# 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 | 廖思锞 |
| 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 |
|  |  |  |  |  |

# 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只能客服模式，赋能企业降本增效。该功能能实现监听，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。此处之外，我们还提供第三方客户crm对接到我们的人机耦合，实现客户在有自己的客户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

# 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，所以第三方系统比较是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

# 对接细则

## 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》

{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 |

## 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 | |
| targetUserName | 需要修改坐席状态的用户名 | |
| status | 坐席状态 | |
| statustime | 坐席状态时间 | |