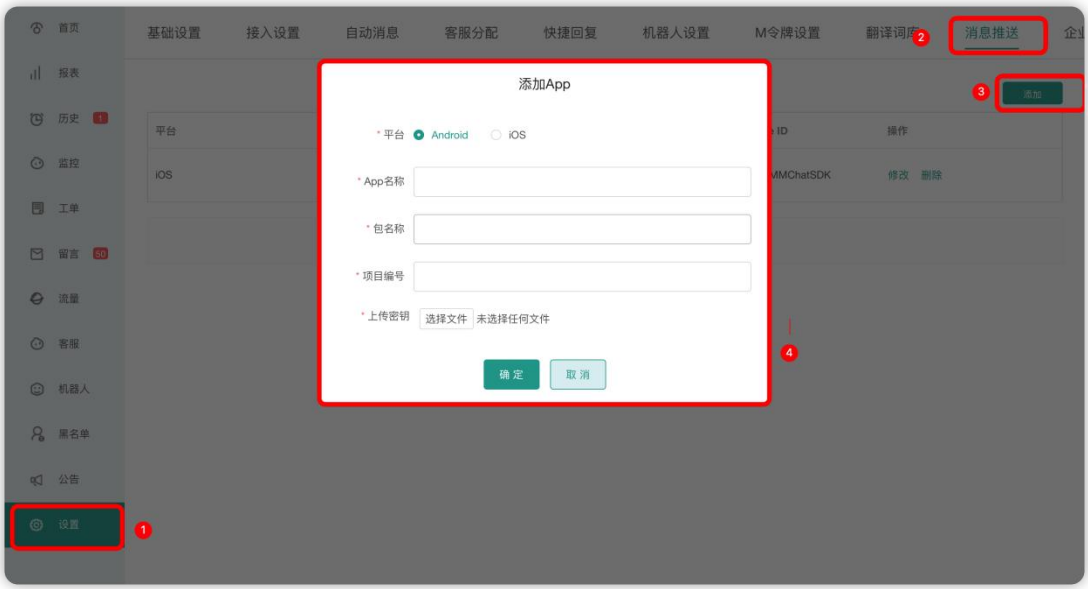


图 5.3 推送设置

3、下载 SDK：见上面

4、环境要求

在您集成 MM 客服 SDK 前环境要求如下:

- Android SDK `Build-tools` 请升级到 21 及以上版本

```
- JAVA 编译版本 `JDK 1.8` 及以上版本

- Android SDK 最低支持 Android API 21: Android 5.0(Lollipop)

- SDK 兼容包内使用了 Android X，不支持 Android Support
```

5.2.2 SDK 接入

1、 添加 SDK 到项目中

mmkf-sdk-v1.0.0.aar 拷贝到项目 libs 下面

在工程 build.gradle 文件中添加对应依赖：

```
    repositories {
        flatDir {
            dirs '../app/libs' //就是你放 aar 的目录地址
        }
    }
```

在应用的 App 模块的 build.gradle 里的 dependencies 中添加 SDK 对应的依赖：

```
    compile(name: 'mmkf-sdk-v1.0.0', ext: 'aar')
    //mmsdk 中需要引入的，以下依赖 如果本来已经有了就不用添加
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.5.1'  // or downgrade -> 2.2.0
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.5.1'  // or downgrade -> 2.2.0
    implementation 'androidx.recyclerview:recyclerview:1.2.1'  // or downgrade -> 1.1.0
    implementation 'com.fasterxml.jackson.core:jackson-databind:2.13.2' // or downgrade -> 2.8.3
    implementation 'com.squareup.okhttp3:okhttp:3.12.1'  // or downgrade -> 3.6.0
    implementation 'com.alibaba:fastjson:1.2.73'
    implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0'
    implementation 'com.github.CymChad:BaseRecyclerViewAdapterHelper:3.0.4'
    implementation 'com.github.bumptech.glide:glide:4.11.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'
    implementation 'org.greenrobot:eventbus:3.2.0'
    implementation 'top.zibin:Luban:1.1.8'
```

注意：kotlin 版本接入时 jackson 需要额外添加:

