上海久科有限公司

Saas管理需求说明书

上海久科信息技术有限公司

2016年9月18日

作者：产品事业部

**保 密**

本文件及其内容为上海久科信息技术有限公司保密资料。本文件及其内容永远属于上海久科信息技术有限公司拥有财产，在无上海久科信息技术有限公司采用书面形式表示同意外，本文件

1. 不可以用于其他非原指定用途；
2. 不可以向第三方披露；
3. 不可采用任何形式进行复制和传播，包括但不限于传真、纸张邮件和电子邮件。

**文档修改历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **版 本** | **修改人** | **日 期** | **描 述** |
| V1.0 | Sara | 20160918 | 版本创建 |
|  |  |  |  |

目录

[1 概述 6](#_Toc468568598)

[1.1 本文目的 6](#_Toc468568599)

[1.2 适用范围 6](#_Toc468568600)

[1.3 术语与缩略词 6](#_Toc468568601)

[1.4 运行环境 6](#_Toc468568602)

[1.5 参考文档 6](#_Toc468568603)

[2 功能列表 7](#_Toc468568604)

[3 总体说明 9](#_Toc468568605)

[3.1 SDK API说明 9](#_Toc468568606)

[3.1.1 设置客户信息 9](#_Toc468568607)

[3.1.2 修改客户信息 10](#_Toc468568608)

[3.1.3 设置用户设备唯一标识 10](#_Toc468568609)

[3.1.4 设置Visitor上线 10](#_Toc468568610)

[3.1.5 获取当前客户的visitorID 11](#_Toc468568611)

[3.1.6 绑定当前客户的visitorID 并设置上线 11](#_Toc468568612)

[3.1.7 创建一个新的客户 11](#_Toc468568613)

[3.1.8 指定客服或者业务类型 12](#_Toc468568614)

[3.1.9 发送文字/图片/语音消息 12](#_Toc468568615)

[3.1.10 发送轨迹消息 13](#_Toc468568616)

[3.1.11 输入预知 13](#_Toc468568617)

[3.1.12 接收即时消息 14](#_Toc468568618)

[3.1.13 获取服务器聊天数据 16](#_Toc468568619)

[3.1.14 获取客户本地聊天数据 16](#_Toc468568620)

[3.1.15 获取未读消息数量 16](#_Toc468568621)

[3.1.16 获取未读消息 17](#_Toc468568622)

[3.1.17 删除客户聊天数据 17](#_Toc468568623)

[3.1.18 结束当前对话 18](#_Toc468568624)

[3.1.19 获取当前正在接待的客服信息 18](#_Toc468568625)

[3.1.20 更新阅读状态 18](#_Toc468568626)

[3.1.21 获取SDK版本号 18](#_Toc468568627)

[3.1.22 设置日志模式 19](#_Toc468568628)

[3.2 快速集成SDK 19](#_Toc468568629)

[3.2.1 初始化SDK 19](#_Toc468568630)

[3.2.1 启动对话界面 20](#_Toc468568631)

[3.2.1 留言 21](#_Toc468568632)

[3.2.1 满意度评价 22](#_Toc468568633)

[3.3 消息推送 23](#_Toc468568634)

[3.3.1 关闭服务 23](#_Toc468568635)

[3.3.2 开启服务 23](#_Toc468568636)

[3.3.3 推送消息数据结构 24](#_Toc468568637)

