

# AI 人工智能人机耦合对接说明 V1.0



系统名称	AI 人工智能人机耦合对接说明文档
项目负责人	廖思铎
文档提交日期	2019-06-04

深圳市一号互联科技有限公司

(版权所有, 翻版必究)

修改记录

No	修改后 版本号	修改内容简介	修改日期	修改人

# 1 背景

小 A 人工智能客服提供人机耦合的功能，开启全新的人机协作智能客服模式，赋能企业降本增效。该功能能够实现监听，无感介入及协同工作，保证客服中心服务质量的同时，有效辅助一线坐席缩短通话时长，提高坐席技能，减轻客服压力。此外，我们还提供第三方客户 crm 对接到我们的人机耦合，实现客户在自己的客户 crm 系统也能使用到小 A 的人机耦合功能。可以通过阅读本接口文档来帮助开发。



图 1 人机耦合的功能示意图

## 2 相关说明

人机耦合采用的是 WebRtc 跟底层进行对接，WebRTC 实现了基于网页的视频会议，标准是 WHATWG 协议，目的是通过浏览器提供简单的 javascript 就可以达到实时通讯 (Real-Time Communications (RTC)) 能力。WebRtc 目前只支持 https，所以第三方系统必须是 https 向目标服务器发起请求，请求协议是 wss (web\_socket)，浏览器需要支持 web\_socket 协议，建议使用 chrome (谷歌) 浏览器。

摘自网络：WebRTC 是一种开放式程序结构（框架），可在 Web 浏览器中提供实时通信（即实时通信），即 在点对点模式下，浏览器和其他应用程序之间的高质量音频/视频数据传输。包括使用 Javascript API 创建语音和视频通信的 Web 开发人员的组件。浏览器支持 Chrome, Opera, Firefox。WebRTC 项目提供完整的语音协议栈。目前支持的是 G.711, G.722, iLBC 和 iSAC 音频编解码器，以及 VP8 视频编解码器。有几个用 Javascript 编写的 SIP 客户端，例如 SipML5, sip-js, jsSIP, SIPjs

## 3 对接细则

### 3.1 获取子账号分机

目前人机耦合功能只针对子账号（一个子账号相当于一个坐席），对接前需要获取到当前子账号的分机信息，参数为三个公共参数即可。除此之外，子账号需要添加或者编辑的时候开通 web 分机，并且注册方式为“第三方注册”。保存好之后，就可以通过改接口返回一些关于对接人机耦合的注册信息。对接此接口可以参考《小 A 人工智能客服接口文档》

修改子账号

请输入以中文或英文开头的姓名

密码

确认密码

角色

A数

主叫号码

呼入话术

呼入号码

WEB分机

待分配分机数: 0

自动创建分机

注册方式

网页分机注册

第三方注册

保存

分机注册信息

分机号

分机注册密码

注册域名

端口号

{BaseUrl} /api/getExtToRegister 参数全部为 POST 方式

json 返回值示例:

```
{
  "linkext": "2",
  "user_ext": "1",
  "sipName": "xxxxxxx",
  "sipPwd": "xxxxxxx",
  "agentName": "xxxxxxx",
  "agentPwd": "$2y$10$Ge7j4*****qAqMNteMi",
  "serverIp": "test.com",
  "code": 0
}
```

返回参数说明:

参数名	类型	描述
linkext	string	分机注册方式：1:web 注册；2:第三方注册
user_ext	string	是否开通分机，1:开通
sipName	string	Sip 账号
sipPwd	string	Sip 密码
agentName	string	坐席账号
agentPwd	string	坐席密码
serverIp	String	目标服务器域名
code	Int	状态 为 0 表示请求成功 5000 表示异常

### 3.2 人机耦合对接

通过 3.1 获取的分机及坐席信息，就可以通过 web\_socket 连接到小 A 了，连接分俩步，1. 分机注册，2. webRTC 对接到 fs

1. 分机注册提供了一个 demo，在 sipip 文件夹里面里面，在前端页面引用 jssip-3.0.13.min.js，可以参考以下部分代码，请以 demo 为准

```

var ws_uri_ = 'wss://' + serverIp + ':' + 7443
var socket = new JsSIP.WebSocketInterface(ws_uri_);
var configuration = {
    sockets: [socket],
    outbound_proxy_set: ws_uri_,
    uri:'sip:' + sipName + '@' + serverIp,
    password: sipPwd,
    register: true,
    session_timers: false,
    audio: true
};
userAgent = new JsSIP.UA(configuration);
//注册成功
userAgent.on('registered', function (data) {
    console.info("registered: ", data.response.status_code, "", "",
data.response.reason_phrase);
});

```

其中，serverIp、sipName、sipPwd 取自 3.1 接口返回的 serverIp、sipName、sipPwd，new 成功之后就可以开启一些监听事件，按需开发，注册成功之后会回调对应的事件。

2.webRTC 对接到 fs 提供了一个 demo，在 fssocket 文件夹里面，需要引用 4 个 js，分别是 xml2json.min.js、ctimsg.min.js、ctilink.min.js、sip-0.12.0.min.js。可以参考以下部分代码，请以 demo 为准

```

var serverPort = 5008;
var serverAddr = 'wss://' + serverIp;
link = new CtiLink();
window.cti = link;

link.onReady = function() {
    console.log("onReady");
    link.setDebug(0);
    link.connect(serverAddr, serverPort);
}

link.onConnect = function(success, data) {
    console.log("onConnect");
    if(success) {
        link.login(agentName, agentPwd);
    } else {
        console.log('连接服务器失败, 请联系管理员. 原因: ' + data);
    }
}

```

其中，serverIp、agentName、agentPwd 取自 3.1 接口返回的 serverIp、agentName、agentPwd。new 成功之后就可以开启一些监听事件，按需开发，注册成功之后会回调对应的事件。

事件监听回调参数说明：

参数名	解释
callee	呼入号码
caller	呼出号码
callNo	会话 id
Confno	配置 id
icalltype	呼叫类型

msg	实时推送 json 数据	
	account_sn_callin	通话记录 sn
	uuid	会话 id
	user_name	用户名
	team_name	用户组
	connected_at	接通时间
	disconnected_at	挂断时间
	phone	电话
	project_caption	项目名称
	talktimes	通话时长
	endreason	挂断原因
	asks_count	询问次数
	name	姓名
	parent_sn	父级账号名称
	recordpath	分段录音路径
	talk_info	会话信息
	status	状态
	score	分数
	labelling	标签
	rec_sex	识别性别
	source	任务名称
	mark	备注
	sex	性别
	project_sn	项目 sn
	title	当前话术标题
	caption	当前话术描述
	skillgroup_sn	技能组 sn
	operator	操作人
	data1-data30	扩展字段
project_no	项目 sn	
task_no	任务 sn	
targetUser Name	需要修改坐席状态的用户名	
status	坐席状态	
statustime	坐席状态时间	