

# 呼叫中心接入文档

## 1.登录授权

### 说明

通过访问http接口授权接口获取token

### 请求方式

访问地址： /api/v1/base/auth

访问方法： POST application/json

### 参数说明

参数	类型	说明
username	string	用户名
password	string	密码(md5加密后提交)

### 返回值说明

参数		类型	说明
code		integer	0: 成功；大于0： 失败
data		object	
-	user	string	用户名
-	nickname	string	昵称
-	extNo	string	分机号
-	token	string	token
-	expiresAt	long	过期时间(过期时间为12小时)
msg		string	返回信息

### 请求案例

```
1 curl --request POST \  
2   --url http://127.0.0.1:9999/api/v1/base/auth \  
3   --header 'content-type: application/json' \  
4   --data '{  
5     "username": "admin",  
6     "password": "c33367701511b4f6020ec61ded352059"  
7   }'
```

## 成功返回值案例

```
1 {  
2   "code": 0,  
3   "data": {  
4     "user": "admin",  
5     "nickname": "admin",  
6     "extNo": "1001",  
7     "token":  
      "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJCi6MSwiVXNlcm5hbWUiOiJhZG1pb2IIsIkF1dGhvcml0eUlkIjozLCJleHAiOiJlE20DkzNjU2OTAsImZcyI6ImIldC1zZXJ2ZXIiLCJuYmYiOiJlE20DkzMjE0OTB9.ihAk9z1YXZWSxVvwyAjsxHseT7_o0vkYhNaRgJcalGSQ",  
8     "expiresAt": 1689365690  
9   },  
10  "msg": "登录成功"  
11 }
```

## 失败返回值案例

```
1 {  
2   "code": 7,  
3   "data": {},  
4   "msg": "用户名不存在或者密码错误"  
5 }
```

## 2.CTI电话条接入

### 说明

CTI电话条是以websocket长链接模式与后台交互，请保证电话条支持websocket能力，如使用webrtc模式尽可能使用chrome内核浏览器

# 引入JSSDK

```
1 <script src="dist/ctibar.js"></script>
```

## 初始化SDK

### 方法

```
var ctibar = new Ctibar(config, callback);
ctibar.initSDK();
```

### 说明

参数		类型	说明
config			
-	url	string	ws链接地址
-	token	string	登录授权获取的token
callback		function	callback为函数回调

### 参考

```
1 let ctibar = null;
2 let token = "";
3 let wsAddr = "ws://xxx.xxx.xxx.xxx/ws";
4
5 function sdkInit() {
6   //初始化config配置
7   let config = {
8     url: wsAddr,
9     token: token,
10    infoLog: (info, type) => {
11
12    },
13    errLog: (err) => {
14
15    },
16  };
17  let callback = {}; //callback为状态回调函数
18  ctibar = new Ctibar(config, callback);
19  ctibar.initSDK();
```

callback回调

说明

callback是jssdk内部推出电话条事件回调函数使用，参考一下代码实现。

回调	说明	事件参数	参数说明
onConnect	websocket成功链接		
onDisconnect	websocket断开链接		
onLogin	签入成功		
onLogout	签出成功		
onReady	示闲		
onBusy	示忙		
onRest	小休		
onRing	振铃事件		
--	振铃返回参数	callId	呼叫callId
--	振铃返回参数	popScreenNo	弹屏号码
--	振铃返回参数	direction	呼叫方向 (INBOUND/OUTBOUND/INTERNAL) 呼入/ 呼出/内部呼叫
--	振铃返回参数	enableAsr	是否启用实时ASR
onAnswer	坐席应答		
onCalleeAnswer	被叫接通（外呼使用判断 被叫是否接听）		
onHangup	通话结束		

