**扣子websocketapi-上行**

扣子提供流式语音对话 WebSocket OpenAPI，向指定的智能体发起语音对话。

双向流式语音对话场景下的各类事件详细信息可参考[双向流式对话事件](https://www.coze.cn/open/docs/developer_guides/streaming_chat_event)。

**接口信息**

|  |  |
| --- | --- |
| **URL** | wss://ws.coze.cn/v1/chat |
| **Headers** | Authorization Bearer *$Access\_Token*  用于验证客户端身份的访问令牌。你可以在扣子平台中生成访问令牌，详细信息请参考[准备工作](https://www.coze.cn/docs/developer_guides/preparation)。 |
| **权限** | chat |
| **接口说明** | 向指定的智能体发起语音对话。 |

**Query**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| bot\_id | String | 必选 | 需要关联的智能体 ID。  进入智能体的开发页面，开发页面 URL 中 bot 参数后的数字就是智能体 ID。例如 https://www.coze.com/space/341\*\*\*\*/bot/73428668\*\*\*\*\*，Bot ID 为 73428668\*\*\*\*\*。 |
| workflow\_id | String | 可选 | 待执行的对话流 ID，此对话流应已发布。  进入对话流编排页面，在页面 URL 中，workflow 参数后的数字就是对话流 ID。例如 https://www.coze.com/work\_flow?space\_id=42463\*\*\*&workflow\_id=73505836754923\*\*\*，对话流 ID 为 73505836754923\*\*\*。 |

**建连示例代码**

|  |
| --- |
| javascript import WebSocket from 'ws';    const url = `wss://ws.coze.cn/v1/chat?bot\_id=${BOT\_ID}&authorization=Bearer ${ACCESS\_TOKEN}`;  const ws = new WebSocket(url);    ws.on('open', function open() {   console.log('Connected to server.');  });    ws.on('message', function incoming(*message*) {   console.log(JSON.parse(message.toString()));  }); |

