腾讯云智能体开发平台

应用接口文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

应用接口文档
- 对话接口总体概述
- 对话端接口文档（WebSocket）
- 对话端接口文档（HTTP SSE）
- 图片对话或文件对话（实时文档解析+对话）
- 离线文档上传
- 腾讯云智能体开发平台操作 cos 指南

版权所有：腾讯云计算（北京）有限责任公司

应用接口文档
对话接口总体概述

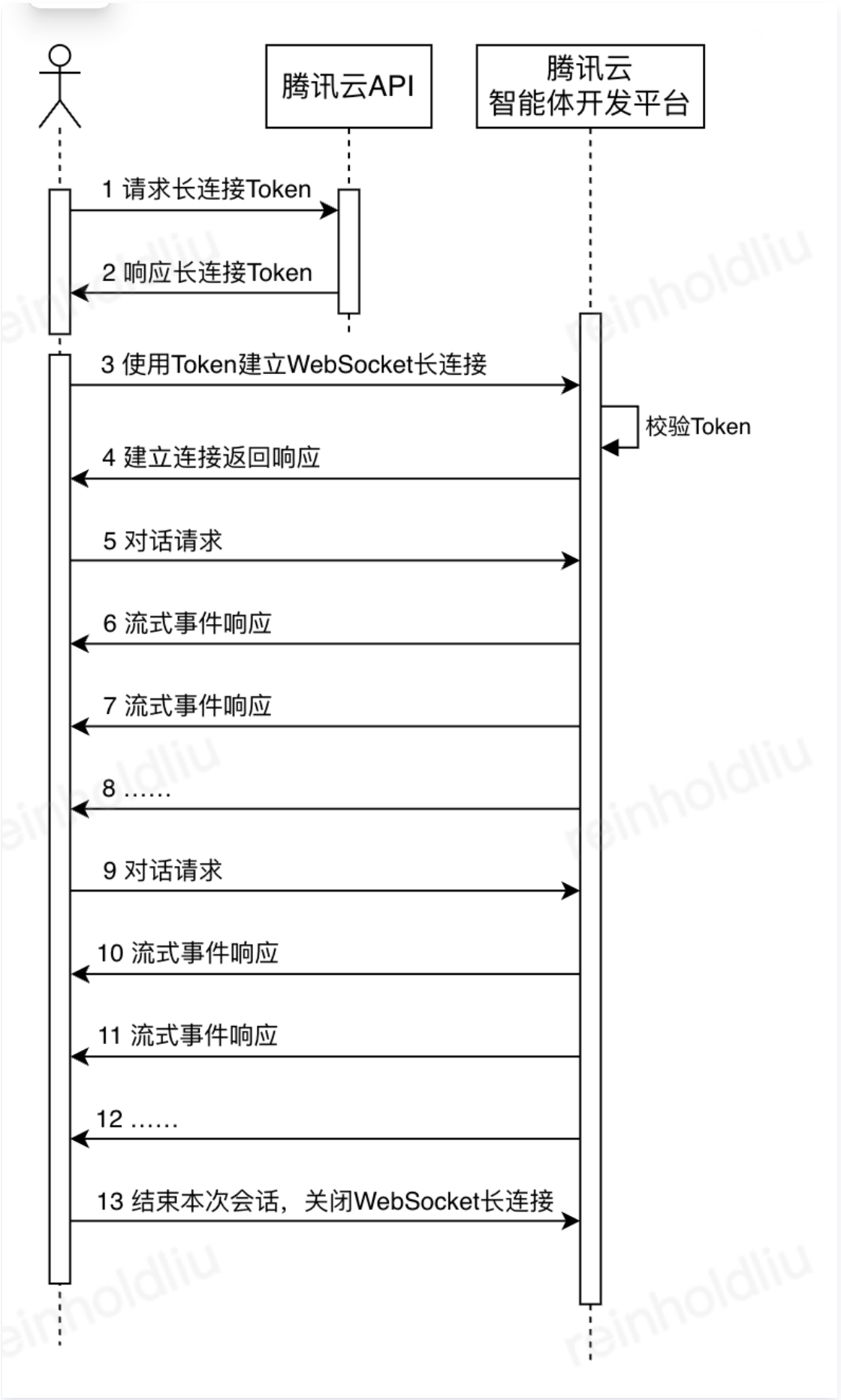
最近更新时间：2025-05-20 00:00:52

概述

在智能体开发平台创建应用，完成应用配置、对话测试和发布并获取 AppKey 以后，就可以使用对话端接口与智能体开发平台进行交互。
平台提供了两种常用的接入方式：WebSocket 与 HTTP SSE 的方式。

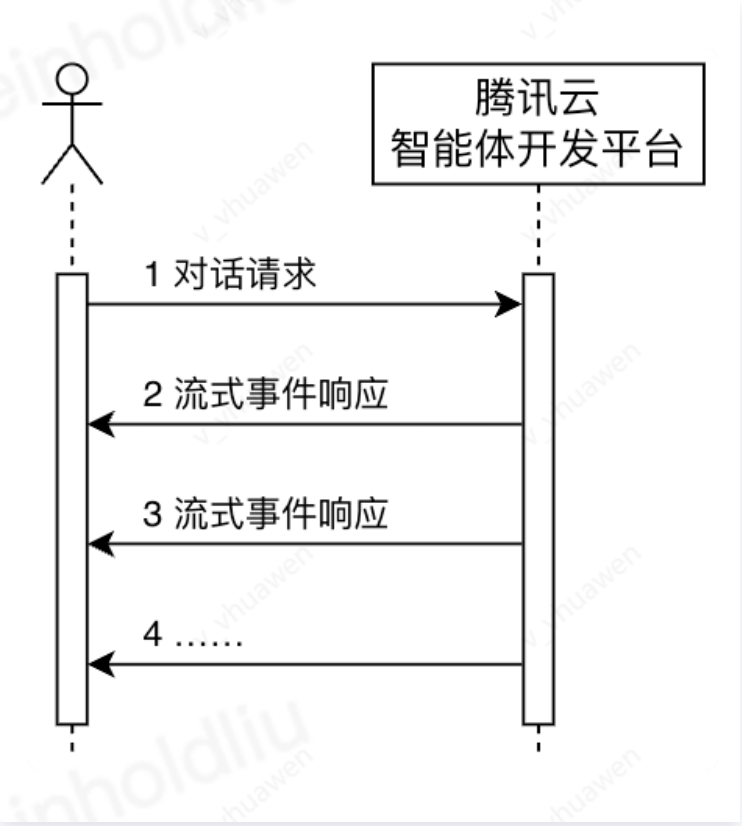
1. WebSocket 接入方式简介

WebSocket 是面向链接的，全双工通道。建立链接前需要先从服务端获取 Token。通过这个 token 才能与服务端创建 WebSocket 链接。Token 仅建立链接时有效，链接建立成功后废弃。
接入流程图：


腾讯云API
腾讯云
智能体开发平台
1 请求长连接Token
2 响应长连接Token
3 使用Token建立WebSocket长连接
校验Token
4 建立连接返回响应
5 对话请求
6 流式事件响应
7 流式事件响应
8 ……
9 对话请求
10 流式事件响应
11 流式事件响应
12 ……
13 结束本次会话，关闭WebSocket长连接

2. HTTP SSE 接入方式简介

HTTP SSE 是单向通道，客户端发起 HTTP 请求之后，服务端持续推送流式数据到客户端，此时不支持双向交互。



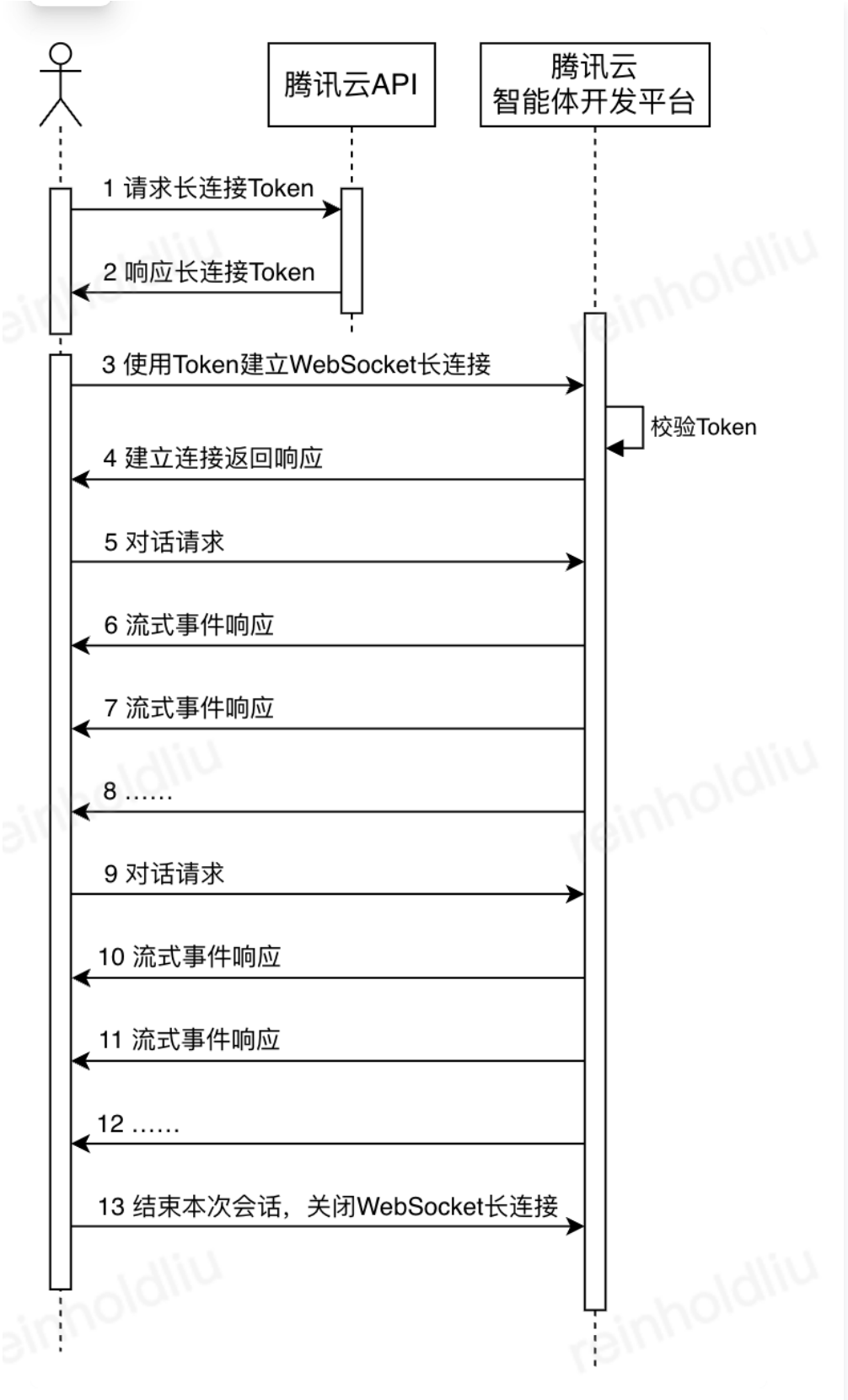
注意：

如果想了解配置端相关接口，请访问：API 概览 。

对话端接口文档（WebSocket）

最近更新时间：2025-06-24 10:14:52

WebSocket 是面向链接的，全双工通道。建立链接前需要先从服务端获取 Token。通过这个 token 才能与服务端创建 WebSocket 链接。Token 仅建立链接时有效，链接建立成功后废弃。
接入流程图：