参考

```
1  let callback = {
2    onConnect: () => {
3      document.getElementById("state").innerHTML = "已连接";
4    },
5
6    onDisconnect: () => {
7      document.getElementById("state").innerHTML = "断开连接";
8    },
9
10   onLogin: (data) => {
11     document.getElementById("state").innerHTML = "已登录";
12     document.getElementById("extNo").innerHTML = data.extNo;
13   },
14
15   onLogout: () => {
16     document.getElementById("state").innerHTML = "登出";
17     document.getElementById("extNo").innerHTML = "";
18   },
19
20   onReady: () => {
21     document.getElementById("state").innerHTML = "空闲";
22   },
23
24   onBusy: () => {
25     document.getElementById("state").innerHTML = "忙碌";
26   },
27
28   onRest: () => {
29     document.getElementById("state").innerHTML = "小休";
30   },
31
32   onRing: (data) => {
33     document.getElementById("state").innerHTML = "振铃中";
34     document.getElementById("info").innerHTML = data.popScreenNo;
35   },
36
37   onAnswer: (data) => {
38     if (data.direction === "OUTBOUND") {
39       document.getElementById("state").innerHTML = "外呼中";
40     } else {
41       document.getElementById("state").innerHTML = "通话中";
42     }
43   },
44
45   onCalleeAnswer: () => {
46     document.getElementById("state").innerHTML = "通话中";
47   },
```

```
48
49     onHangup: () => {
50         document.getElementById("state").innerHTML = "挂机";
51         document.getElementById("info").innerHTML = "";
52     },
53 };
54 }
```

## 签入

### 方法

```
ctibar.login()
```

### 说明

### 参考

## 签出

### 方法

```
ctibar.logout()
```

### 说明

### 参考

## 示闲

### 方法

```
ctibar.ready()
```

### 说明

### 参考

## 示忙

### 方法

```
ctibar.busy()
```

说明

参考

### 小休

方法

```
ctibar.rest()
```

说明

参考

### 外呼

方法

```
ctibar.makecall(to)
```

说明

参数	类型	说明
to	string	被叫号码，可以是内部分机号码，也可以是外线号码

注：通常坐席状态需变更到示忙状态下外呼，以防止与呼入电话冲突。

参考

### 摘机

方法

```
ctibar.offhook()
```

说明

注：当使用webrtc电话时直接通过内置webrtc模块应答电话，若使用sip话机则通过想服务器发送指令命令话机接听，非所有话机试用。

参考

挂断

方法

```
ctibar.hangup()
```

说明

参考

保持

方法

```
ctibar.hold()
```

说明

参考

取消保持

方法

```
ctibar.unhold()
```

说明

参考

转接

方法

```
ctibar.transfer(to, type)
```

说明

参数	类型	说明
type	string	转接类型。 to_queue：转技能组; to_phone：转分机
to	string	被转接者。当为转技能组时 to为技能组ID, 转分机时直接填写分机号即可



参考

咨询

方法

```
ctibar.consult(to)
```

说明

参数	类型	说明
to	string	被叫分机

参考

咨询转接

方法

```
ctibar.consultComplete()
```

说明

参考

咨询取回

方法

```
ctibar.consultCancel()
```

说明

参考

### 3.话单回调

说明

话单回调由客户(CRM或其他第三方系统提供接口), 话务系统在生成话单保存至数据库时同时发起http请求(POST方式), 将话单推送至第三方平台管理, 第三方返回无需返回任务结果, 只需返回http code 200成功状态即可。

请求参数

参数	类型	说明
callId	string	通话唯一ID
callerUuid	string	主叫uuid
calleeUuid	string	被叫uuid
caller	string	主叫
callee	string	被叫
pstn	string	呼入呼出使用的线路号码
direction	string	呼叫方向, (INBOUND/OUTBOUND/INTERNAL/UNKNOWN)
startTime	string	呼叫开始时间,格式:2023-07-17T14:04:00+08:00
ringTime	string	呼叫振铃时间,格式同上
answerTime	string	呼叫接通时间,格式同上
endTime	string	呼叫挂机时间,格式同上
ivrTime	string	进入IVR时间,格式同上
acdTime	string	进入队列时间,格式同上
billSec	integer	通话时长(秒)
billMin	integer	通话时长(分钟),不足1分钟按1分钟计算
ringSec	integer	振铃时长
duration	integer	通话持续总时长
answered	bool	是否接通, true/false
userId	integer	关联用户ID
nickname	string	用户昵称
acdId	integer	技能组ID

acdName	string	技能组名称
stereoRecordFile	string	录音文件名,(拼接前缀 <a href="http://xx.xx.xx.xx/">http://xx.xx.xx.xx/</a> 即可播放的网络地址)
stereoFullPath	string	所在服务器完整路径