**上行事件**

**更新对话配置**

* **事件类型**：chat.update
* **事件说明**：此事件可以更新当前对话连接的配置项，若更新成功，会收到 chat.updated 的下行事件，否则，会收到 error 下行事件。
* **事件结构**：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| id | String | 必选 | 客户端自行生成的事件 ID，方便定位问题。 |
| event\_type | String | 必选 | 固定为 chat.update。 |
| data | Object | 可选 | 事件数据，包含对话配置的详细信息。 |
| data.chat\_config | Object | 可选 | 对话配置。 |
| data.chat\_config.meta\_data | Map<String, String> | 可选 | 附加信息，通常用于封装一些业务相关的字段。查看对话消息详情时，系统会透传此附加信息。自定义键值对，应指定为 Map 对象格式。长度为 16 对键值对，其中键（key）的长度范围为 1～64 个字符，值（value）的长度范围为 1～512 个字符。 |
| data.chat\_config.custom\_variables | Map<String, String> | 可选 | 智能体中定义的变量。在智能体 prompt 中设置变量 {{key}} 后，可以通过该参数传入变量值，同时支持 Jinja2 语法。详细说明可参考变量示例。变量名只支持英文字母和下划线。 |
| data.chat\_config.extra\_params | Map<String, String> | 可选 | 附加参数，通常用于特殊场景下指定一些必要参数供模型判断，例如指定经纬度，并询问智能体此位置的天气。自定义键值对格式，其中键（key）仅支持设置为：   * latitude（纬度，此时值（Value）为纬度值，例如 39.9800718）。 * longitude（经度，此时值（Value）为经度值，例如 116.309314）。 |
| data.chat\_config.user\_id | String | 可选 | 标识当前与智能体的用户，由使用方自行定义、生成与维护。user\_id 用于标识对话中的不同用户，不同的 user\_id，其对话的上下文消息、数据库等对话记忆数据互相隔离。如果不需要用户数据隔离，可将此参数固定为一个任意字符串，例如 123，abc 等。 |
| data.chat\_config.conversation\_id | String | 可选 | 标识对话发生在哪一次会话中。会话是智能体和用户之间的一段问答交互。一个会话包含一条或多条消息。对话是会话中对智能体的一次调用，智能体会将对话中产生的消息添加到会话中。可以使用已创建的会话，会话中已存在的消息将作为上下文传递给模型。创建会话的方式可参考创建会话。对于一问一答等不需要区分 conversation 的场合可不传该参数，系统会自动生成一个会话。不传的话会默认创建一个新的 conversation。 |
| data.chat\_config.auto\_save\_history | Boolean | 可选 | 是否保存本次对话记录。   * true：（默认）会话中保存本次对话记录，包括本次对话的模型回复结果、模型执行中间结果。 * false：会话中不保存本次对话记录，后续也无法通过任何方式查看本次对话信息、消息详情。在同一个会话中再次发起对话时，本次会话也不会作为上下文传递给模型。 |
| data.chat\_config.parameters | Map<String, any> | 可选 | 设置对话流的输入参数。   * 对话流的输入参数 USER\_INPUT 应在 additional\_messages 中传入，在 parameters 中的 USER\_INPUT 不生效。 * 如果 parameters 中未指定 CONVERSATION\_NAME 或其他输入参数，则使用参数默认值运行对话流；如果指定了这些参数，则使用指定值。 |
| data.input\_audio | Object | 可选 | 输入音频格式。 |
| data.input\_audio.format | String | 可选 | 输入音频的格式，支持 pcm、wav、ogg。默认为 wav。 |
| data.input\_audio.codec | String | 可选 | 输入音频的编码，支持 pcm、opus、g711a、g711u。默认为 pcm。  如果音频编码格式为 g711a 或 g711u，format 请设置为 pcm。 |
| data.input\_audio.sample\_rate | Integer | 可选 | 输入音频的采样率，默认是 24000。支持 8000、16000、22050、24000、32000、44100、48000。  如果音频编码格式 codec 为 g711a 或 g711u，音频采样率需设置为 8000。 |
| data.input\_audio.channel | Integer | 可选 | 输入音频的声道数，支持 1（单声道）、2（双声道）。默认是 1（单声道）。 |
| data.input\_audio.bit\_depth | Integer | 可选 | 输入音频的位深，默认是 16，支持8、16和24。 |
| data.output\_audio | Object | 可选 | 输出音频格式。 |
| data.output\_audio.codec | String | 可选 | 输出音频编码，支持 pcm、opus。默认是 pcm。 |
| data.output\_audio.pcm\_config | Object | 可选 | 当 codec 设置为 opus 时，不需要设置此字段。   |  | | --- | | 当 codec 设置为 pcm 时，返回的 PCM 数据将固定为单声道，采样深度为 16 位。 | |
| data.output\_audio.pcm\_config.sample\_rate | Integer | 可选 | 输出 pcm 音频的采样率，默认是 24000。支持 8000、16000、22050、24000、32000、44100、48000。 |
| data.output\_audio.pcm\_config.frame\_size\_ms | Float | 可选 | 输出每个 pcm 包的时长，单位 ms，默认不限制。 |
| data.output\_audio.pcm\_config. limit\_config | Object | 可选 | 输出音频限流配置，默认不限制。 |
| data.output\_audio.pcm\_config. limit\_config.period | Integer | 可选 | 周期的时长，单位为秒。例如设置为 10 秒，则以 10 秒作为一个周期。 |
| data.output\_audio.pcm\_config. limit\_config.max\_frame\_num | Integer | 可选 | 周期内，最大返回 pcm 包数量。 |