1. 获取建立 Websocket 连接的 Token

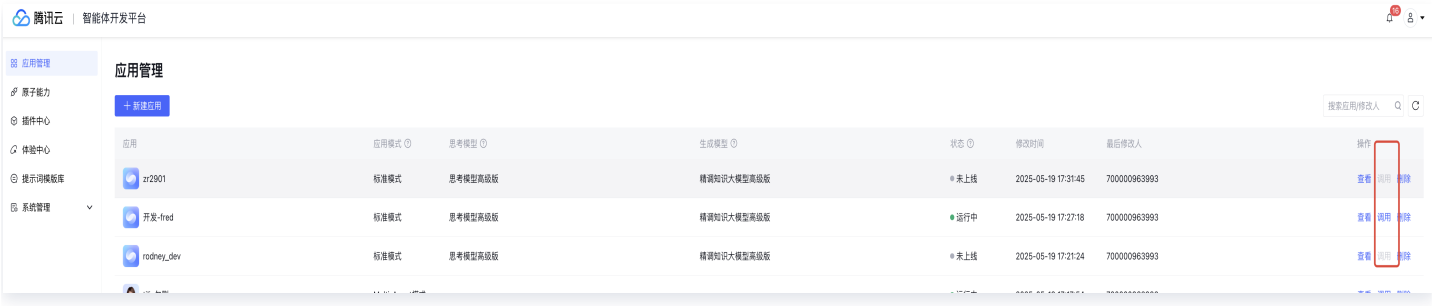
1.1 调用方式

通过腾讯云 SDK 调用 GetWsToken 接口获取 Token，可参考 GetWsToken 文档。

> **说明：**
>
> - 获取的 Token **仅供一次会话连接使用**，**并且会过期**，请在获取到 Token 后及时建立长连接，如需建立其他连接，需要重新获取 Token。
> - **标签相关的值不再使用该接口传递，请参考下面** 数据结构 **中的**custom_variables **传入知识库检索范围相关参数 。**

1.2 如何获取 AppKey

在**应用管理**界面，找到您处于运行中的应用（需要先发布），单击**调用**，会弹出“调用信息”窗口。



在**调用信息**窗口可以看到 AppKey，单击**复制**即可。

2. 使用 Token 建立 Websocket 连接

请求地址：

`wss://wss.lke.cloud.tencent.com/v1/qbot/chat/conn/?EIO=4&transport=websocket`

请求协议：Socket.IO v4（ 参考文档 ）

> **说明：**
> 可参考本页面下方 对话端 Demo 代码 。

2.1 建立连接

Websocket 连接建立成功后，服务器回包如下所示：

```
0{"sid":"xxx","upgrades":[],"pingInterval":25000,"pingTimeout":5000}
```

2.2 传递 token 鉴权

通过 Socket.IO 的 auth 消息传递。
在建立连接并收到服务器回包后，发送 Token 鉴权。发送 Token 格式如下所示：

```
40{"token":"xx-xx-xx-xx-xx"}
```

2.3 处理心跳包

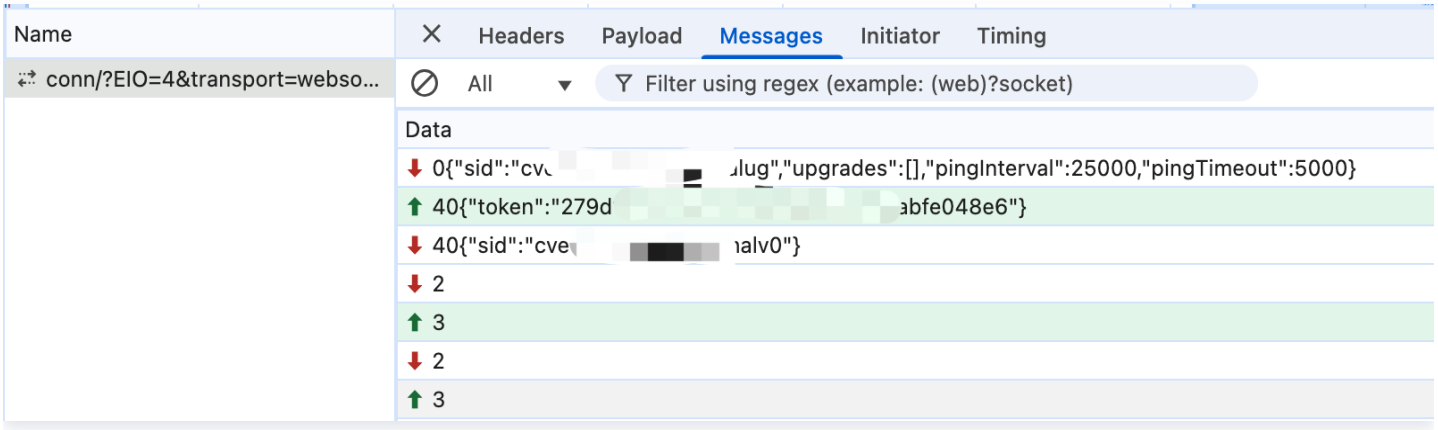
服务器发送的心跳包如下（"2"是心跳包的内容）：

```
2
```

此时客户端需要响应（"3"是需要响应的内容）：

```
3
```

> **注意：**
> - Socket io V4 有心跳包， 必须处理，否则会被服务器断开连接。
> - 如果自己实现 Client，需要注意处理心跳包。本文档提供的 Demo 已经自动处理了心跳包。



3. WebSocket 支持的事件

Socket.IO 事件格式如下，实现时需要注意其结构。建议尽可能使用 Socket.IO 提供的 标准 Client，或参考本文档下方提供的 前后端 Demo 。

```
42["类型",{"payload":{事件体}}]

例如：
42["send",{"payload":{请参考下面各个不同事件的数据结构}}]
```

3.1 发送事件

事件名：send
事件方向：**前端 > 后端**

> ⚠ **注意：**
> - 发送 send 事件前，需要先发布应用。
> - 用户发出一条消息（发送事件）时，服务器会将该消息原样返回（回复事件，其中 is_from_self = true），以便确认消息被服务端收到并对应更新消息 ID、时间戳。
> - 如果需要携带知识标签信息，请在获取 token 时携带。

数据结构：

名称	类型	是否必填	说明
request_id	string(255)	否	请求 ID，用于标识一个请求（作消息串联，建议每个请求使用不同的 request_id）。 为了便于问题排查，**建议必填。**
session_id	string(64)	是	会话 ID，用于标识一个会话（外部系统提供，建议不同的用户端会话传入**不同的**session_id，否则同一个应用下的不同用户的消息记录会串掉） **参数长度：2-64个字符** **校验规则： ^[a-zA-Z0-9_-]{2,64}$ ，一般可以用 uuid 来生成该值** **uuid 示例：1b9c0b03-dc83-47ac-8394-b366e3ea67ef**
content	string(6000)	是	消息内容，如果发送图片，在此传递 markdown 格式的图片链接，例如，其中图片链接需要可公有读。
file_infos	Object 数组	否	文件信息，如果填写该字段，content 字段可以为空。数据结构请参考 file_infos ，依赖 实时文档解析 。 > ⚠ **注意：** > 没有文件信息时，该字段可不填或指定为空数组。
custom_variables	map[string]string	否	**工作流场景：** 自定义参数可用于传递参数给工作流开始节点的API参数字段，通过该API参数字段可以被工作流识别并使用，具体参考 开始节点 使用文档。

			自定义API参数的值。可以配置多个 key: value 对，key 为自定义参数的参数名称，value 为对应的自定义参数的运行时的值。 **注意：** map的key和value类型都是string，如 {"UserID":"10220022"}， 不合法的参数如：{"UserID": 10220022}； 其中value也可以是json字符串，如 {"Data": "{\"UserID\":\"10220022\",\"Score\":{\"Chinese\":89,\"Math\":98}}"} **检索知识库范围设置场景：** 自定义参数可用于设置知识库检索范围，如需传入多个参数值，使用英文竖线分隔(|)，例如： "user1|user2"。 **注意：** 如果需要在问题中指定知识库进行检索，可以通过该字段传入相关的标签值， 具体请参考 知识库检索范围设置 。
system_role	string(4000)	否	角色指令（提示词），为空时使用应用配置默认设定，填写时取当前值。
incremental	bool	否	控制**回复事件**和**思考事件**中的 content 是否是增量输出的内容，默认 false 。 **注意：** 该字段设置为 true , 只对回复事件, 思考事件的生效。 回复事件效果： 思考事件效果：

search_network	string	否	是否开启联网搜索。取值范围： • 空字符串：跟随应用配置 • enable：打开联网搜索 • disable：关闭联网搜索 **注意：** 应用配置在未开启联网搜索时，如上取值字段无意义。
model_name	string	否	支持在对话的过程传入指定模型， 同一个 session_id 下传入不同的模型名称可以保持上下文之间的关联。 当前支持下面几个自定义模型： • hunyuan：混元大模型高级版 • hunyuan-13B：混元大模型标准版 • hunyuan-turbo：混元大模型Turbo版 • hunyuan-standard-256K：混元大模型长文本版 • hunyuan-role：混元大模型角色扮演版 • lke-deepseek-r1：DeepSeek-R1 • lke-deepseek-v3：DeepSeek-V3 • lke-deepseek-r1-0528: DeepSeek-R1-0528 • lke-deepseek-v3：DeepSeek-V3-0324 **注意：** 请确保选择的模型有足够的 token 或者开启了后付费模式，避免在使用的过程中报错。
stream	string	否	是否开启流式传输。取值范围： • 空字符串：跟随应用配置 • enable：流式传输 • disable：非流式传输
workflow_status	string	否	是否开启工作流。取值范围： • 空字符串：跟随应用配置（页面无配置工作流开关的入口，默认是开启） • enable：开启工作流

			• disable：关闭工作流

file_infos 文件信息的数据结构：

名称	类型	是否必填	说明
file_name	string	是	文件名称。
file_size	string	是	实时文档解析接口返回的文件大小。
file_url	string	是	实时文档解析接口返回的文件 URL。
file_type	string	是	文件类型。
doc_id	string	是	实时文档解析接口返回的 doc_id。

3.2 回复事件

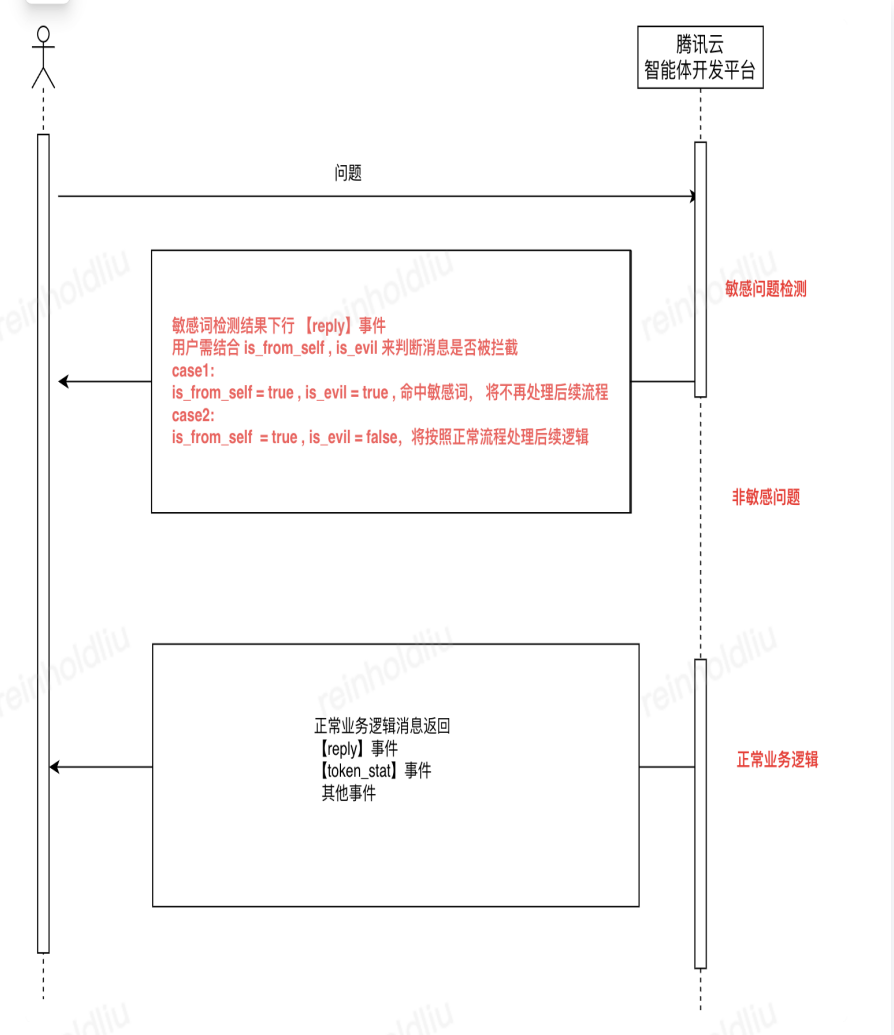
事件名：reply
事件方向：后端 > 前端

> ⚠ **注意：**
> - 如果收到的消息中 is_evil == true 表示该消息命中敏感内容，发送失败。
> - 因并发超限导致排队超时，会下发 "超出并发数限制" 错误。