# 概述

## 本文目的

为了让企业app快速拥有在线客服功能而提供的一套应用程序接口。通过该接口可以实现人工在线客服和智能机器人客服，以及为app用户提供帮助中心文档。

## 适用范围

本文档适用于上海久科关于Saas需求的开发人员和测试人员。

## 术语与缩略词

|  |  |
| --- | --- |
| **缩略语** | **描述** |
|  |  |
|  |  |
|  |  |

## 运行环境

硬件环境：

CPU：2.0GHz及以上  
内存：2G及以上

硬盘：80G及以上

软件环境：

服务器：支持Windows Service 2008以上版本

## 参考文档

|  |  |
| --- | --- |
| **序号** | **参考文档** |
|  |  |
|  |  |
|  |  |

# 功能列表

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **子功能** |
| 初始化 | 初始化SDK，获取设备标识 | SDK初始化获取APPKey和公司ID，获取用户的设备标识（DeviceToken） |
| 初始化客户信息（创建/更新） | 开发者可以上传客户系统字段信息，若已存在，则更新；否则创建客户信息。供客服查看。 |
| 聊天 | 聊天 | 客户点击“在线客服”打开聊天窗口 |
| 获取机器人设置 | 获取服务端的机器人设置 |
| 客户排队信息 | 获取客户排队消息 |
| 根据业务类型分配客服 | 根据业务类型进行分配客服 |
| 发送消息 | 支持文本、表情、图片、语音 |
| 输入预知 | 将用户正在输入的内容，实时显示在客服对话窗口，每500毫秒发送一次消息 |
| 获取未读消息 | 获取所有的未读消息，用户可以在需要显示未读消息数是调用此接口，此接口会自动判断并合并本地和服务器上的未读消息，当用户进入聊天界面后，未读消息将会清零 |
| 系统提示语 |  |
| 满意度评价 | 满意度评价 | 客服推送满意度评价或客服关闭对话；客户填写后，提交满意度评价 |
| 留言 | 留言 | 当前客服繁忙或者不在线，输入内容发送消息，弹出留言提示，如果客户点击则跳转到表单界面 |
| 其他 | 消息推送 | 开发者可上传 App 的推送证书；在 App 进入后台时，应该通知服务端，让其将以后的消息推送给APP；在 App 进入前台时，应该通知服务端，让其将以后的消息发送给SDK |
| 消息回执 | 客户发送消息后，要检测客服发送的回执消息。 |

图2-1 功能列表



图2-2 工作流程

# 总体说明

## SDK API说明

### 设置客户信息

开发者可以上传客户系统字段信息，若已存在，则更新；否则创建客户信息生成VisitorID（**字母 / 数字**），即认为该客户已上线。开发者可以直接传入这些用户信息，供客服查看。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| VisitorInfo | {  "nickname ": "sara",  "gender ": "1",  "name ": "李静",  "tel ": "13000000000",  "qq": "474760818",  "email": "sara.li@9Visitor.com",  "address": "上海松江",  "company": "上海久科 "  } | nicknam：昵称  gender：性别 0男 1女  name：李静  tel：联系电话  qq：QQ  email：邮箱  address：地址  company：公司 |
| onVisitorInfoCallback |  | 回调 |

样例：

|  |
| --- |
| setVisitorInfo(Map<String, String> VisitorInfo, OnVisitorInfoCallback onVisitorInfoCallback)  实例：  //创建自定义信息  HashMap<String, String> VisitorInfo = new HashMap<>();  VisitorInfo.put("nickname", "sara");  VisitorInfo.put("gender", "女");  VisitorInfo.put("name", "李静");  VisitorInfo.put("tel", "1300000000");  VisitorInfo.put("qq", "474760818");  MQManager.getInstance(context).setVisitorInfo(info, new OnVisitorInfoCallback()); |

客户信息包括昵称（必填）、性别、姓名、手机、QQ、邮箱、地址、公司。视后台客户信息字段及属性而定。

客户信息包括以下字段：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名称 | 字段名称 | 类型 | 值范围 | 提示语 | 备注 |
| nickname | 昵称 | 文本框 | 非空 | 请输入昵称 |  |
| gender | 性别 | 单选框 | 非空 |  |  |
| name | 姓名 | 文本框 |  |  |  |
| tel | 手机号 | 文本框 | 唯一 |  | 手机号格式验证 |
| qq | QQ | 文本框 |  |  | QQ格式验证 |
| email | 邮箱 | 文本框 |  |  | 邮箱格式验证 |
| address | 地址 | 文本框 |  |  |  |
| company | 公司 | 文本框 |  |  |  |