| data.output\_audio.opus\_config | Object | 可选 | 当 codec 设置为 pcm 时，不需要设置此字段。 |
| data.output\_audio.opus\_config.bitrate | Integer | 可选 | 输出 opus 的码率，默认 48000。 |
| data.output\_audio.opus\_config.use\_cbr | Boolean | 可选 | 输出 opus 是否使用 CBR 编码，默认为 false。 |
| data.output\_audio.opus\_config.frame\_size\_ms | Float | 可选 | 输出 opus 的帧长，默认是 10。可选值：  2.5、5、10、20、40、60 |
| data.output\_audio.opus\_config.limit\_config | Object | 可选 | 输出音频限流配置，默认不限速。 |
| data.output\_audio.opus\_config.limit\_config.period | Integer | 可选 | 周期的时长，单位为秒。例如设置为 10 秒，则以 10 秒作为一个周期。 |
| data.output\_audio.opus\_config.limit\_config.max\_frame\_num | Integer | 可选 | 周期内最大返回的 Opus 帧数量。 |
| data.output\_audio.speech\_rate | Integer | 可选 | 输出音频的语速，取值范围 [-50, 100]，默认为 0。-50 表示 0.5 倍速，100 表示 2 倍速。 |
| data.output\_audio.voice\_id | String | 可选 | 输出音频的音色 ID，默认是柔美女友音色。你可以调用[查看音色列表](https://www.coze.cn/open/docs/developer_guides/list_voices) API 查看当前可用的所有音色 ID。 |
| data.event\_subscriptions | Array<String> | 可选 | 需要订阅下行事件的事件类型列表。不设置或者设置为空为订阅所有下行事件。 |
| data.need\_play\_prologue | Boolean | 可选 | 是否需要播放开场白，默认为 false。 |
| data.prologue\_content | String | 可选 | 自定义开场白，need\_play\_prologue 设置为 true 时生效。如果不设定自定义开场白则使用智能体上设置的开场白。 |
| data.turn\_detection | Object | 可选 | 转检测配置。 |
| data.turn\_detection.type | String | 可选 | 用户演讲检测模式，包括：   * **server\_vad** ：语音数据会传输到服务器端进行实时分析，服务器端的语音活动检测算法会判断用户是否在说话。 * **client\_interrupt**：（默认）客户端实时分析语音数据，并检测用户是否已停止说话。 |
| data.turn\_detection.prefix\_padding\_ms | Integer | 可选 | server\_vad 模式下，VAD 检测到语音之前要包含的音频量，单位为 ms。默认为 600ms。 |
| data.turn\_detection.silence\_duration\_ms | Integer | 可选 | server\_vad 模式下，检测语音停止的静音持续时间，单位为 ms。默认为 500ms。 |
| data.turn\_detection.interrupt\_config | Object | 可选 | server\_vad 模式下打断策略配置 |
| data.turn\_detection.interrupt\_config.mode | String | 可选 | 打断模式, 包括:   * keyword\_contains模式下，说话内容**包含**关键词才会打断模型回复。例如关键词"扣子"，用户正在说“你好呀扣子......” / “扣子你好呀”，模型回复都会被打断。 * keyword\_prefix模式下，说话内容**前缀匹配**关键词才会打断模型回复。例如关键词"扣子"，用户正在说“扣子你好呀......”，模型回复就会被打断，而用户说“你好呀扣子......”，模型回复不会被打断。 |
| data.turn\_detection.interrupt\_config.keywords | Array<String> | 可选 | 打断的关键词配置，最多同时限制 5 个关键词，每个关键词限定长度在6-24个字节以内(2-8个汉字以内), 不能有标点符号。 |
| data.asr\_config | Object | 可选 | 语音识别配置，包括热词和上下文信息，以便优化语音识别的准确性和相关性。 |
| data.asr\_config.hot\_words | Array<String> | 可选 | 请输入热词列表，以便提升这些词汇的识别准确率。  所有热词加起来最多100个 Tokens，超出部分将自动截断。 |
| data.asr\_config.context | String | 可选 | 请输入上下文信息。  最多输入 800 个 Tokens，超出部分将自动截断。 |
| data.asr\_config.user\_language | String | 可选 | 用户说话的语种，默认为 common。选项包括：   * common：大模型语音识别，可自动识别中英粤。 * zh：小模型语音识别，中文。 * cant：小模型语音识别，粤语。 * sc：小模型语音识别，川渝。 * en：小模型语音识别，英语。 * ja：小模型语音识别，日语。 * ko：小模型语音识别，韩语。 * fr：小模型语音识别，法语。 * id：小模型语音识别，印尼语。 * es：小模型语音识别，西班牙语。 * pt：小模型语音识别，葡萄牙语。 * ms：小模型语音识别，马来语。 * ru：小模型语音识别，俄语。 |
| data.asr\_config.enable\_ddc | Boolean | 可选 | 将语音转为文本时，是否启用语义顺滑。默认为 true。   * true：系统在进行语音处理时，会去掉识别结果中诸如 “啊”“嗯” 等语气词，使得输出的文本语义更加流畅自然，符合正常的语言表达习惯，尤其适用于对文本质量要求较高的场景，如正式的会议记录、新闻稿件生成等。 * false：系统不会对识别结果中的语气词进行处理，识别结果会保留原始的语气词。 |
| data.asr\_config.enable\_itn | Boolean | 可选 | 将语音转为文本时，是否进行文本纠错，默认为 true。 |
| data.asr\_config.enable\_punc | Boolean | 可选 | 将语音转为文本时，是否给文本加上标点符号。默认为 true。 |