参考数据

```
1 {
2     "callId": "f684f271-6d6a-4821-9245-fff6cc5c5ea3",
3     "callerUuid": "4ef5778f-ad1a-49eb-a580-d89ca2a819b9",
4     "calleeUuid": "3b4c75e0-80b7-43ef-bc89-c61e81617b23",
5     "caller": "1001",
6     "callee": "13606254580",
7     "pstn": "051287660011",
8     "direction": "OUTBOUND",
9     "startTime": "2023-07-17T14:03:59+08:00",
10    "ringTime": "2023-07-17T14:04:00+08:00",
11    "answerTime": "2023-07-17T14:04:01+08:00",
12    "endTime": "2023-07-17T14:04:04+08:00",
13    "ivrTime": "2023-07-17T14:03:59+08:00",
14    "acdTime": null,
15    "billSec": 3,
16    "billMin": 1,
17    "ringSec": 1,
18    "duration": 5,
19    "answered": true,
20    "userId": 1,
21    "nickname": "admin",
22    "acdId": "",
23    "acdName": "",
24    "stereoRecordFile": "/recordings/20230717/20230717140400_f684f271-6d6a-4821-9245-fff6cc5c5ea3_1001_13606254580.mp3",
25    "stereoFullPath": "/opt/recordings/20230717/20230717140400_f684f271-6d6a-4821-9245-fff6cc5c5ea3_1001_13606254580.mp3"
26 }
```

4.http接口接入

# 外呼

## 说明

通过http接口方式（POST）发起呼叫，接口授权参考【登录授权】。

注：每次外呼的主叫分机是由授权token解析而出，与账号绑定的分机号关联

## 参数

参数	类型	说明
callee	string	被叫号码

## 请求案例

```
1 curl --location --request POST
  'http://127.0.0.1:9999/api/v1/thirdparty/makecall' \
2 --header 'x-token:
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJJRCI6MSwiVXNlcm5hbWUiOiJhZG1pbiIsIkF1dG
  hvcm10eUlkIjoxLCJleHAiOjE2ODk2MjgyOTQsImZcyI6ImIwdC1zZXJ2ZXIiLCJuYmYiOjE2ODk1O
  DQwOTR9.xZ5UKirUx3rQK00gCuaE55pLfk3DegshVVfJ0C45s7Q' \
3 --header 'Content-Type: application/json' \
4 --data-raw '{
5   "callee": "13606254580"
6 }'
```

注：将获取的token以header的方式添加，key值为『x-token』

## 返回值

参数	子参数	类型	说明
code		integer	0:成功；其余失败
data		object	
-	callId	string	通话唯一callId
msg		string	成功或失败原因

```
1 // 成功返回值
2 {
```

```

3     "code": 0,
4     "data": {
5         "callId": "c128c823-7e1b-4663-864a-b8822b8919bb"
6     },
7     "msg": "操作成功"
8 }
9
10 // 失败返回值
11 {
12     "code": 7,
13     "data": {
14         "reload": true
15     },
16     "msg": "未登录或非法访问"
17 }

```

## 附录代码

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>电话条测试工具</title>
6   </head>
7   <body>
8     <script src="dist/ctibar.js"></script>
9     <script src="./lib/axios.min.js"></script>
10    <script src="./lib/dayjs.min.js"></script>
11    <script>
12      //获取授权token保存
13      let token = "";
14
15      let ctibar = null;
16
17      //websocket连接地址
18      let wsAddr = "ws://127.0.0.1:9999/ws";
19
20      //登录授权方法
21      function auth() {
22        let username = document.getElementById("username").value;
23        let password = document.getElementById("password").value;
24        axios({
25          method: "post",

```

```
26     url: "http://127.0.0.1:9999/api/v1/base/auth",
27     //url: "https://demo.cdxxzz.com/api/v1/base/auth",
28     data: {
29         username: username,
30         password: password,
31     },
32 }).then(function (res) {
33     console.info(res);
34     if (res.data.code === 0) {
35         token = res.data.data.token;
36         document.getElementById("token").value = token;
37     } else {
38         document.getElementById("token").value = "";
39         alert("获取token失败!!!");
40     }
41 });
42 }
43
44 function sdkInit() {
45     //初始化config配置
46     let config = {
47         url: wsAddr,
48         token: token,
49         errLog: (err) => {
50             console.log(err);
51         },
52     };
53     let callback = {
54         onConnect: () => {
55             document.getElementById("state").innerHTML = "已连接";
56         },
57
58         onDisconnect: () => {
59             document.getElementById("state").innerHTML = "断开连接";
60         },
61
62         onLogin: (data) => {
63             document.getElementById("state").innerHTML = "已登录";
64             document.getElementById("extNo").innerHTML = data.extNo;
65         },
66
67         onLogout: () => {
68             document.getElementById("state").innerHTML = "登出";
69             document.getElementById("extNo").innerHTML = "";
70         },
71
72         onReady: () => {
```