### 修改客户信息

样例：

|  |
| --- |
| updateVisitorInfo (Map<String, String> updateInfo, OnVisitorInfoCallback onVisitorInfoCallback)  实例：  //创建自定义信息  HashMap<String, String> updateInfo = new HashMap<>();  updateInfo.put("nickname", "jade newName");  UCCManager.getInstance(this).updateVisitorInfo(updateInfo, OnVisitorInfoCallback()); |

### 设置用户设备唯一标识

设置用户的设备唯一标识

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| deviceToken | 9774d56d682e549c | 设备唯一标识 |

样例：

|  |
| --- |
| registerDeviceToken(String deviceToken, OkHttpUtils.OnRegisterDeviceTokenCallback onRegisterDeviceTokenCallback) |

### 设置Visitor上线

初始化 SDK 成功后，会默认生成一个顾客，如果没有更改过顾客 id，将以默认生成的顾客上线。

设置当前Visitor上线

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| onlineCallback |  | 回调 |

样例：

|  |
| --- |
| setCurrentVisitorOnline(final OnVisitorOnlineCallback onlineCallback) |

### 获取当前客户的visitorID

获取当前客户的visitorID，开发者可保存该客户id，下次使用 setVisitorOnlineWithUCCVisitorId 接口来让该顾客登陆客服系统。

参数样例

|  |  |  |
| --- | --- | --- |
| **返回参数名** | **参数值eg** | **备注** |
| visitorID |  | 当前顾客 id |

样例：

|  |
| --- |
| getCurrentVisitorId() |

### [绑定当前客户的visitorID 并设置上线](http://meiqia.com/docs/meiqia-android-sdk/#绑定美洽-id-并设置上线)

开发者可通过[3.4.1](#_获取当前客户的visitorID)接口，取得顾客 id ，保存到开发者的服务端，以此来绑定客户和开发者用户系统。如果开发者保存了UCC的客户visitorID，可调用如下接口让其上线。调用此接口后，当前可用的客户即为开发者传的客户visitorID。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| visitorID |  | 当前顾客 id |
| onlineCallback |  | 回调 |

样例：

|  |
| --- |
| setVisitorOnlineWithVisitorId(String visitorID, final OnVisitorOnlineCallback onlineCallback) |

### 创建一个新的客户

如果开发者想初始化一个新的客户，可调用此接口。

该顾客没有任何历史记录及用户信息。

获取一个新的客户

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| onGetUCCVisitorIdCallBack |  | 回调 |

样例：

|  |
| --- |
| createUCCVisitor(OnGetUCCVisitorIdCallBackOn onGetVisitorVisitorIdCallBack) |

### [指定客服或者业务类型](http://meiqia.com/docs/meiqia-android-sdk/#指定客服或者分组)

（需要ID查询模块，否则开发者无法获取）

默认会按照后台设置的分配方式进行智能分配客服，但如果开发者需要将来自 App 的客户指定分配给某个客服、某个业务类型下的客服。

在创建客户之后，调用此接口可主动指定分配客服或客服组并自动打开人工客服界面。

指定客服或业务类型接口一次只能用一个

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| agentId | 非空 | 客服id |
| groupId | 非空 | 业务类型id |
| scheduleRule |  | 分配规则 |

样例：

|  |
| --- |
| setScheduledAgentOrGroupWithId(String agentId, String groupId, UCCScheduleRule scheduleRule) |

分配规则：

|  |
| --- |
| UCCScheduleRule.REDIRECT\_NONE // 指定分配客服失败，则进入留言 |
| UCCScheduleRule.REDIRECT\_GROUP // 分配给业务类型下的客服，分配失败，则进入留言 |

**注意：**

该选项需要在用户上线前设置。

### [发送文字/图片/语音消息](http://meiqia.com/docs/meiqia-ios-sdk/#发送消息)

开发者调用此接口来发送文字消息：

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| content |  | 消息内容 |
| onMessageSendCallback |  | 消息状态回调 |