* **事件示例**：

|  |
| --- |
| json {   "id": "event\_id",   "event\_type": "chat.update",   "data": {   "chat\_config": {   "auto\_save\_history": true, // 保存历史记录。默认 true   "conversation\_id": "xxxx", // conversation\_id   "user\_id": "xxx", // 标识当前与智能体的用户，由使用方自行定义、生成与维护。user\_id 用于标识对话中的不同用户，不同的 user\_id，其对话的上下文消息、数据库等对话记忆数据互相隔离。如果不需要用户数据隔离，可将此参数固定为一个任意字符串   "meta\_data": {}, // 附加信息,通常用于封装一些业务相关的字段。查看对话消息详情时,系统会透传此附加信息。   "custom\_variables": {}, // 智能体中定义的变量。在智能体prompt中设置变量{{key}}后,可以通过该参数传入变量值,同时支持Jinja2语法。详细说明可参考变量示例。变量名只支持英文字母和下划线。   "extra\_params": {}, // 附加参数,通常用于特殊场景下指定一些必要参数供模型判断,例如指定经纬度,并询问智能体此位置的天气。自定义键值对格式,其中键(key)仅支持设置为:latitude:纬度,此时值(Value)为纬度值,例如39.9800718。longitude:经度,此时值(Value)为经度值,例如116.309314。   "parameters": {"custom\_var\_1": "测试"}   },   "input\_audio": { // 输入音频格式   "format": "pcm", // 输入音频格式，支持 pcm/wav/ogg。默认 wav   "codec": "pcm", // 输入音频编码。 pcm/opus。默认 pcm   "sample\_rate": 24000, // 采样率   "channel": 1, // 通道数   "bit\_depth": 16 // 位深   },   "output\_audio": { // 输出音频格式   "codec": "pcm",   "pcm\_config": {   "sample\_rate": 16000, // 默认 24000   "frame\_size\_ms": 50,   "limit\_config": {   "period": 1,   "max\_frame\_num": 22   }   },   "speech\_rate": 0, // 回复的语速，取值范围 [-50, 100]，默认为 0，-50 表示 0.5 倍速，100 表示 2倍速   "voice\_id": "7426720361733046281"   }   }  } |

**流式上传音频片段**

* **事件类型**：input\_audio\_buffer.append
* **事件说明**：流式向服务端提交音频的片段。
* **事件结构**：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| id | String | 必选 | 客户端自行生成的事件 ID，方便定位问题。 |
| event\_type | String | 必选 | 固定为 input\_audio\_buffer.append。 |
| data | Object | 必选 | 事件数据，包含音频片段信息。 |
| data.delta | String | 必选 | base64 编码后的音频片段。 |

|  |
| --- |
| json {   "id": "event\_id",   "event\_type": "input\_audio\_buffer.append",   "data": {   "delta": "base64EncodedAudioDelta"   }  } |

**提交音频**

* **事件类型**：input\_audio\_buffer.complete
* **事件说明**：客户端发送 input\_audio\_buffer.complete 事件来告诉服务端提交音频缓冲区的数据。服务端提交成功后会返回 input\_audio\_buffer.completed 事件。在 server\_vad 模式下，提交此事件无效。
* **事件结构**：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| id | String | 必选 | 客户端自行生成的事件 ID，方便定位问题。 |
| event\_type | String | 必选 | 固定为 input\_audio\_buffer.complete。 |

|  |
| --- |
| Markdown {   "id": "event\_id",   "event\_type": "input\_audio\_buffer.complete"  } |

**清除缓冲区音频**

* **事件类型**：input\_audio\_buffer.clear
* **事件说明**：客户端发送 input\_audio\_buffer.clear 事件来告诉服务端清除缓冲区的音频数据。服务端清除完后将返回 input\_audio\_buffer.cleared 事件。在 server\_vad 模式下，提交此事件无效。
* **事件结构**：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| id | String | 必选 | 客户端自行生成的事件 ID，方便定位问题。 |
| event\_type | String | 必选 | 固定为 input\_audio\_buffer.clear。 |

|  |
| --- |
| Markdown {   "id": "event\_1",   "event\_type": "input\_audio\_buffer.clear"  } |

**清除上下文**

* **事件类型**：conversation.clear
* **事件说明**：清除上下文，会在当前 conversation 下新增一个 section，服务端处理完后会返回 conversation.cleared 事件。
* **事件结构**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| id | String | 必选 | 客户端自行生成的事件 ID，方便定位问题。 |
| event\_type | String | 必选 | 固定为 conversation.clear。 |

|  |
| --- |
| Markdown {   "id": "event\_1",   "event\_type": "conversation.clear"  } |

**打断智能体输出**

* **事件类型**：conversation.chat.cancel
* **事件说明**：发送此事件可取消正在进行的对话，中断后，服务端将会返回 conversation.chat.canceled 事件。
* **事件结构**：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **是否必选** | **说明** |
| id | String | 必选 | 客户端自行生成的事件 ID，方便定位问题。 |
| event\_type | String | 必选 | 必填 conversation.chat.cancel。 |

|  |
| --- |
| Markdown {   "id": "7446668538246561827",   "event\_type": "conversation.chat.cancel"  } |