数据结构：

名称	类型	说明
request_id	string(255)	请求 ID，用于标识一个请求（作消息串联，建议每个请求使用不同的 request_id）
content	string	回复消息内容
file_infos	Object 数组	文件信息
record_id	string(64)	消息唯一 ID
related_record_id	string(64)	关联的消息唯一 ID

session_id	string(64)	会话 ID，用于标识一个会话（外部系统提供，建议不同的用户端会话传入**不同的** session_id，否则同一个应用下的不同用户的消息记录会串掉）
is_from_self	bool	消息是否由客户端发出
can_rating	bool	该消息记录是否能评价
timestamp	int64	消息时间戳（秒级）
is_final	bool	reply 事件消息是否已输出完成 • 流式模式下，消息会多次返回，每次返回均覆盖之前的答案 • 当 is_final == true 时，停止生成按钮隐藏，并且显示点赞点踩按钮 • reference 事件和 reply 事件无固定顺序关系，不建议以该字段作为消息完整结束的标识。
is_evil	bool	是否命中敏感内容 **注意：消息上行后会先进行敏感内容检测，并返回一条【reply】事件，告知敏感内容检测结果， 敏感问题检测通过后才会进入正常业务逻辑处理**

		
is_llm_generated	bool	是否为模型生成内容
reply_method	uint8	回复方式： 1: 大模型回复 2: 未知问题回复 3: 拒答问题回复 4: 敏感回复 5: 已采纳问答对优先回复 6: 欢迎语回复 7: 并发数超限回复 8: 全局干预知识 9: 任务流回复 10: 任务流答案 11: 搜索引擎回复 12: 知识润色后回复 13: 图片理解回复 14: 实时文档回复 15: 澄清确认回复

		16: 工作流回复 17: 工作流运行结束 18: 智能体回复 19: 多意图回复
knowledge	Object 数组	命中的知识
option_cards	string 数组	选项卡，任务流程专有 说明：该字段可能为空，为空时该字段不返回
custom_params	string 数组	用户自定义业务参数，用于透传问答中业务参数 说明：该字段可能为空，为空时该字段不返回
task_flow	Object	任务流程调试信息 说明：该字段可能为空，为空时该字段不返回
work_flow	Object	工作流调试信息 说明：该字段可能为空，为空时该字段不返回
quote_infos	Object 数组	引用信息

knowledge 命中的知识的数据结构：

名称	类型	说明
id	string	命中的知识 ID
type	uint32	命中的知识类型： 1: 问答 2: 文档片段
seg_id	string	片段 ID，可通过 DescribeSegments 接口查看片段详情

work_flow 工作流调试信息的数据结构：

名称	类型	说明
workflow_name	string	工作流名称
workflow_id	string	工作流ID
qworkflow_run_id	string	工作流运行ID

option_cards	string 数组	选项卡
current_node	Object	当前节点信息
outputs	string	输出结果

current_node 当前工作流节点信息：

名称	类型	说明
NodeID	string	节点ID
NodeType	int32	节点类型： 0：未指定 1： 开始节点 2：参数提取节点 3：大模型节点 4：知识问答节点 5：知识检索节点 6：标签提取节点 7：代码执行节点 8：工具节点 9：逻辑判断节点 10：回复节点 → 消息节点 11：选项卡节点 12：循环节点 13：意图识别节点 14：工作流节点 15：插件节点 16：结束节点
NodeName	string	节点名称
Status	int32	状态： 0：初始状态 1： 运行中 2：运行成功 3：运行失败
Input	string	节点的输入。json字符串（含普通字符串）
Output	string	节点的最终输出。json字符串（含普通字符串）
TaskOutput	string	任务的输出。（原始输出）

FailMessage	string	异常信息
CostMilliSeconds	string	节点的总耗时。如果节点有多次调用，耗时为多次调用的总和。
Reply	string	当前节点的回复内容。当前参数提取、消息节点、结束回复节点可能非空
BelongNodeID	string	节点所属工作流被引用时的引用节点的ID，所属的工作流是被引用的时候非空。
StatisticInfo	Object	LLM 统计信息

StatisticInfo LLM 统计信息：

名称	类型	说明
ModelName	string	模型名称
FirstTokenCost	uint32	首token耗时
TotalCost	uint32	推理总耗时
InputTokens	uint32	输入token数量
OutputTokens	uint32	输出token数量
TotalTokens	uint32	输入+输出总token

quote_infos 引用信息数据结构：

名称	类型	说明
index	int	引用的角标索引， 角标 icon 请接入方自行设计实现
position	int	在回复内容 content 中的位置(中文字符，英文字符，其他符号都算一个字符)

效果：

quote_infos 示例 (此处只展示，省略了部分结构)：

```
[
    "reply",
    {
        "type": "reply",
        "payload": {
            "can_feedback": true,
            "can_rating": false,
            "content": "2025年2月13日腾讯股价情况如下：\n\n### 腾讯控股（00700.HK）港股市场表现\n\n- **股价波动**：腾讯控股股价在2025年2月13日经历了显著的波动，盘中振幅达到4.35%。这一波动既反映了市场情绪与资金博弈的短期影响，也体现了行业环境与公司基本面的长期逻辑。\n- **股价范围**：腾讯股价在当天最高达到464.6港币，最低跌至437港币。\n\n### 腾讯控股（TCTZF.us）美股市场表现\n\n- **股价上涨**：在美股市场，腾讯控股（TCTZF.us）股价在2025年2月13日盘中一度上涨5.05%，最新报58.10美元/股。\n- **市场反应**：这一异动引发了市场的广泛关注，投资者纷纷开始挖掘其背后的原因及影响。\n\n请注意，以上信息仅供参考，投资有风险，入市请谨慎。",

            "quote_infos": [
                {
                    "index": 1,
                    "position": 323
```

```
                }
            ],

        },
        "message_id": "xxxxxxxxxxxxxxxxxxxxx"
    }
]
```

3.3 token 统计事件

事件名：token_stat
事件方向：后端 > 前端
数据结构：

名称	类型	说明
session_id	string(64)	会话 id
request_id	string(255)	对应发送事件对应的请求 id
record_id	string(64)	对应发送事件对应的消息记录 id
status_summary	string	本轮对话状态, 处理中: processing, 成功: success, 失败: failed
status_summary_title	string	本轮对话状态描述
elapsed	int	本轮调用耗时, 单位 ms
token_count	int	本轮请求消耗 token 数(当包含多个过程时, 计算将汇总)
procedures	Object 数组	调用过程列表

procedures 调用过程列表的数据结构：

名称	类型	说明
name	string	英文名, 与下面的 title 字段一一对应. knowledge, task_flow, search_engine, image, large_language_model, pot_math, file
title	string	调用过程描述, 对应 name 字段, 各中文含义如下:

		调用知识库, 调用任务流程, 调用搜索引擎, 调用图片理解, 大模型回复, 调用计算器, 阅读文件
status	string	调用过程状态, 处理中: processing, 成功: success, 失败: failed
input_count	int	当次过程输入消耗 token 数
output_count	int	当次过程输出消耗 token 数
count	int	当次过程消耗总 token 数：输入 + 输出

示例:


```
[
    "token_stat",
    {
        "type": "token_stat",
        "payload": {
            "elapsed": 1616,
            "order_count": 50000000,
            "procedures": [
                {
                    "count": 323,
                    "input_count": 308,
                    "name": "knowledge",
                    "output_count": 15,
                    "status": "success",
                    "title": "调用知识库"
                }
            ],
            "record_id": "Hpe_20240625_185659_215_EsH2uf8L",
            "request_id": "8PUcDU6xyQ-301747294000",
            "session_id": "2d071ef7-ef76-44df-84a4-9210672ed700c8",
            "status_summary": "success",
            "status_summary_title": "调用知识库",
            "token_count": 323,
            "used_count": 553
        },
        "message_id": "89d91395-06bc-4f2e-b240-06f7b4498b0c6e"
    }
]
```

3.4 评价事件

事件名：rating
事件方向：**双向**

> ⚠ **注意：**
> 客户端发出评价事件时候，客户端也会收到这个事件，以便客户端确认消息发出成功。

数据结构：

名称	类型	是否必填	说明
record_id	string(64)	是	消息ID（被评价的 reply 事件的消息 ID）
score	uint8	是	评分： 1：点赞 2：点踩
reasons	string 数组	否	所选原因（用户反馈的内容，可以有多个）

3.5 停止生成事件

事件名：stop_generation
事件方向：**前端 > 后端**
数据结构：

名称	类型	是否必填	说明
record_id	string(64)	是	消息ID（需要停止生成的 reply 事件消息 ID）

3.6 参考来源事件

事件名：reference
事件方向：**后端 > 前端**
数据结构：

名称	类型	说明
record_id	string(64)	消息唯一 ID
references	Object 数组	参考来源

references 参考来源的数据结构：

名称	类型	说明
id	string	该 id 使用方法分为两个部分，可参考下面两个示例： 1. 当参考来源类型为1,2,3时，可调用 获取来源详情列表 接口查看参考来源详情。如需预览，可通过上述接口获取对应参数拼接进行跳转访问：https://lke.cloud.tencent.com/preview?id=${docid}&botBizId=${appid}&page=${跳转的doc文档页码}&name=${跳转的sheet名称}&test=1 **说明：** 该 id 字段对应 DescribeRefer 中的 ReferBizIds 字段 2. 当参考来源类型为4的时候，该 id 表示返回多个参考来源的序号
type	uint32	参考来源类型 1：问答 2：文档片段 4：联网检索到的内容 **说明：** 当 type = 2 的时候， 参考来源返回的粒度是文档切片维度的数据，需要配合上述 id 字段使用。 一篇文档【 doc_id 】可能存在多个切片，请自行决定是否需要进行去重展示。
url	string	type = 2 时， 该 url 为文档打开外部引用链接设置的自定义跳转链接地

		址 type = 4 时，该 url 为联网搜索到的信源地址
name	string	参考来源名称
doc_id	uint64	参考来源文档 ID
doc_biz_id	uint64	参考来源文档业务 ID，可调用 文档详情 接口查看对应文档的基础信息
doc_name	string	参考来源文档名称
qa_biz_id	string	参考来源问答业务 ID