com.fasterxml.jackson.module:jackson-module-kotlin， 如：版本号 2.9.6

com.fasterxml.jackson.core:jackson-core， 如：版本号为 2.9.6

这个版本号需要根据项目的 sdk 及 gradle 版本来调试确定，如果版本不对，大部分会编译不通过，如果编译通过且能正常使用即可。

2、在你的 App 的合适页面添加客服入口按钮，并在响应函数中加入如下代码：

```
/**
     * prefix:公司 ID 在客服管理平台->设置->接入设置可以查看到
     * appId:接入的 AppID 客服管理平台->设置->消息推送 中添加接入接入的应用后可以查看到
     * appKey:appKey 在客服管理平台->设置->接入设置可以查看到
     * AppSecret:在客服管理平台->设置->接入设置可以查看到
     * authToken:接入的应用的授权 Token，没有时不传
     * uid:接入的应用的 UUID,没有账号系统时时不传
     * device_token:FCM 推送的设备 token
     */
    private void startChatActivity() {
        Bundle bundle = new Bundle();
        bundle.putString("prefix","crm_R2u6TmLx");
        bundle.putString("appId","2a6fc13c-e48e-4c43-aa22-854f4a1d5c32");

bundle.putString("device_token","fXkNLOfNFQ8:APA91bER3dhBAPoGGkGbX67uCLZ58EQQqKzUgEo2RD5RGgTDI-zHo-3bQNDJzX2LggNiQ_etbiOnhyxcm98tWp8qjO4hCZ6oguxE6MkGp2wWp79EpUxS6pvnzAXWIFQim-iWpIFdX3a5");
        bundle.putString("AppSecret","VpNoboqbiKdAjxg3mBC5DrvwL85kwQEB");
        bundle.putString("appKey","NxuqCWDHuQjWyl8mEDW9KJ3OlFh8C6Vg");

        MMClientAgent.getInstance().startMMChatActivityWithBundle(getApplicationContext(), bundle);
    }
```

5.2.3 SDK 包具体内容

```
    sdk
    ├── libs
    │    ├── mmkf-sdk-v1.0.0.aar
    ├── Demo
    │    └── ***
    └── MM 客服 Android SDK 接入文档
```

上面文件中, mmkf-sdk-v1.0.0.aar, 是 MM 客服的 SDK 包, Demo 为接入了 SDK 的 DEMO 工程,MM 客服 Android SDK 接入文档为接入的文档。

5.2.4 混淆配置

如果你的 apk 最终会经过代码混淆，请在 proguard 配置文件中加入以下代码：

```
-dontwarn com.mm.sdk.**
-keep class com.mm.sdk.** {*;}
```

5.2.5 注意事项

Android 10 分区权限变更适配，在 androidmanifest.xml 中的 application 标签中入：

```
android:requestLegacyExternalStorage="true"
```

5.3 原生 SDK 参数使用说明

（iOS 和 Android 原生的 SDK 简称原生 SDK，web 版的 SDK 简称 web SDK，web SDK 的参数使用见上面 Web 版接入）

键	类型	值	说明
prefix	字符串	注册时生成的固定值	必传，唯一标识的固定 token
app_key	字符串	后台自己配	可选
app_secret	字符串	后台自己配	可选
app_id	字符串	后台自己配	可选，某款 APP 的 id 标识。
device_token	字符串	后台自己配	可选，和 app_id 是一一对应的。
auth_token	字符串		可选，每次登录授权的 token
uid	字符串	用户 uid	1、可选 2、使用时需带有 app_key、app_secret 和 auth_token 这三个参数。 *(目前 auth_token*

			的值和 app_secret 一样，后台的设置--接入设置--事件接收 url 不要填值）
uname	字符串	用户昵称	可选
name	字符串	名字	可选。这个字段可以用来扩展使用，在 name 里可以传任意的字符串（长度最好控制在 4096 字符），里面可以带回车，后台会原样展示。
vip	字符串	vip 相关的信息	可项
vip_level	字符串	vip 等级	1、可选 2、这个字符串须是正整数类型的字符串，如“0”，“1”，“99”，” 65535”，不能超过 65535
email	字符串	邮箱	可选
phone	字符串	电话号码	可选
icon	字符串	头像地址	可选
sex	字符串	男或女	可选
reg_date	字符串		可选，格式："2020-10-17 15:30:10"
last_login_time	字符串		可选，格式："2020-10-17 15:30:10"

后台目前可显示传递参数为 uid，uname，name 这三个参数，其它参数目前是预留的。如果想显示更多的信息，可以添加在 name 这个字段里面。
比如在 name 字段里添加如下字符“test\r\n 你好\r\nHello”

在客服端聊天聊界面，访问信息栏，点开更多，显示如下：

6.语言支持

当前支持的语言种类，根据手机当前的系统语言会自动切换

英语
葡萄牙语
简中
俄语
法语/法语(加拿大)
德语
意大利语
日语
韩语

阿拉伯语
丹麦
芬兰
希腊
马来语
泰语
越南语
西班牙语
捷克
荷兰
希伯来语
匈牙利语
挪威语
波兰语
瑞典语
土耳其语
加泰罗尼亚语
克罗地亚语
印地语
印尼语
罗马尼亚语
斯洛伐克语

乌克兰语

7、Demo 程序

iOS 下载地址：

https://resource-epository.oss-cn-shanghai.aliyuncs.com/OC-MMChatSDK.zip

Android 下载地址：

https://resource-epository.oss-cn-shanghai.aliyuncs.com/mmkf_demo.zip