```
73         document.getElementById("state").innerHTML = "空闲";
74     },
75
76     onBusy: () => {
77         document.getElementById("state").innerHTML = "忙碌";
78     },
79
80     onRest: () => {
81         document.getElementById("state").innerHTML = "小休";
82     },
83
84     onRing: (data) => {
85         document.getElementById("state").innerHTML = "振铃中";
86         document.getElementById("info").innerHTML = data.popScreenNo;
87     },
88
89     onAnswer: (data) => {
90         if (data.direction === "OUTBOUND") {
91             document.getElementById("state").innerHTML = "外呼中";
92         } else {
93             document.getElementById("state").innerHTML = "通话中";
94         }
95     },
96
97     onCalleeAnswer: () => {
98         document.getElementById("state").innerHTML = "通话中";
99     },
100
101     onHangup: () => {
102         document.getElementById("state").innerHTML = "挂机";
103         document.getElementById("info").innerHTML = "";
104     },
105
106     onAsrResult: (data) => {
107         console.log(data);
108     },
109 };
110 ctibar = new Ctibar(config, callback);
111 ctibar.initSDK();
112 }
113
114 //拨打电话
115 function call() {
116     let called = document.getElementById("makecall").value;
117     ctibar.makecall(called);
118 }
119 </script>
```

```
120 <div>
121   <h3>1. 获取Auth授权</h3>
122   <div>
123     <label>用户名</label>
124     <input id="username" type="text" value="admin" />
125   </div>
126   <div>
127     <label>密码</label>
128     <input
129       id="password"
130       type="password"
131       value="c33367701511b4f6020ec61ded352059"
132     />
133   </div>
134   <button onclick="auth()">获取授权</button>
135   <input id="token" style="width: 500px" type="text" value="" />
136 </div>
137
138 <div>
139   <h3>2. 加载电话条</h3>
140   <div>
141     <p><label>分机号: </label><label id="extNo"></label></p>
142     <p><label>当前状态: </label><label id="state">未连接</label></p>
143     <p>
144       <label>持续时长:</label>
145     </p>
146     <p><label>呼叫信息:</label><label id="info"></label></p>
147     <p>
148       <button onclick="sdkInit()">初始化</button>
149     </p>
150     <p>
151       <button onclick="ctibar.login()">登录</button>
152       <button onclick="ctibar.logout()">登出</button>
153       <button onclick="ctibar.ready()">空闲</button>
154       <button onclick="ctibar.busy()">置忙</button>
155       <button onclick="ctibar.rest()">小休</button>
156     </p>
157     <p>
158       <input id="makecall" value="" />
159       <button onclick="call()">拨打</button>
160       <button onclick="ctibar.offhook()">摘机</button>
161       <button onclick="ctibar.hangup()">挂机</button>
162       <button onclick="ctibar.hold()">保持</button>
163       <button onclick="ctibar.unhold()">取消保持</button>
164     </p>
165     <p>
166       <input id="transferNo" value="" />
```



```
167         <button>转接</button>
168
169         <input id="transferQueue" value="" />
170         <button>转接技能组</button>
171     </p>
172     <p>
173         <input id="consult" value="" />
174         <button>咨询</button>
175         <button>咨询转</button>
176         <button>咨询取回</button>
177     </p>
178 </div>
179 </div>
180 <div
181     id="log_panel"
182     style="
183         width: 1100px;
184         height: 600px;
185         background: black;
186         position: fixed;
187         top: 30px;
188         right: 150px;
189         font-family: auto;
190         color: chartreuse;
191         padding-left: 10px;
192         overflow-y: scroll;
193         word-break: break-all;
194     "
195 ></div>
196 </body>
197 </html>
```