样例：

|  |
| --- |
| sendUCCTextMessage(String content, final OnMessageSendCallback onMessageSendCallback) |

开发者调用此接口来发送图片消息：

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| localPath |  | 图片的本地路径 |
| onMessageSendCallback |  | 消息状态回调 |

样例：

|  |
| --- |
| sendUCCPhotoMessage(String localPath, final OnMessageSendCallback onMessageSendCallback) |

开发者调用此接口来发送语音消息：

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| localPath |  | 语音的本地路径 |
| onMessageSendCallback |  | 消息状态回调 |

样例：

|  |
| --- |
| sendUCCVoiceMessage(String localPath, final OnMessageSendCallback onMessageSendCallback) |

注意，调用发送消息接口后，回调中会返回一个消息实体，开发者可根据此消息的状态，来判断该条消息是发送成功还是发送失败。

### [发送轨迹消息](http://meiqia.com/docs/meiqia-ios-sdk/#发送消息)

开发者调用此接口来发送客户轨迹消息：

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| VisitorTrack | {  "title": "test\_track1",  "price": "¥235",  "desc": "假两件衬衣+V领毛衣上衣",  "imgUrl": "<http://o8ugkv090.bkt.clouddn.com/em_three.png>",  "itemUrl": "[http://www.baidu.com](http://www.baidu.com/)"  } | title：显示标题  price：显示价格  desc：描述  imgUrl：显示图片  itemUrl：点击跳转链接 |
| onMessageSendCallback |  | 消息状态回调 |

样例：

|  |
| --- |
| sendUCCTrackMessage(Map<String, String> VisitorTrack, final OnMessageSendCallback onMessageSendCallback) |

### 输入预知

将用户正在输入的内容，实时显示在客服对话窗口，提供给客服查看。该接口没有调用限制，但每1秒内只会向服务器发送一次数据）。

注意：需要在初始化成功后，且客服是在线状态时调用才有效。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| content |  | 正在输入的内容 |
| onMessageSendCallback |  | 消息状态回调 |

样例：

|  |
| --- |
| sendVisitorInputtingWithContent(String content) |

### 接收即时消息

在未开启[消息推送](#_消息推送)的情况下，开发者可以通过注册一个 BroadcastReceiver ，监听广播

注意：必须通过 LocalBroadcastManager 注册 和 取消注册 BroadcastReceiver。

注册样例：

|  |
| --- |
| LocalBroadcastManager.getInstance(this).registerReceiver(messageReceiver, intentFilter); |

取消注册样例：

|  |
| --- |
| LocalBroadcastManager.getInstance(this).unregisterReceiver(messageReceiver); |

BroadcastReceiver：

|  |
| --- |
| public class MessageReceiver extends BroadcastReceiver {  @Override  public void onReceive(Context context, Intent intent) {  // 获取 ACTION  final String action = intent.getAction();  // 接收新消息  if (UCCMessageManager.ACTION\_NEW\_MESSAGE\_RECEIVED.equals(action)) {  // 从 intent 获取消息 id  String msgId = intent.getStringExtra("msgId");  // 从 MCMessageManager 获取消息对象  UCCMessageManager messageManager = UCCMessageManager.getInstance(context);  UCCMessage message = messageManager.getUCCMessage(msgId);  // do something  }  // 客服正在输入  else if (UCCMessageManager.ACTION\_AGENT\_INPUTTING.equals(action)) {  // do something  }  // 客服接入  else if (UCCMessageManager.ACTION\_AGENT\_JOHN\_EVENT.equals(action)) {  // 获取接入后的客服  UCCAgent UCCAgent = messageManager.getCurrentAgent();  // do something  }  // 客服转接  else if (UCCMessageManager.ACTION\_AGENT\_CHANGE\_EVENT.equals(action)) {  // 获取转接后的客服  UCCAgent UCCAgent = messageManager.getCurrentAgent();  // do something  }  // 客服邀请  else if (UCCMessageManager.ACTION\_AGENT\_CHANGE\_EVENT.equals(action)) {  // 获取邀请的客服  UCCAgent UCCAgent = messageManager.getCurrentAgent();  // do something  }  // 客服邀请评价  else if (UCCMessageManager.ACTION\_AGENT\_ENQUIRY\_EVENT.equals(action)) {  // 获取满意度选项并展示  }  // 客服关闭  else if (UCCMessageManager.ACTION\_AGENT\_CLOSE\_EVENT.equals(action)) {  // 获取关闭后的客服  UCCAgent UCCAgent = messageManager.getCurrentAgent();  // do something  }  }  } |

