

功能描述:

- 1. 对话平台语音机器人接口文档

API 接口

api名称	请求方式	url	comments
query	websoket		语音机器人对话

Query 请求

请求格式

JSON

请求参数

参数	类型	是否必选	描述
bot_id	uint64	是	机器人id
sess_id	string	是	session id
time_zone	string	否	如果存在，表示当前query的具体时区 取值范围： 'Etc/GMT+12','Etc/GMT+11','Etc/GMT+10','Etc/GMT+9', 'Etc/GMT+8','Etc/GMT+7','Etc/GMT+6','Etc/GMT+5','Etc/GMT+4', 'Etc/GMT+3','Etc/GMT+2','Etc/GMT+1','Etc/GMT-1','Etc/GMT-2','Etc/GMT-3','Etc/GMT-4', 'Etc/GMT-5','Etc/GMT-6','Etc/GMT-7','Etc/GMT-8','Etc/GMT-9','Etc/GMT-10', 'Etc/GMT-11','Etc/GMT-12','Etc/GMT-13','Etc/GMT-14'
country_code	string	是	
need_vad	bool	是	机制待确定
debug	string	否	是否返回调试信息，"on"：返回，"off"：不返回。不填时等同于"off"
status	string	是	StartTranscription 表示准备传送语音流 StopTranscription 表示语音流传送完成

返回参数

参数	是否必选	类型	描述
request_id	是	string	表示该请求的唯一标识符,与对应请求时一致
country_code	是	string	
bot_id	是	uint64	机器人id,与对应请求时一致
query	是	string	asr后的结果
sess_id	是	string	session id,与对应请求时一致
bot_type	是	string	bot类型: "FAQ_BOT", "EAOG_BOT", "KG_BOT", "CHAT_BOT"
voice	否	string	返回base64编码的音频, (err_code为非零时, 没有此字段)
sentence	否	string	语音识别的结果,也是本次对话的query
result	否	list[json]	返回文本结果(err_code为非零时, 没有此字段)
+question	否	string	当bot_type=FAQ_BOT时, 并且最大置信度在0.2和0.8之间, 那么FAQ会列出相似问题,用于推荐给用户。 当bot_type=EAOG_BOT时,没有该字段。 当bot_type=CHAT_BOT时, 没有该字段。
+confidence	否	float	当bot_type=FAQ_BOT时, 表示FAQ的置信度 当bot_type=AOG_BOT时, 没有该字段。 当bot_type=CHAT_BOT时, 没有该字段。
+answer	是	string	问题答案 当bot_type=EAOG_BOT时, 表示当前EAOG_BOT回复指令的输出
+instruction	否	dict	当bot_type=EAOG_BOT时, 表示当前EAOG_BOT动作指令的输出 其他bot_type没有该字段
faq_type	否	string	当bot_type=FAQ_BOT时, type=keyword时表示是核心词匹配, type=synonym时表示是核心词同义词匹配, type=keyword_rule时表示, 核心词规则泛化, type=no_keyword时表示没有用到核心词相关的。 当bot_type=EAOG_BOT时, 没有该字段。 当bot_type=KG_BOT时, 后续再定义。 当bot_type=CHAT_BOT时, 没有该字段。
intent	否	string	调试信息：debug模式时返回该字段内容 当bot_type=EAOG_BOT时, 表示当前用户query的意图 其他bot_type没有该字段
params	否	list[dict]	调试信息：debug模式时返回该字段内容 当bot_type=EAOG_BOT时, 表示当前用户query的意图对应的槽位 其他bot_type没有该字段
+param_name	否	string	槽名
+origin	否	string	槽的原始值
+value	否	string	槽的标准值
score	否	float	调试信息：debug模式时返回该字段内容 当bot_type=EAOG_BOT时, 表示当前用户query的意图置信度 当bot_type=FAQ_BOT时, 表示当前用户query的Top-1匹配度 其他bot_type没有该字段
dialogue_status	否	string	调试信息：debug模式时返回该字段内容 当bot_type=EAOG_BOT时, 表示当前对话流的状态, 有以下三种取值： "dialogue_flow_start": 处于当前对话流第一轮的回答, 如果对话流只有一轮, 应该标识为end。 "dialogue_flow_end": 处于当前对话流最后一轮的回答 "dialogue_flow_in": 处于当前对话流中间的回答, 如果对话流只有一轮, 应该标识为end。 其他bot_type没有该字段
err_code	是	int	非零时表示有错误, 并返回错误消息
err_msg	否	string	提示性错误消息