示例1：

参考来源类型为：

1：问答

2：文档片段

可调用 获取来源详情列表 接口查看参考来源详情，此时 id 字段对应 DescribeRefer 中的 ReferBizIds 字段。

reference 事件：

```
[
    "reference",
    {
        "type": "reference",
        "payload": {
            "record_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "references": [
                {
                    "doc_biz_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_name": "一百个成语典故及其历史人物故事+ (1).docx",
                    "id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "name": "一百个成语典故及其历史人物故事+ (1)",
                    "qa_biz_id": "0",
                    "type": 2,
                    "url": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
                }
            ],
            "trace_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        },
        "message_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```

```
    }
]
```

示例2：

参考来源类型为：

4：联网检索到的内容

此时，在收到的 thought 事件中或者 reply 事件中，content 字段下的 [3] [4] 表示的就是 references 中联网检索到的来源的下标。

reply 事件(此处只展示，省略了部分结构)：

```
[
    "reply",
    {
        "type": "reply",
        "payload": {
            "can_feedback": true,
            "can_rating": false,
            "content": "\n\n截至2025年2月13日14:59，《哪吒2》（《哪吒之魔童闹海》）总票房（含预售）
            已突破**98亿元**，距离100亿大关仅一步之遥[3][4][7][8]。该票房成绩已超越《黑豹》，
            成为全球影史票房榜前18名中唯一一部非好莱坞电影[3][4][8]。  \n\n根据灯塔专业版预测，《哪吒2》最终总票房可能达到**153.38亿元**，而猫眼预测则给出更高的**160.32亿元**[4]。
            此外，影片海外票房已超过627万元人民币，观影",
        },
        "message_id": "**************************"
    }
]
```

reference 事件：

```
[
    "reference",
    {
        "type": "reference",
        "payload": {
            "record_id": "**************************",
            "references": [
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
```

```
                    "doc_name": "",
                    "id": "1",
                    "name": "破82亿元，进入全球票房榜前31！哪吒2海外首映看哭观
众，有周边上架3天就卖光",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250210A04P8O00"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "2",
                    "name": "刚刚！哪吒2票房破90亿",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250211A07V6J00"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "3",
                    "name": "冲刺百亿票房！《哪吒2》总票房破98亿",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url":
"https://view.inews.qq.com/a/20250213A04F7F00?scene=qqsearch"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "4",
                    "name": "《哪吒2》票房突破98亿元，影视股掀起涨停潮",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url":
"https://view.inews.qq.com/a/20250213A04L4M00?scene=qqsearch"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
```


版权所有：腾讯云计算（北京）有限责任公司

```
                    "doc_name": "",
                    "id": "5",
                    "name": "哪吒2票房破30亿",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250202A03TL500"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "6",
                    "name": "哪吒2票房超去年冠军",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250203A02PAT00"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "7",
                    "name": "《哪吒2》票房突破98亿元",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url":
"https://view.inews.qq.com/a/20250213A04DPH00?scene=qqsearch"
                }

            ],
            "trace_id": "************************"
        },
        "message_id": "************************"
    }
]
```


3.7 错误事件

事件名： error

事件方向： **后端 > 前端**

数据结构：

名称	类型	说明

request_id	string(255)	请求 ID，用于标识一个请求（作消息串联，建议每个请求使用不同的 request_id）
error	Object	错误

error 错误的数据结构：

名称	类型	说明
code	uint32	错误码
message	string	错误信息

3.8 思考事件

事件名：thought
事件方向：**后端 > 前端**

> ⚠ **注意：**
> 该事件目前只在使用 DeepSeek-R1 模型的时候有返回。

数据结构：

名称	类型	说明
elapsed	int	本轮调用耗时, 单位 ms
is_workflow	bool	是否工作流
procedures	Object 数组	调用过程列表
record_id	string(64)	对应发送事件对应的消息记录 id
request_id	string(255)	对应发送事件对应的请求 id
session_id	string(64)	会话id
trace_id	string	链路id
workflow_name	string	工作流名称

procedures 的数据结构：

名称	类型	说明

debugging	Object	调试过程信息
index	uint32	过程索引
name	string	英文名, 与下面的 title 字段一一对应.knowledge, task_flow, search_engine, image, large_language_model, pot_math, file, thought
title	string	调用过程描述, 对应 name 字段, 各中文含义如下:调用知识库, 调用任务流程, 调用搜索引擎, 调用图片理解, 大模型回复, 调用计算器, 阅读文件, 思考
status	string	状态：processing-使用中 success-成功 failed-failed
icon	string	图标, 用于展示
switch	string	是否切换Agent
workflow_name	string	工作流名称
plugin_type	int32	插件类型 0: 自定义插件；1: 官方插件；2: 工作流
elapsed	uint32	当前请求执行时间, 单位 ms

debugging 的数据结构：

名称	类型	说明
content	string	调试过程中输出的内容

效果：

示例：

```
[
    "thought",
    {
        "type": "thought",
        "payload": {
            "elapsed": 0,
            "is_workflow": false,
            "procedures": [
                {
                    "debugging": {
                        "content": "\n好的，用户询问DeepSeek-R1的特点，这已经是第三次了。前两次我已经详细列出了它的高效性能、长上下文处理、多领域通用性、安全合规、多语言支持、灵活部署和高性价比等特点。用户可能是在确认信息，或者希望得到更简洁或不同角度的回答。\n\n我需要考虑用户为什么重复提问。可能的情况包括：用户没有完全理解之前的回答，需要更简明的总结；或者他们想确认信息的准确性；亦或是他们希望了解是否有新的更新。此外，用户可能是在测试回答的一致性，或者他们需要将信息用于特定场景，比如报告或比较不同模
```

型。\n\n接下来，我应该检查之前的回答是否全面，是否有遗漏的重要特点。例如，是否提到了模型的训练数据、支持的框架、具体性能指标等。另外，用户可能对技术细节感兴趣，比如模型结构、训练方法，或者是实际应用"
                    },
                    "elapsed": 9954,
                    "icon": "https://lke-realtime-1251316161.cos.ap-guangzhou.myqcloud.com/icon/thinking.png",
                    "index": 0,
                    "name": "thought",
                    "plugin_type": 0,
                    "status": "processing",
                    "switch": "",
                    "title": "思考",
                    "workflow_name": ""
                }
            ],
            "record_id": "piQ_20250208_140511_254_45ja7HPq",
            "request_id": "rjbuFuYeDB-8487350785",
            "session_id": "dfd04cd0-ef69-4fb6-a447-a0b44018a93f",
            "trace_id": "9a9613dbaa9edd27958bd1180c524295 / piQ_20250208_140511_254_45ja7HPq",
            "workflow_name": ""
        },
        "message_id": "7070ef0d-4c28-4d5a-abc8-2ea2e7fa027d"
    }
]

4. 错误码

错误码	错误信息
400	请求参数错误, 请参阅接入文档
460001	Token 校验失败
460002	事件处理器不存在
460004	应用不存在
460006	消息不存在或没有操作权限
460007	会话创建失败
460008	Prompt 渲染失败

460009	访客用户不存在
460010	会话不存在或没有操作权限
460011	超出并发数限制
460020	模型请求超时
460021	知识库未发布
460022	访客创建失败
460023	消息点赞点踩失败
460024	标签不合法
460025	图像识别失败
460031	当前应用连接数超出请求限制，请稍后再试
460032	当前应用模型余额不足
460033	应用不存在或没有操作权限
460034	输入内容过长
460035	计算内容过长，已经停止
460036	任务流程节点预览参数异常
460037	搜索资源已用尽，调用失败
460038	该 AppID 请求存在异常行为，调用失败

5. 对话端 Demo 代码

Demo 代码描述了一个完整链接建立和消息收发流程。

5.1 前端版本

请查看 前端页面接入（含demo） 。

5.2 后端版本

Golang 版本
Python 版本
JAVA 版本
其他编程语言暂无 Demo，可以参考文档和现有 Demo 自行实现。

对话端接口文档（HTTP SSE）

最近更新时间：2025-06-24 10:14:52

HTTP SSE 是单向通道，客户端发起 HTTP 请求之后，服务端持续推送流式数据到客户端，此时不支持双向交互。

1. HTTP SSE 接口请求

请求地址： `https://wss.lke.cloud.tencent.com/v1/qbot/chat/sse`
请求方式： POST

> ⚠ **注意：**
> 触发对话接口前，需要有已发布的应用。

1.1 参数说明

请放到 HTTP Body 中, 以 JSON 的形式发送，具体如下：

名称	类型	是否必填	说明
request_id	string(255)	否	请求 ID，用于标识一个请求（作消息串联，建议每个请求使用不同的 request_id）。 为了便于问题排查，**建议必填。**
content	string(6000)	是	消息内容，如果发送图片，在此传递 markdown 格式的图片链接，例如，其中图片链接需要可公有读。
file_infos	Object 数组	否	文件信息，如果填写该字段，content 字段可以为空。数据结构请参考 file_infos ，依赖 实时文档解析 。 ⚠ **注意：** 没有文件信息时，该字段可不填或指定为空数组。
session_id	string(64)	是	会话 ID，用于标识一个会话（外部系统提供，建议不同的用户端会话传入**不同的** session_id，否则同一个应用下的不同用户的消息记录会串掉） **参数长度：2-64个字符** **校验规则：** ^[a-zA-Z0-9_-]{2,64}$ ，**一般可以用 uuid 来生成该值**

			uuid 示例：1b9c0b03-dc83-47ac-8394-b366e3ea67ef
bot_app_key	string(128)	是	应用密钥（可参考 1.2 如何获取 AppKey 章节）
visitor_biz_id	string(64)	是	访客 ID（外部输入，建议唯一，标识当前接入会话的用户）
streaming_throttle	int32	否	流式回复频率控制：控制应用回包频率。该值表示应用每积攒多少字符向调用方回包一次，值越小回包越频繁（体验上越流畅，但流量开销也越大）。当不传值或者为 0 时以系统配置为准。 **默认值：5** **建议最大值：100** **注意：该设置项也不会加快大模型输出的时间，只是改变了向调用方回包的频率。因此如果设置很大，则会出现很长时间没有回包的现象。**
custom_variables	map[string]string	否	**工作流场景：** 自定义参数可用于传递参数给工作流开始节点的API参数字段，通过该API参数字段可以被工作流识别并使用，具体参考 开始节点 使用文档。 自定义**API参数**的值。可以配置多个 key: value 对，key 为自定义参数的参数名称，value 为对应的自定义参数的运行时的值。 ⚠ **注意：** map的key和value类型都是string，如{"UserID":"10220022"}， 不合法的参数如： {"UserID": 10220022}； 其中value也可以是json字符串，如 {"Data": "{\"UserID\":\"10220022\",\"Score\":{\"Chinese\":89,\"Math\":98}}"} **检索知识库范围设置场景：** 自定义参数可用于设置知识库检索范围，如需传入多个参数值，使用英文竖线分隔(|)，例如："user1|user2"。 ⚠ **注意：**

			如果需要在问题中指定知识库进行检索，可以通过该字段传入相关的标签值， 具体请参考 知识库检索范围设置 。
system_role	string(4000)	否	角色指令（提示词），为空时使用应用配置默认设定，填写时取当前值。
incremental	bool	否	控制**回复事件**和**思考事件**中的 content 是否是增量输出的内容，默认 false **注意：** 该字段设置为 true，只对回复事件，思考事件的生效。 回复事件效果： 思考事件效果：
search_network	string	否	是否开启联网搜索。取值范围： • 空字符串：跟随应用配置 • enable：打开联网搜索 • disable：关闭联网搜索 **注意：** 应用配置在未开启联网搜索时，如上取值字段无意义。
model_name	string	否	支持在对话的过程传入指定模型， 同一个 session_id 下传入不同的模型名称可以保持上下文之间的关联。 当前支持下面几个自定义模型： • hunyuan：混元大模型高级版 • hunyuan-13B： 混元大模型标准版

			• hunyuan-turbo：混元大模型Turbo版 • hunyuan-standard-256K：混元大模型长文本版 • hunyuan-role：混元大模型角色扮演版 • lke-deepseek-r1：DeepSeek-R1 • lke-deepseek-v3：DeepSeek-V3 • lke-deepseek-r1-0528: DeepSeek-R1-0528 • lke-deepseek-v3：DeepSeek-V3-0324 **⚠ 注意：** 请确保选择的模型有足够的 token 或者开启了后付费模式，避免在使用的过程中报错。
stream	string	否	是否开启流式传输。取值范围： • 空字符串：跟随应用配置 • enable：流式传输 • disable：非流式传输
workflow_status	string	否	是否开启工作流。取值范围： • 空字符串：跟随应用配置（页面无配置工作流开关的入口，默认是开启） • enable：开启工作流 • disable：关闭工作流

visitor_labels 知识标签列表的数据结构：

名称	类型	说明
name	string	知识标签名
values	string 数组	知识标签值

file_infos 文件信息的数据结构：

名称	类型	是否必填	说明
file_name	string	是	文件名称
file_size	string	是	实时文档解析接口返回的文件大小
file_url	string	是	实时文档解析接口返回的文件 URL

file_type	string	是	文件类型
doc_id	string	是	实时文档解析接口返回的 doc_id

1.2 如何获取 AppKey

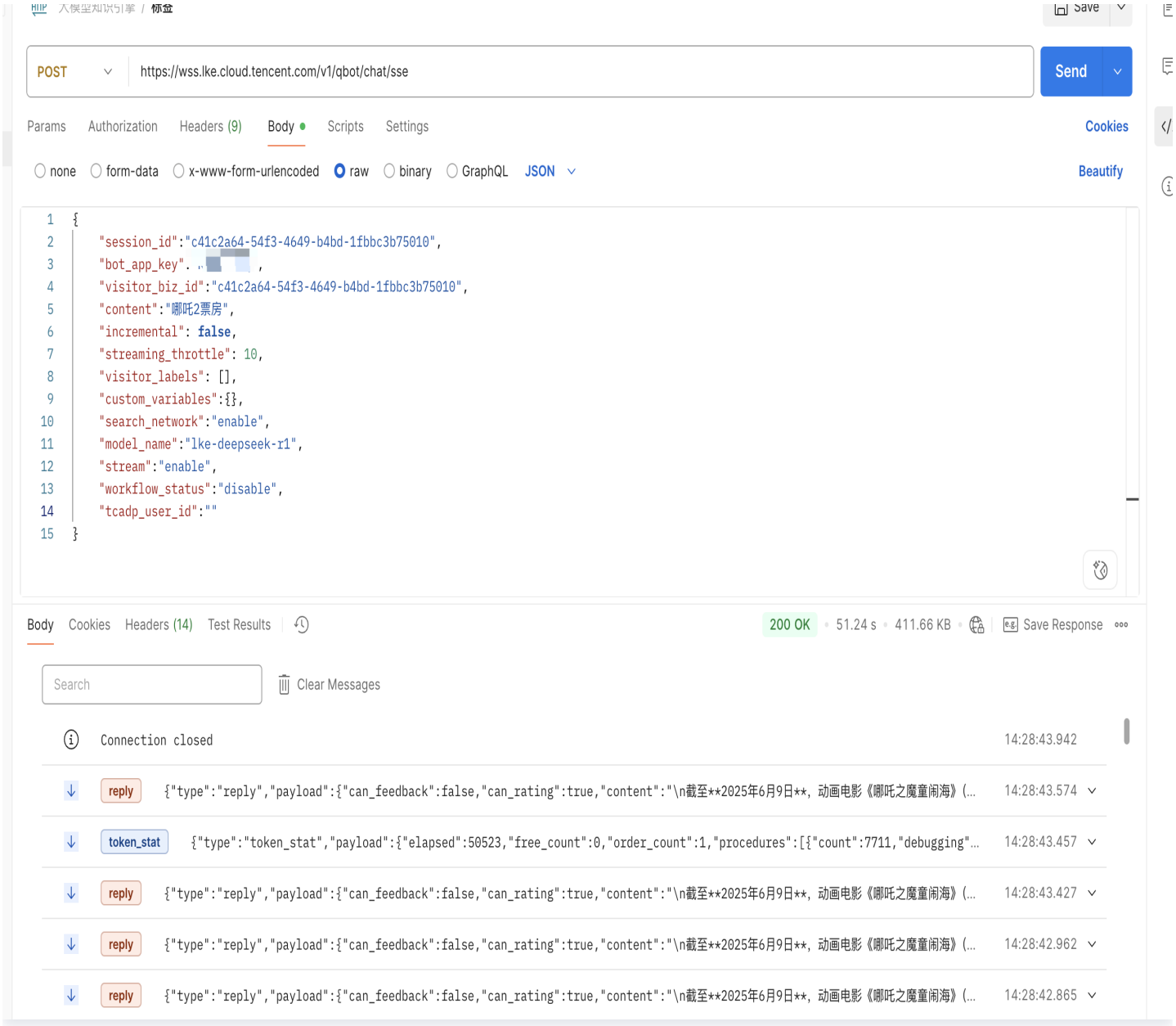
在应用管理界面，找到您处于运行中的应用（需要先发布），单击调用，会弹出“调用信息”窗口。

在调用信息窗口可以看到 AppKey，单击复制即可。

1.3 curl 调用示例