### 获取服务器聊天数据

开发者可使用此接口来获取服务器的历史消息。服务端的历史消息是该顾客在**所有平台上**产生的消息，包括网页端、Android SDK、iOS SDK、微信，可在聊天界面的下拉刷新处调用

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| lastMessageCreateOn |  | 获取该日期之前的消息 |
| length |  | 获取的消息长度 |
| onGetMessageListCallback |  | 回调 |

样例：

|  |
| --- |
| getUCCMessageFromService(final long lastMessageCreateOn, final int length, final OnGetMessageListCallback onGetMessageListCallback) |

### 获取客户本地聊天数据

开发者也可使用此接口来获取本地的历史消息。由于没有同步服务端的消息，所以本地数据库的历史消息有可能少于服务端的消息。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| lastMessageCreateOn |  | 获取该日期之前的消息 |
| length |  | 获取的消息长度 |
| onGetMessageListCallback |  | 回调 |

样例：

|  |
| --- |
| getUCCMessageFromDatabase(final long lastMessageCreateOn, final int length, final OnGetMessageListCallback onGetMessageListCallback) |

### 获取未读消息数量

开发者可以在需要显示未读消息数是调用此接口，当用户进入聊天界面后，未读消息将会清零。

获取当前 Visitor 的未读消息数。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| onGetMessageListCallback |  | 回调 |

样例：

|  |
| --- |
| getUnreadMessagesCount(new OnGetMessageListCallback()) |

获取指定 VisitorID 客户的未读消息数。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| VisitorID |  | 客户ID：VisitorID |
| onGetMessageListCallback |  | 回调 |

样例：

|  |
| --- |
| getUnreadMessagesCount(String VisitorID, new OnGetMessageListCallback()) |

### 获取未读消息

开发者可以在需要显示未读消息时调用此接口，此接口会自动判断并合并本地和服务器上的未读消息，当用户进入聊天界面后，未读消息将会清空。

退出聊天界面后收到的消息，都将算作未读消息。

获取当前 Visitor 的未读消息。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| onGetMessageListCallback |  | 回调 |

样例：

|  |
| --- |
| getUnreadMessages(new OnGetMessageListCallback()) |

获取指定 VisitorID 客户的未读消息。

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| VisitorID |  | 客户ID：VisitorID |
| onGetMessageListCallback |  | 回调 |

样例：

|  |
| --- |
| getUnreadMessages(String VisitorID, new OnGetMessageListCallback()) |

### 删除客户聊天数据

SDK初始化成功，创建客户后，调用此接口可删除当前客户的聊天记录信息

样例：

|  |
| --- |
| deleteMsg() |

### 结束当前对话

结束当前对话

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| onEndConversationCallback |  | 回调 |

样例：

|  |
| --- |
| endCurrentConversation(OnEndConversationCallback onEndConversationCallback) |

### 获取当前正在接待的客服信息

获取当前正在接待的客服信息

参数样例

|  |  |  |
| --- | --- | --- |
| **返回参数名** | **参数值eg** | **备注** |
| Agent |  | 如果存在，返回当前客服信息；不存在，返回 null  客服ID、客服昵称 |