语音流接口

协议

使用socket.io协议进行通讯, 文档可以参考 [Socket.io-Java](#), 有如下事件

msg 事件: 传送二进制语音流, 是二进制

identify 事件: 发送语音开始和结束的标志, 是json格式

流程

#1 open channel

当客户端打开通道的时候, 服务端会回复下面内容, 表示通道已经打开了.

"state": "ChannelOpened" 表示服务端和客户端channel已经打开

```
{ "state": "ChannelOpened", "request_id": "a0f49ac1-6533-499c-ad5a-026ca57b8787" }
```

#2 发送Payload并且开启语音识别

socket.io中的identify事件

客户端发送如下Json给服务端, 在第一次语音识别之前需要发送下面字段, 然后服务端会保存当前channel中bot\_id/sess\_id/need\_vad/time\_zone/country\_code/debug. 这些参数在当前的channel中不可改变. 后面再次发送语音前, 只用传送 ("status": "StartTranscription")

```
{  "status": "StartTranscription",  "bot_id": 74,  "sess_id": "777",  "need_vad": false,  "time_zone": 'Etc/GMT+12',  "country_code" : "123456",  "debug": "on"}
```

服务端收到开启语音流识别的接口会返回

```
{ "action": "TranscriptionStarted", "err_code": 0, "request_id": "a0f49ac1-6533-499c-ad5a-026ca57b8787" }
```

"action": "TranscriptionStarted" 表示客户端可以开始发送语音流

#3 发送二进制制语音流

socket.io中的msg事件

要求的语音流格式为 "Format"; "audio/L16;rate=16000"

客户端发送的语音块大小可以是任意的, 服务端会自动调整识别时候语音块的大小.

服务端语音流识别过程会回复如下, message 是语音识别中间结果

```
{ "message": "Let me show you our new school welcome let me show you our new ", "state": "TranscriptionResultChanged", "request_id": "a0f49ac1-6533-499c-ad5a-026ca57b8787" }
```

"state": "TranscriptionResultChanged" 表示语音流中间识别结果变化

当服务端完整识别一句话结束会返回下面内容, message是语音识别的结果

```
{ "message": "Let me show you our new school welcome let me show you our new school. Thank you. ", "state": "SentenceEnd", "request_id": "a0f49ac1-6533-499c-ad5a-026ca57b8787" }
```

"state": "SentenceEnd" 表示的是一句话识别的结果

#4 发送语音结束符

socket.io中的identify事件

```
{ "status": "StopTranscription" }
```

对话流识别的结果如下, 还有其他返回参数请参考上面「返回参数」内容, 语音机器人比文本机器人多了 voice/sentence/state 这三个字段

```
{ "bot_id": 74, "bot_type": "FAQ_BOT",  "err_code": 0, "request_id": "633a50df215442789ea384c48aa1fb46",  "result": [ { "answer": "...", "confidence": 0.8500000238418579, "qa_id": 295,    "question": "\u4e0d\u60f3\u7ed3\u5a5a\u4e6e\u5ff6\u5f69\u793c\u4e00\u5b9a\u8981\u9000\u56de\u53bb\u5417\u5ff1" } ],  "sess_id": "777",  "type": "hit",  "user_id": -1,  ....  "voice": "VwtsR1Jp....",  "sentence": "Let me show you our new school welcome...",  "state": "TranscriptionCompleted" }
```

"state": "TranscriptionCompleted" 表示本次语音输入的对话流返回结果

#5 开启新的一段语音识别

socket.io中的identify事件

只需要发送以下开启字段

```
{ "status": "StartTranscription" }
```