```
curl --location 'https://wss.lke.cloud.tencent.com/v1/qbot/chat/sse' \
--header 'Content-Type: application/json' \
--data '{
    "session_id":"a29bae68-cb1c-489d-8097-6be78f136acf",
    "bot_app_key":"请自行获取应用对应的key",
    "visitor_biz_id":"a29bae68-cb1c-489d-8097-6be78f136acf",
    "content":"哪吒2票房",
    "incremental": true,
    "streaming_throttle": 10,
    "visitor_labels": [],
    "custom_variables":{},
    "search_network":"disable",
    "stream":"enable",
    "workflow_status":"disable",
    "tcadp_user_id":""
}'
```

1.4 postman 调用示例



2. HTTP SSE 接口返回

2.1 回复事件

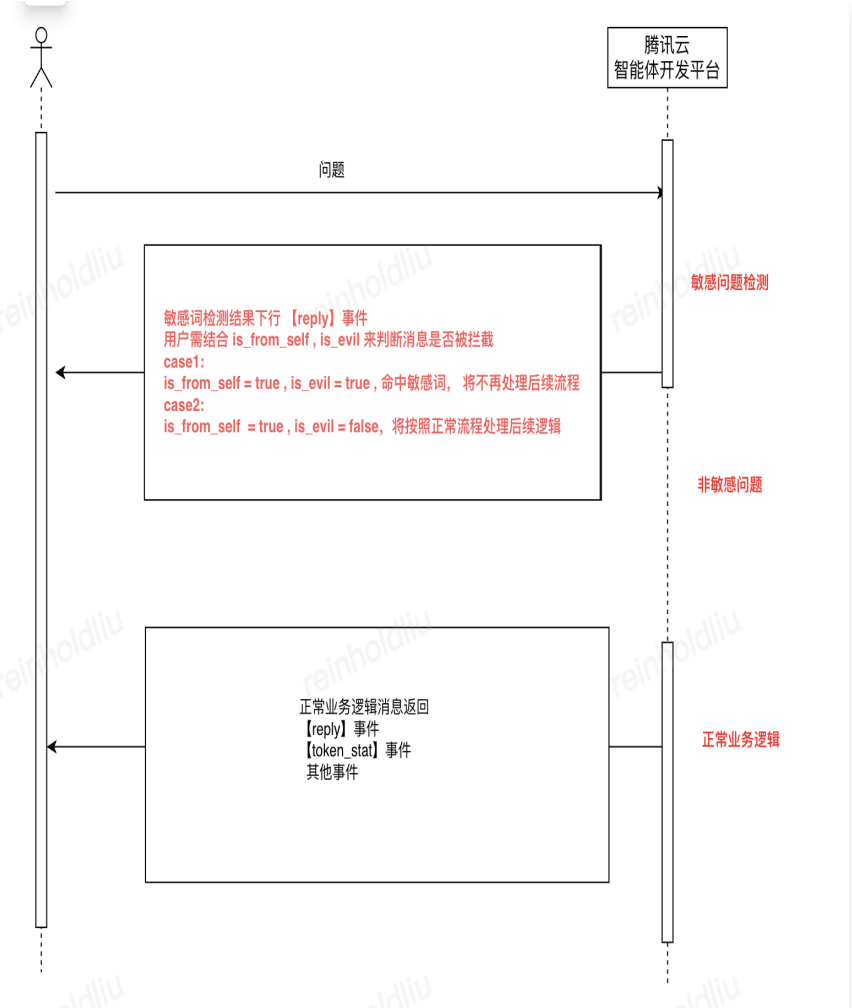
事件名：reply

事件方向：**后端 > 前端**

> **注意：**
>
> - 如果收到的消息中 is_evil == true 表示该消息命中敏感内容，发送失败。
> - 因并发超限导致排队超时，会下发 "超出并发数限制" 错误。

数据结构：

名称	类型	说明
request_id	string(255)	请求 ID，用于标识一个请求（作消息串联，建议每个请求使用不同的 request_id）
content	string	消息内容
file_infos	Object 数组	文件信息
record_id	string(64)	消息唯一 ID
related_record_id	string(64)	关联的消息唯一 ID
session_id	string(64)	会话 ID，用于标识一个会话（外部系统提供，建议不同的用户端会话传入**不同的** session_id，否则同一个应用下的不同用户的消息记录会串掉）
is_from_self	bool	消息是否由自己发出 （如果是自己发出，显示在聊天框右侧，否则在左侧）
can_rating	bool	该消息记录是否能评价
timestamp	int64	消息时间戳（秒级）
is_final	bool	reply 事件消息是否已输出完 • 流式模式下，消息会多次返回，每次返回均覆盖之前的答案。 • 当 is_final == true 时，停止生成按钮隐藏，并且显示点赞点踩按钮。 • reference 事件和 reply 事件无固定顺序关系，不建议以该字段作为消息完整结束的标识。
is_evil	bool	是否命中敏感内容 **注意：消息上行后会先进行敏感内容检测，并返回一条【reply】事件，告知敏感内容检测结果， 敏感问题检测通过后才会进入正常业务逻辑处理**

		
is_llm_generated	bool	是否为模型生成内容
reply_method	uint8	回复方式： 1: 大模型回复 2: 未知问题回复 3: 拒答问题回复 4: 敏感回复 5: 已采纳问答对优先回复 6: 欢迎语回复 7: 并发数超限回复 8: 全局干预知识 9: 任务流回复 10: 任务流答案 11: 搜索引擎回复 12: 知识润色后回复 13: 图片理解回复 14: 实时文档回复 15: 澄清确认回复 16: 工作流回复

		17: 工作流运行结束 18: 智能体回复 19: 多意图回复
knowledge	Object 数组	命中的知识
option_cards	string 数组	选项卡，任务流程专有 注：该字段可能为空，为空时该字段不返回
custom_params	string 数组	用户自定义业务参数，用于透传问答中业务参数 注：该字段可能为空，为空时该字段不返回
task_flow	Object	任务流程调试信息 注：该字段可能为空，为空时该字段不返回
work_flow	Object	工作流调试信息 说明：该字段可能为空，为空时该字段不返回
quote_infos	Object 数组	引用信息

knowledge 命中的知识的数据结构：

名称	类型	说明
id	string	命中的知识 ID
type	uint32	命中的知识类型： 1: 问答 2: 文档片段
seg_id	string	片段 ID，可通过 DescribeSegments 接口查看片段详情

work_flow 工作流调试信息的数据结构：

名称	类型	说明
workflow_name	string	工作流名称
workflow_id	string	工作流ID
qworkflow_run_id	string	工作流运行ID
option_cards	string 数组	选项卡

版权所有：腾讯云计算（北京）有限责任公司

current_node	Object	当前节点信息
outputs	string	输出结果

current_node 当前工作流节点信息：

名称	类型	说明
NodeID	string	节点ID
NodeType	int32	节点类型： 0：未指定 1： 开始节点 2：参数提取节点 3：大模型节点 4：知识问答节点 5：知识检索节点 6：标签提取节点 7：代码执行节点 8：工具节点 9：逻辑判断节点 10：回复节点 → 消息节点 11：选项卡节点 12：循环节点 13：意图识别节点 14：工作流节点 15：插件节点 16：结束节点
NodeName	string	节点名称
Status	int32	状态: 0：初始状态 1： 运行中 2：运行成功 3：运行失败
Input	string	节点的输入。json字符串（含普通字符串）
Output	string	节点的最终输出。json字符串（含普通字符串）
TaskOutput	string	任务的输出。（原始输出）
FailMessage	string	异常信息

CostMilliSeconds	string	节点的总耗时。如果节点有多次调用，耗时为多次调用的总和。
Reply	string	当前节点的回复内容。当前参数提取、消息节点、结束回复节点可能非空
BelongNodeID	string	节点所属工作流被引用时的引用节点的ID，所属的工作流是被引用的时候非空。
StatisticInfo	Object	LLM 统计信息

StatisticInfo LLM 统计信息：

名称	类型	说明
ModelName	string	模型名称
FirstTokenCost	uint32	首token耗时
TotalCost	uint32	推理总耗时
InputTokens	uint32	输入token数量
OutputTokens	uint32	输出token数量
TotalTokens	uint32	输入+输出总token

quote_infos 引用信息数据结构：

名称	类型	说明
index	int	引用的角标索引， 角标 icon 请接入方自行设计实现
position	int	在回复内容 content 中的位置(中文字符，英文字符，其他符号都算一个字符)

效果：

quote_infos 示例 (此处只展示，省略了部分结构)：