样例：

|  |
| --- |
| getCurrentAgent() |

### 更新阅读状态

更新阅读状态

参数样例

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| messageId |  | 消息id |
| isRead |  | 将替换的状态 1：已读 0：未读 |

样例：

|  |
| --- |
| updateMessage(long messageId, boolean isRead) |

### 获取SDK版本号

获取 SDK 版本号

参数样例

|  |  |  |
| --- | --- | --- |
| **返回参数名** | **参数值eg** | **备注** |
| SDK版本号 | V1.1.1 | SDK版本号 |

样例：

|  |
| --- |
| getUCCSDKVersion() |

### 设置日志模式

设置日志调试模式。设置为true后，将打印日志到控制台，发布时要关闭该选项。

样例：

|  |
| --- |
| setDebugMode（true/false） |

## 快速集成SDK

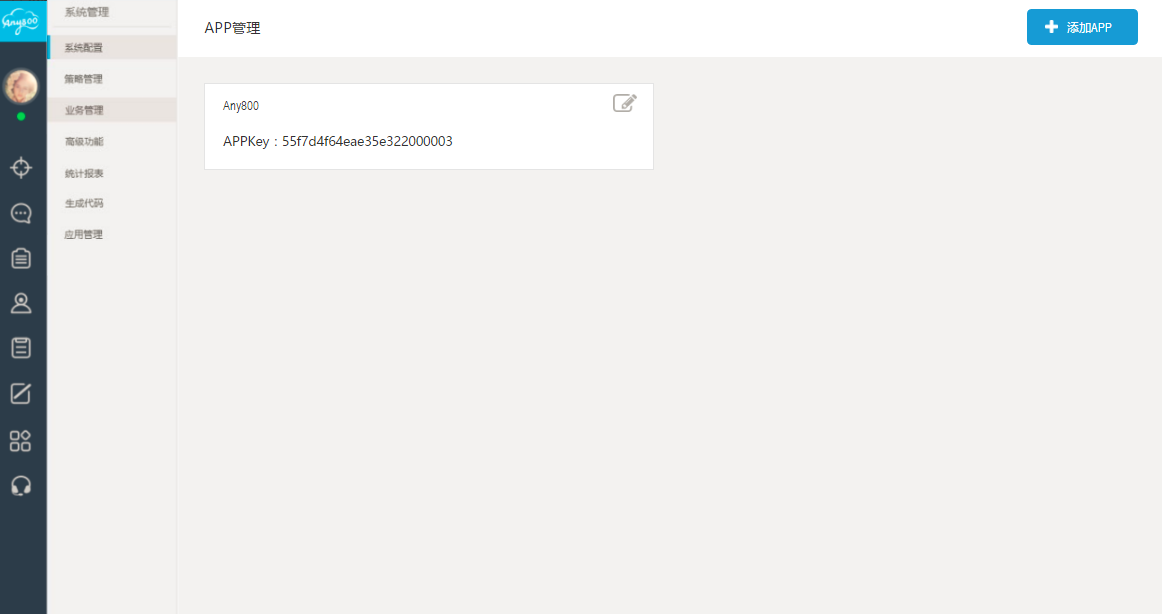
### 初始化SDK

第一次初始化SDK，SDK 会向UCC服务端发送一个初始化客户的请求，SDK 其他接口都必须是在初始化 SDK 成功后进行，所以 App 应尽早初始化 SDK。

样例

|  |
| --- |
| {  String appKey = "a71c257c80dfe883d92a64dca323ec20";  UCCConfig.init(this, appKey , new OnInitCallback() {  @Override  public void onSuccess(String VisitorId) {  Toast.makeText(MainActivity.this, "init success", Toast.LENGTH\_SHORT).show();  }  @Override  public void onFailure(int code, String message) {  Toast.makeText(MainActivity.this, "int failure", Toast.LENGTH\_SHORT).show();  }  });  } |