```
[
    "reply",
    {
        "type": "reply",
        "payload": {
            "can_feedback": true,
            "can_rating": false,
            "content": "2025年2月13日腾讯股价情况如下：\n\n### 腾讯控股（00700.HK）港股市场表现\n\n- **股价波动**：腾讯控股股价在2025年2月13日经历了显著的波动，盘中振幅达到4.35%。这一波动既反映了市场情绪与资金博弈的短期影响，也体现了行业环境与公司基本面的长期逻辑。\n- **股价范围**：腾讯股价在当天最高达到464.6港币，最低跌至437港币。\n\n### 腾讯控股（TCTZF.us）美股市场表现\n\n- **股价上涨**：在美股市场，腾讯控股（TCTZF.us）股价在2025年2月13日盘中一度上涨5.05%，最新报58.10美元/股。\n- **市场反应**：这一异动引发了市场的广泛关注，投资者纷纷开始挖掘其背后的原因及影响。\n\n请注意，以上信息仅供参考，投资有风险，入市请谨慎。",

            "quote_infos": [
                {
                    "index": 1,
                    "position": 323
                }
```

```
            ],

        },
        "message_id": "xxxxxxxxxxxxxxxxxxxxxx"
    }
]
```

2.2 token 统计事件

事件名：token_stat
事件方向：后端 > 前端
数据结构：

名称	类型	说明
session_id	string(64)	会话 id
request_id	string(255)	对应发送事件对应的请求 id
record_id	string(64)	对应发送事件对应的消息记录 id
status_summary	string	本轮对话状态, 处理中: processing, 成功: success, 失败: failed
status_summary_title	string	本轮对话状态描述
elapsed	int	本轮调用耗时, 单位 ms
token_count	int	本轮请求消耗 token 数(当包含多个过程时, 计算将汇总)
procedures	Object 数组	调用过程列表

procedures 调用过程列表的数据结构：

名称	类型	说明
name	string	英文名, 与下面的 title 字段一一对应. knowledge, task_flow, search_engine, image, large_language_model, pot_math, file
title	string	调用过程描述, 对应 name 字段, 各中文含义如下: 调用知识库, 调用任务流程, 调用搜索引擎, 调

		用图片理解, 大模型回复, 调用计算器, 阅读文件
status	string	调用过程状态, 处理中: processing, 成功: success, 失败: failed
input_count	int	当次过程输入消耗 token 数
output_count	int	当次过程输出消耗 token 数
count	int	当次过程消耗总 token 数：输入 + 输出

示例:


```
[
    "token_stat",
    {
        "type": "token_stat",
        "payload": {
            "elapsed": 1616,
            "order_count": 50000000,
            "procedures": [
                {
                    "count": 323,
                    "input_count": 308,
                    "name": "knowledge",
                    "output_count": 15,
                    "status": "success",
                    "title": "调用知识库"
                }
            ],
            "record_id": "Hpe_20240625_185659_215_EsH2uf8L",
            "request_id": "8PUcDU6xyQ-301747294000",
            "session_id": "2d071ef7-ef76-44df-84a4-9210672ed700c8",
            "status_summary": "success",
            "status_summary_title": "调用知识库",
            "token_count": 323,
            "used_count": 553
        },
        "message_id": "89d91395-06bc-4f2e-b240-06f7b4498b0c6e"
    }
]
```


2.3 参考来源事件

事件名：reference
事件方向：**后端 > 前端**
数据结构：

名称	类型	说明
record_id	string(64)	消息唯一 ID
references	Object 数组	参考来源

references 参考来源的数据结构：

名称	类型	说明
id	string	该 id 使用方法分为两个部分，可参考下面两个示例： 1. 当参考来源类型为1,2,3时，可调用 获取来源详情列表 接口查看参考来源详情。如需预览，可通过上述接口获取对应参数拼接进行跳转访问：https://lke.cloud.tencent.com/preview?id=${docid}&botBizId=${appid}&page=${跳转的doc文档页码}&name=${跳转的sheet名称}&test=1 **说明：** 该 id 字段对应 DescribeRefer 中的 ReferBizIds 字段 2. 当参考来源类型为4的时候，该 id 表示返回多个参考来源的序号
type	uint32	参考来源类型 1：问答 2：文档片段 4：联网检索到的内容 **说明：** 当 type = 2 的时候， 参考来源返回的粒度是文档切片维度的数据，需要配合上述 id 字段使用。 一篇文档【

		doc_id 】可能存在多个切片，请自行决定是否需要进行去重展示。
url	string	type = 2 时， 该 url 为文档打开外部引用链接设置的自定义跳转链接地址 type = 4 时，该 url 为联网搜索到的信源地址
name	string	参考来源名称
doc_id	uint64	参考来源文档 ID
doc_biz_id	uint64	参考来源文档业务 ID， 可调用 文档详情 接口查看对应文档的基础信息
doc_name	string	参考来源文档名称
qa_biz_id	string	参考来源问答业务 ID

示例1:

参考来源类型为:

1：问答

2：文档片段

可调用 获取来源详情列表 接口查看参考来源详情, 此时 id 字段对应 DescribeRefer 中的 ReferBizIds 字段。

reference 事件：


```
[
    "reference",
    {
        "type": "reference",
        "payload": {
            "record_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "references": [
                {
                    "doc_biz_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "doc_name": "一百个成语典故及其历史人物故事+ (1).docx",
                    "id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                    "name": "一百个成语典故及其历史人物故事+ (1)",
                    "qa_biz_id": "0",
                    "type": 2,
                    "url": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
                }
            ],
```

```
            "trace_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        },
        "message_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }
]
```

示例2：

参考来源类型为：

4：联网检索到的内容

此时，在收到的 thought 事件中或者 reply 事件中，content 字段下的 [3] [4] 表示的就是 references 中联网检索到的来源的下标。

reply 事件(此处只展示，省略了部分结构)：

```
[
    "reply",
    {
        "type": "reply",
        "payload": {
            "can_feedback": true,
            "can_rating": false,
            "content": "\n\n截至2025年2月13日14:59，《哪吒2》（《哪吒之魔童闹
海》）总票房（含预售）
            已突破**98亿元**，距离100亿大关仅一步之遥[3][4][7][8]。该票房成绩已
超越《黑豹》，
            成为全球影史票房榜前18名中唯一一部非好莱坞电影[3][4][8]。  \n\n根据灯
塔专业版预测，《哪吒2》最终总票房可能达到**153.38亿元**，而猫眼预测则给出更高的
**160.32亿元**[4]。
            此外，影片海外票房已超过627万元人民币，观影",
        },
        "message_id": "**************************"
    }
]
```

reference 事件：

```
[
    "reference",
    {
        "type": "reference",
        "payload": {
            "record_id": "**************************",
            "references": [
```

```
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "1",
                    "name": "破82亿元，进入全球票房榜前31！哪吒2海外首映看哭观
众，有周边上架3天就卖光",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250210A04P8O00"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "2",
                    "name": "刚刚！哪吒2票房破90亿",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250211A07V6J00"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "3",
                    "name": "冲刺百亿票房！《哪吒2》总票房破98亿",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url":
"https://view.inews.qq.com/a/20250213A04F7F00?scene=qqsearch"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "4",
                    "name": "《哪吒2》票房突破98亿元，影视股掀起涨停潮",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url":
"https://view.inews.qq.com/a/20250213A04L4M00?scene=qqsearch"
                },
```

```
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "5",
                    "name": "哪吒2票房破30亿",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250202A03TL500"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "6",
                    "name": "哪吒2票房超去年冠军",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url": "https://new.qq.com/rain/a/20250203A02PAT00"
                },
                {
                    "doc_biz_id": "0",
                    "doc_id": "0",
                    "doc_name": "",
                    "id": "7",
                    "name": "《哪吒2》票房突破98亿元",
                    "qa_biz_id": "0",
                    "type": 4,
                    "url":
"https://view.inews.qq.com/a/20250213A04DPH00?scene=qqsearch"
                }

            ],
            "trace_id": "**************************"
        },
        "message_id": "**************************"
    }
]
```


2.4 错误事件

事件名： error

事件方向： **后端 > 前端**

数据结构：

名称	类型	说明
request_id	string(255)	请求 ID，用于标识一个请求（作消息串联，建议每个请求使用不同的request_id）
error	Object	错误

error 错误的数据结构：

名称	类型	说明
code	uint32	错误码
message	string	错误信息

示例：

```
event:reply
data:{"type":"reply","payload":{"can_rating":false,"content":"你是
谁","from_avatar":"","from_name":"","is_evil":false,"is_final":true,"is_f
rom_self":true,"is_llm_generated":false,"knowledge":null,"record_id":"83
ecd23c-6283-48d0-ac5e-
7d8ab604770d","related_record_id":"","reply_method":0,"request_id":"","s
ession_id":"sse_session8","timestamp":1701330804,"trace_id":"5daf1726c25
4241810bb160b4c8efbed"},"message_id":"808727d6-260a-4b7f-8a70-
99330feaef3f"}

event:reply
data:{"type":"reply","payload":{"can_rating":true,"content":"我是智能体开发
平台，能够回答各种问题和提供信息。","from_avatar":"https://qbot-
1251316161.cos.ap-
nanjing.myqcloud.com/avatar.png","from_name":"bot","is_evil":false,"is_f
inal":true,"is_from_self":false,"is_llm_generated":true,"knowledge":
[{"id":33386,"type":1},{"id":452,"type":1},
{"id":33388,"type":1}],"record_id":"7cfaf2dc-8e95-475b-9aa5-
d6a5d4358f71","related_record_id":"83ecd23c-6283-48d0-ac5e-
7d8ab604770d","reply_method":1,"request_id":"","session_id":"sse_session
8","timestamp":1701330805,"trace_id":"5daf1726c254241810bb160b4c8efbed"}
,"message_id":"21b3eb5b-b0eb-4a2c-907b-2a287ad26a34"}
```

```
event:error
```

```
data:{"type":"error","error":{"code":460004,"message":"应用不存在"}}
```

> **注意：**
> 接口使用时需判断取值是否为200，是则正常返回。

2.5 思考事件

事件名：thought
事件方向：**后端 > 前端**

> **注意：**
> 该事件目前只在使用 DeepSeek-R1 模型的时候有返回。

数据结构：

名称	类型	说明
elapsed	int	本轮调用耗时, 单位 ms
is_workflow	bool	是否工作流
procedures	Object 数组	调用过程列表
record_id	string(64)	对应发送事件对应的消息记录 id
request_id	string(255)	对应发送事件对应的请求 id
session_id	string(64)	会话 id
trace_id	string	链路 id
workflow_name	string	工作流名称

procedures 的数据结构：

名称	类型	说明
debugging	Object	调试过程信息
index	uint32	过程索引
name	string	英文名, 与下面的 title 字段一一对应.knowledge, task_flow, search_engine, image,

		large_language_model, pot_math, file, thought
title	string	调用过程描述, 对应 name 字段, 各中文含义如下:调用知识库, 调用任务流程, 调用搜索引擎, 调用图片理解, 大模型回复, 调用计算器, 阅读文件, 思考
status	string	状态：processing- 使用中 success- 成功 failed-failed
icon	string	图标, 用于展示
switch	string	是否切换Agent
workflow_name	string	工作流名称
plugin_type	int32	插件类型 0: 自定义插件；1: 官方插件；2: 工作流
elapsed	uint32	当前请求执行时间, 单位 ms

debugging 的数据结构：

名称	类型	说明
content	string	调试过程中输出的内容

效果：

示例：

```
[
    "thought",
    {
        "type": "thought",
        "payload": {
            "elapsed": 0,
            "is_workflow": false,
            "procedures": [
                {
                    "debugging": {
                        "content": "\n好的，用户询问DeepSeek-R1的特点，这已经是第三次了。前两次我已经详细列出了它的高效性能、长上下文处理、多领域通用性、安全合规、多语言支持、灵活部署和高性价比等特点。用户可能是在确认信息，或者希望得到更简洁或不同角度的回答。\n\n我需要考虑用户为什么重复提问。可能的情况包括：用户没有完全理解之前的回
```

答，需要更简明的总结；或者他们想确认信息的准确性；亦或是他们希望了解是否有新的更新。此外，用户可能是在测试回答的一致性，或者他们需要将信息用于特定场景，比如报告或比较不同模型。\n\n接下来，我应该检查之前的回答是否全面，是否有遗漏的重要特点。例如，是否提到了模型的训练数据、支持的框架、具体性能指标等。另外，用户可能对技术细节感兴趣，比如模型结构、训练方法，或者是实际应用"
                    },
                    "elapsed": 9954,
                    "icon": "https://lke-realtime-1251316161.cos.ap-guangzhou.myqcloud.com/icon/thinking.png",
                    "index": 0,
                    "name": "thought",
                    "plugin_type": 0,
                    "status": "processing",
                    "switch": "",
                    "title": "思考",
                    "workflow_name": ""
                }
            ],
            "record_id": "piQ_20250208_140511_254_45ja7HPq",
            "request_id": "rjbuFuYeDB-8487350785",
            "session_id": "dfd04cd0-ef69-4fb6-a447-a0b44018a93f",
            "trace_id": "9a9613dbaa9edd27958bd1180c524295 / piQ_20250208_140511_254_45ja7HPq",
            "workflow_name": ""
        },
        "message_id": "7070ef0d-4c28-4d5a-abc8-2ea2e7fa027d"
    }
]

3. 错误码

错误码	错误信息
400	请求参数错误, 请参阅接入文档
460001	Token 校验失败
460002	事件处理器不存在
460004	应用不存在
460006	消息不存在或没有操作权限
460007	会话创建失败

460008	Prompt 渲染失败
460009	访客用户不存在
460010	会话不存在或没有操作权限
460011	超出并发数限制
460020	模型请求超时
460021	知识库未发布
460022	访客创建失败
460023	消息点赞点踩失败
460024	标签不合法
460025	图像识别失败
460031	当前应用连接数超出请求限制，请稍后再试
460032	当前应用模型余额不足
460033	应用不存在或没有操作权限
460034	输入内容过长
460035	计算内容过长, 已经停止
460036	任务流程节点预览参数异常
460037	搜索资源已用尽，调用失败
460038	该 AppID 请求存在异常行为，调用失败

4. 对话端 Demo 代码

4.1 前端版本

请查看 前端页面接入（含demo） 。

4.2 后端版本

Golang版本
Python版本
JAVA版本
其他编程语言暂无 Demo，可以参考文档和现有 Demo 自行实现。

图片对话或文件对话（实时文档解析+对话）

最近更新时间：2025-05-29 10:48:42

1 整体介绍

在对话过程中，可以上传图片进行问答，也可以上传文档进行文件总结和内容提取。
图片问答：

文件问答：

说明：

- 随着产品能力的迭代，对文件类型，文件大小的限制会逐步放开，请参考智能体开发平台页面上的限制。
- 文档问答需进行文档解析后，填充参数，使用 WebSocket 或者 HTTP SSE 进行对话 。

2 图片问答实现步骤

1. 调用 DescribeStorageCredential 接口，获取临时密钥。
2. 通过临时密钥，通过 COS 的SDK上传到 COS。

注意：
智能体开发平台操作 COS 的方法与直接使用固定密钥操作 COS 的方法不一样。详细步骤，请参见 智能体开发平台操作 cos 指南 。

3. 填充参数 使用 WebSocket 或者 HTTP SSE 进行对话。

3.1 图片 COS 地址的拼接方法如下：

```
https:// +
 {DescribeStorageCredential.Response.Bucket} +
 "." +
 {DescribeStorageCredential.Response.Type} +
 "." +
 {DescribeStorageCredential.Response.Region} +
 ".myqcloud.com" +
 {DescribeStorageCredential.Response.UploadPath}
```

如果发送图片，在此传递 markdown 格式的图片链接，例如 `![](图片连接)` ，请注意获取临时密钥的时候 IsPublic 字段。

入参示例：

一张图片

```
{
    "request_id": "LHWObW2Sea-3173301679",
    "session_id": "f5652b8c-e88e-4abc-8629-7952b741a433",
    "bot_app_key":"xxxxxx",
    "visitor_biz_id":"LHWObW2Sea",
    "file_infos": [],
    "content": "描述下下面这张图片![](https://lke-realtime-
1251316161.cos.ap-
guangzhou.myqcloud.com/public/1746827241600319488/1780784842443587584/im
age/xxxx.jpg)"
 }
```

多张图片

```
{
    "request_id": "LHWObW2Sea-317330dwe9",
    "session_id": "f5652b8c-e88e-4111-8629-7952b741a433",
    "bot_app_key":"xxxxxx",
    "visitor_biz_id":"LHWObW2Sea",
    "file_infos": [],
    "content": "描述下下面这张图片![](https://lke-realtime-
1251316161.cos.ap-
guangzhou.myqcloud.com/public/1746827241600319488/1780784842443587584/im
age/xxxx.jpg)![](https://lke-realtime-1251316161.cos.ap-
```

```
guangzhou.myqcloud.com/public/17530304859406336O0/1887842680054218752/im
age/bbbbb.png)"
 }
```

3 文档问答实现步骤

在知识问答中，如果要上传文档进行实时问答，需要先对接实时文档解析接口。过程如下：

1. 调用 DescribeStorageCredential 接口，获取临时密钥。
2. 通过临时密钥，通过 COS 的 SDK 上传到 COS。

> ⚠ **注意：**
> 智能体开发平台操作 COS 的方法与直接使用固定密钥操作 COS 的方法不一样。详细步骤，请参见 智能体开发平台操作 cos 指南 。

3. 调用实时文档解析 docParse 接口，获取 doc_id。
4. 填充参数 使用 WebSocket 或者 HTTP SSE 进行对话。

3.1 实时文档解析

实时文档解析接口地址： `https://wss.lke.cloud.tencent.com/v1/qbot/chat/docParse`
请求方式：POST

3.1.1 请求参数

请放到 HTTP Body 中, 以 JSON 的形式发送，具体如下：

名称	类型	是否必填	说明
session_id	string(64)	是	会话 ID，用于标识一个会话（外部系统提供，建议不同的用户端会话传入**不同的**session_id，否则同一个应用下的不同用户的消息记录会串掉） **参数长度：2-64个字符** **校验规则： ^[a-zA-Z0-9_-]{2,64}$ ，一般可以用 uuid 来生成该值** **uuid 示例：1b9c0b03-dc83-47ac-8394-b366e3ea67ef** > ⚠ **注意：** 文档解析的 session_id 要与会话的保持一致。如果一次多轮会话涉及多次上传文档，该 session_id 也要保持一致，会话时会校验。

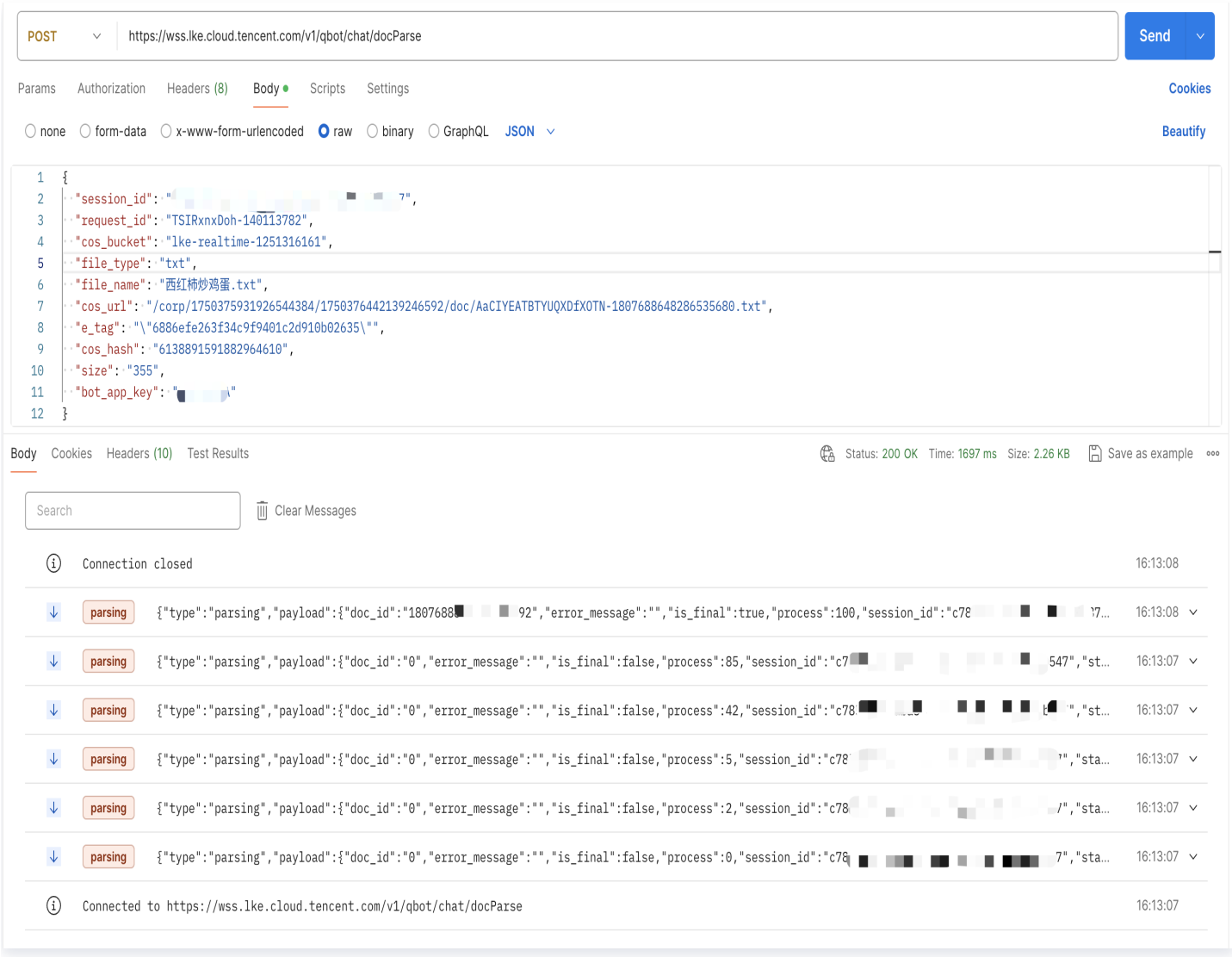
bot_app_key	string(8)	是	应用密钥（运营提供）
request_id	string(255)	是	请求的唯一 ID，建议使用 UUID 保证唯一性。
cos_bucket	string	是	cos 桶， 请从获取临时密钥接口返回中获取
file_type	string	是	文件类型(txt|doc|docx|pdf|ppt|pptx) 示例值：txt|doc|docx|pdf|ppt|pptx
file_name	string	是	文件名 示例值：测试.docx
cos_url	string	是	平台 cos 路径，与 DescribeStorageCredential 接口查询 UploadPath 参数保持一致 示例 值：/corp/17503759319265443 84/1750376442139246592/doc/AaCIYEATBTYUQXDfXOTN-1807688648286535680.txt
cos_hash	string	是	cos_hash x-cos-hash-crc64ecma 头部中的 CRC64 编码进行校验上传到云端的文件和本地文件的一致性；上传 cos 成功之后，从返回头中获取
e_tag	string	是	ETag 全称为 Entity Tag，是对象被创建时标识对象内容的信息标签，可用于检查对象的内容是否发生变化， 上传 cos 成功之后， 从返回头中获取 示例值：\"6886efe263f34c9f9401c2d910b02635\"
size	**string**	是	上传文档的大小 ⚠ **注意：** 请注意，该字段为 string 类型。

3.1.2 curl 调用示例