SDK的开发者需要在Any800管理后台的APP管理添加APP，主要包括APP名称、是否需要推送、开发者的APP服务器地址用于推送。对应的界面如下图所示。



点击添加APP对应界面为：



根据APP名称和公司ID为该APP生成对应的APPKey。

APP名称、APPKey及开发者的APP服务器地址用于消息推送。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 值范围 | 提示语 | 备注 |
| APP名称 | 文本框 | 非空唯一 |  | APP名称不能大于50个字符 |
| 推送方式 | 单选框 |  |  |  |
| 服务器地址 | 文本框 |  |  | 网址格式验证 |

根据APP名称和公司ID为该APP生成对应的APPKey。

APP名称、APPKey及开发者的APP服务器地址用于消息推送。

### [启动对话界面](http://meiqia.com/docs/meiqia-android-sdk/#2.启动对话界面)

初始化成功后，就可以直接启动对话界面

|  |
| --- |
| Intent intent = new UCCIntentBuilder(this).build();  startActivity(intent); |

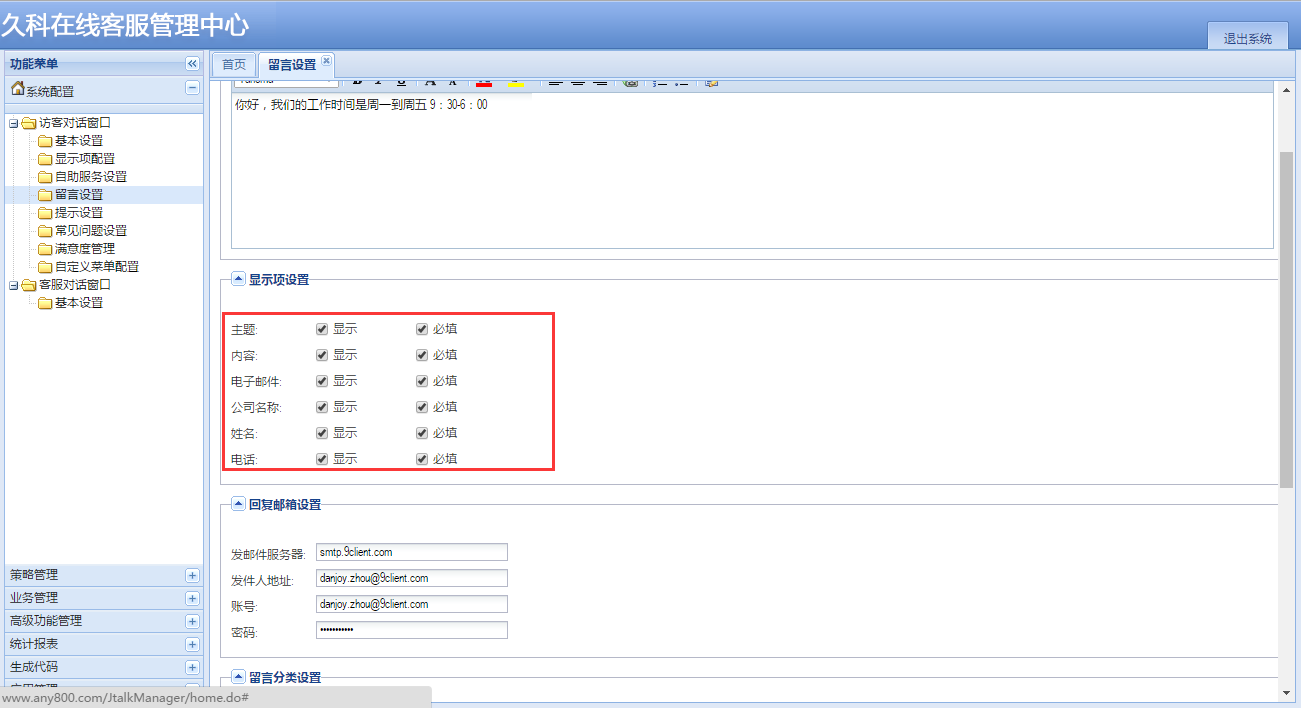
### 留言

进入人工客服对话界面。若无机器人，系统根据客服分配规则安排客服接待，如果客服繁忙或不在线，输入内容发送消息，则弹出用户留言提示；客服上线后能够直接回复，如果客服在线，则进入正常客服对话模式。若有机器人，直接聊天界面输入的话，由机器人回答，客户点击留言就会跳转到留言界面。

|  |
| --- |
| startActivity(new Intent(this, UCCMessageFormActivity.class)); |

留言界面的各字段是否显示、是否必填根据后台设置，进行对应的展示。

后台设置界面如下图所示。

对应的留言界面如下图所示。



|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 格式要求 | 备注 |
| 公司名称 | 文本框 |  | 提示：公司名称不能超过50字符！ |
| 姓名 | 文本框 | 1-20个字符 | 提示：姓名不能超过20字符！ |
| 电话 | 文本框 | 电话的格式验证；提示：请填写正确的电话格式！ | 11位的数字 |
| 电子邮件 | 文本框 | 电子邮件的格式；提示“请填写正确的邮箱！” |  |
| 主题 | 文本框 | 1-50字符 | 提示：主题不能大于50字符！ |
| 内容 | 文本框 | 1000字符 | 提示：内容不能大于1000字符！ |

对于后台设置非空的要提示“\*\*\*\*（字段名称）不能为空！”。

### 满意度评价

客服推送满意度给客户或客服关闭对话自动弹出满意度评价；SDK收到满意度调查消息（enquiry=true标志是满意度调查消息），获取满意度方案，推出满意度选项，客户选择后，提交满意度选项。

SDK收到满意度消息：消息内容为EnquiryOptions（满意度选项）。

SDK提交满意度选项：

|  |  |  |
| --- | --- | --- |
| **传入参数名** | **参数值eg** | **备注** |
| optionId |  | 选项ID |
| onMessageSendCallback |  | 消息状态回调 |

样例：

|  |
| --- |
| sendUCCEnquiryMessage(String optionId, final OnMessageSendCallback onMessageSendCallback) |

## 消息推送

UCC服务端发送消息至开发者的服务端，开发者再推送消息到 App。

设置服务器地址，在APP管理中添加APP时设置。对应的界面如下图所示。



### 关闭服务

关闭服务，UCC推送给开发者服务端的消息数据格式中，会有 deviceToken 的字段。

关闭服务后，将停止监听消息，此时UCC服务端将会推送消息给开发者提供的消息推送服务端，如下代码：

|  |
| --- |
| closeUCCService(); |

建议：在 App 后台以后，关闭服务。App 进入前台，如果需要监听客服消息，再开启服务。

### 开启服务

开启服务后，将重新监听消息，此时UCC服务端将不会推送消息给开发者提供的消息推送服务端，如下代码：

|  |
| --- |
| openUCCService(); |

建议：在 App 后台以后，关闭服务。App 进入前台，如果需要监听客服消息，再开启服务。

### [推送消息数据结构](http://meiqia.com/docs/meiqia-android-sdk/#推送消息数据结构)

当有消息需要推送时，UCC服务器会向开发者设置的服务器地址发送推送消息，方法类型为*POST*，数据格式为 *JSON* 。

发送的请求格式介绍：

request.header.authorization 为数据签名。

request.body 为消息数据，数据结构为：

|  |  |
| --- | --- |
| **Key** | **说明** |
| messageId | 消息id |
| content | 消息内容 |
| messageTime | 发送时间 |
| fromName | 发送人姓名 |
| deviceToken | 发送对象设备的 deviceToken，格式为字符串 |
| visitorID | 发送对象的顾客 id |
| customizedId | 开发者传的自定义 id |
| contentType | 消息类型 - text/photo/audio |
| deviceOS | 设备系统 |
| customizedData | 开发者上传的自定义的属性 |