```
curl --location
'https://wss.lke.cloud.tencent.com/v1/qbot/chat/docParse' \
--header 'Content-Type: application/json' \
```

```
--data '{
  "session_id": "<your session_id>",
  "request_id": "<random uuid>",
  "cos_bucket": "lke-realtime-1251316161",
  "file_type": "docx",
  "file_name": "测试.docx",
  "cos_url":
"/corp/1750375931926544384/1750376442139246592/doc/AaCIYEATBTYUQXDfXOTN-
1807688648286535680.txt",
  "e_tag": "\"6886efe263f34c9f9401c2d910b02635\"",
  "cos_hash": "6138891591882964610",
  "size": "355",
  "bot_app_key": "<your appkey>"
}'
```

3.1.3 postman 调用示例



3.1.4 响应参数

SSE 流式返回

名称	类型	说明
session_id	string(64)	会话 ID，同发起请求时的 session_id
trace_id	string	返回的唯一 ID
is_final	bool	消息是否已输出完成
doc_id	string	文档解析接口返回的 doc_id
process	int32	当前进度，整数，值为100时表示成功结束

status	string	状态：PARSING、SUCCESS、FAILED
timestamp	int64	时间戳，单位：秒
error_message	string	错误信息，出错时返回

3.1.5 返回示例


```
{"type":"parsing","payload":
{"doc_id":"0","error_message":"","is_final":false,"process":0,"session_i
d":"c7852s9d-aba8-4ee8-9c88-
d65f28ddbc47","status":"PARSING","timestamp":1719821535,"trace_id":"1f1e
5bfc9a3588d3abc62b9729fc6f62"},"message_id":"1b28b359-203e-4dbc-a103-
6d92629cb1e0"}

{"type":"parsing","payload":
{"doc_id":"0","error_message":"","is_final":false,"process":2,"session_i
d":"c7852s9d-aba8-4ee8-9c88-
d65f28ddbc47","status":"PARSING","timestamp":1719821535,"trace_id":"1f1e
5bfc9a3588d3abc62b9729fc6f62"},"message_id":"60c2a29a-7658-4186-90a9-
d81c8c0b14b4"}

{"type":"parsing","payload":
{"doc_id":"0","error_message":"","is_final":false,"process":85,"session_
id":"c7852s9d-aba8-4ee8-9c88-
d65f28ddbc47","status":"PARSING","timestamp":1719821536,"trace_id":"1f1e
5bfc9a3588d3abc62b9729fc6f62"},"message_id":"65ca6da3-8909-42c4-9ea1-
4a09be299a7b"}

{"type":"parsing","payload":
{"doc_id":"1807688654434383264","error_message":"","is_final":true,"proc
ess":100,"session_id":"c7852s9d-aba8-4ee8-9c88-
d65f28ddbc47","status":"SUCCESS","timestamp":1719821536,"trace_id":"1f1e
5bfc9a3588d3abc62b9729fc6f62"},"message_id":"43046854-c596-45f5-9195-
3df4f82a67ff"}
```


3.2 文档对话参数示例

文件 COS 地址的拼接方法如下：

```
  https:// +
   {DescribeStorageCredential.Response.Bucket} +
```

```
    "." +
    {DescribeStorageCredential.Response.Type} +
    "." +
    {DescribeStorageCredential.Response.Region} +
    ".myqcloud.com" +
    {DescribeStorageCredential.Response.UploadPath}
```

不带 content 的实时文档：

```
{
    "payload": {
        "request_id": "LkfnMf5IrS-4486169698",
        "session_id": "c60f0463-0176-4c15-a236-e4cbb6c21f97",
        "file_infos": [
            {
                "doc_id": "1833044072669409152",
                "file_name": "河边的美景",
                "file_type": "docx",
                "file_size": "34859",
                "file_url": "https://lke-realtime-1251316161.cos.ap-
guangzhou.myqcloud.com/corp/1753030485940633600/1781229014471147520/doc/
xxxxxxx.docx"
            }
        ],
        "content": ""
    }
}
```

带 content 的实时文档：

```
{
    "payload": {
        "request_id": "XCBLyiWwYV-9639940128",
        "session_id": "c60f0463-0176-4c15-a236-e4cbb6c21f97",
        "file_infos": [
            {
                "doc_id": "1833044598014003968",
                "file_name": "河边的美景",
                "file_type": "docx",
                "file_size": "34859",
                "file_url": "https://lke-realtime-1251316161.cos.ap-
guangzhou.myqcloud.com/corp/1753030485940633600/1781229014471147520/doc/
```

```
xxxxxxdocx"
                }
            ],
            "content": "提取文档中的主要观点"
        }
}
```

3.3 . 特别说明

文档解析完成后，需要组装成 WebSocket 或者 HTTP SSE 对话时的 file_infos 字段，然后调用对话即可使用， 下面是几个简单 file_infos 字段示例。

> **注意：**
>
> 1. doc_id 和 file_size 是字符串类型，如果传入整型会报参数错误。
> 2. file_name 不要加文件后缀。
> 3. 对话时候的 session_id 请和调用 docParse 接口传入的 session_id 保持一致， 否则会导致检索不到传入的文件。
> 4. 实时文档解析上传的文档有会话轮次限制（3轮），并且有时效性（24h），请勿反复使用同一个文档 id 问答。

4 图片问答，文档问答 Demo

Golang 版本
Python 版本
Java 版本

> **说明：**
>
> - 其他编程语言暂无 Demo，可以参考文档和现有 Demo 自行实现。
> - Demo 的目的为快速验证， 使用的是 HTTP SSE 的对话方式，用户可结合实际业务需求，选择 Websocket 对话方式。

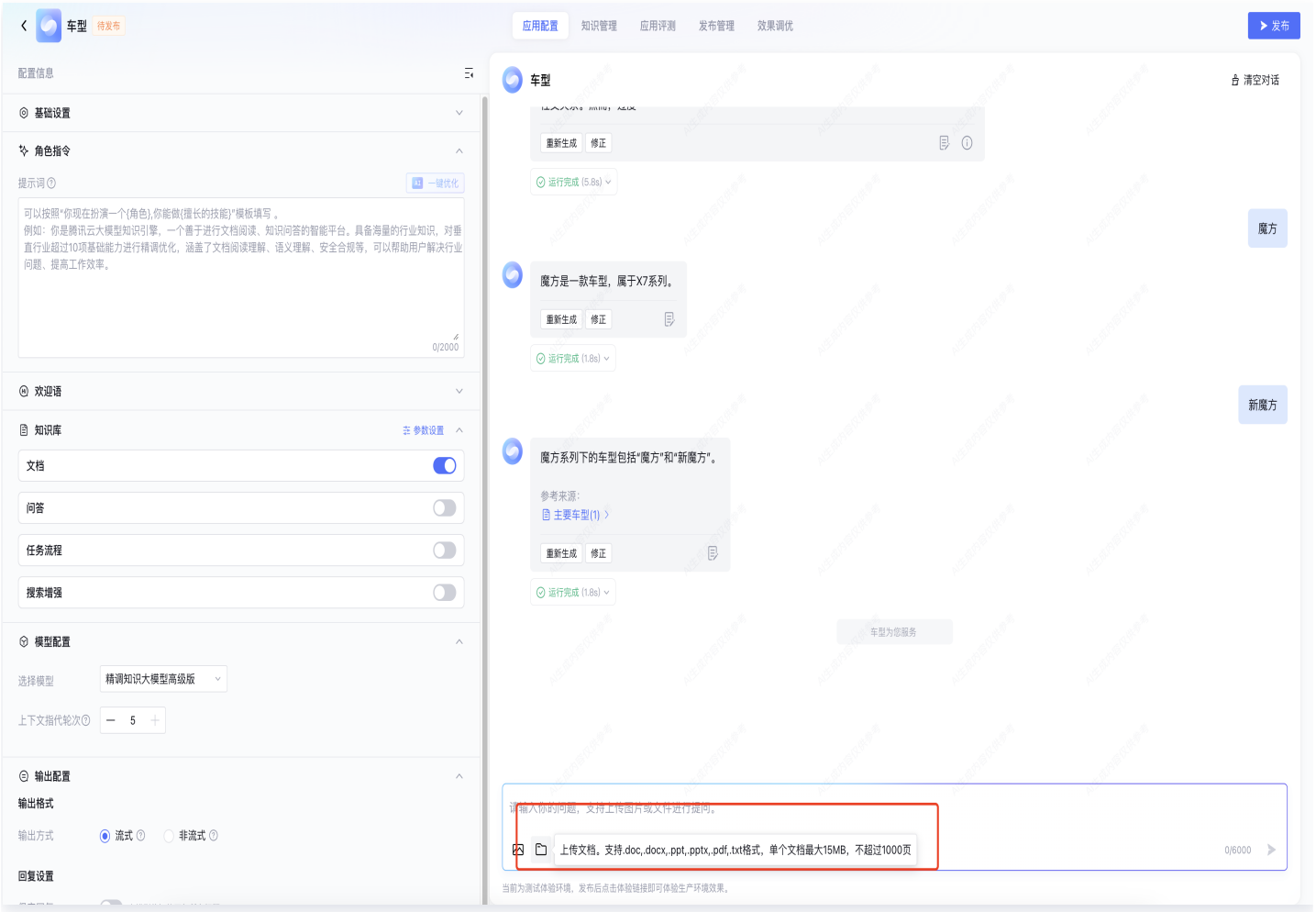
离线文档上传

最近更新时间：2025-05-20 00:00:52

名词解释

离线文档：离线文档主要是指智能体开发平台【知识管理】下面的【文档】【问答】。

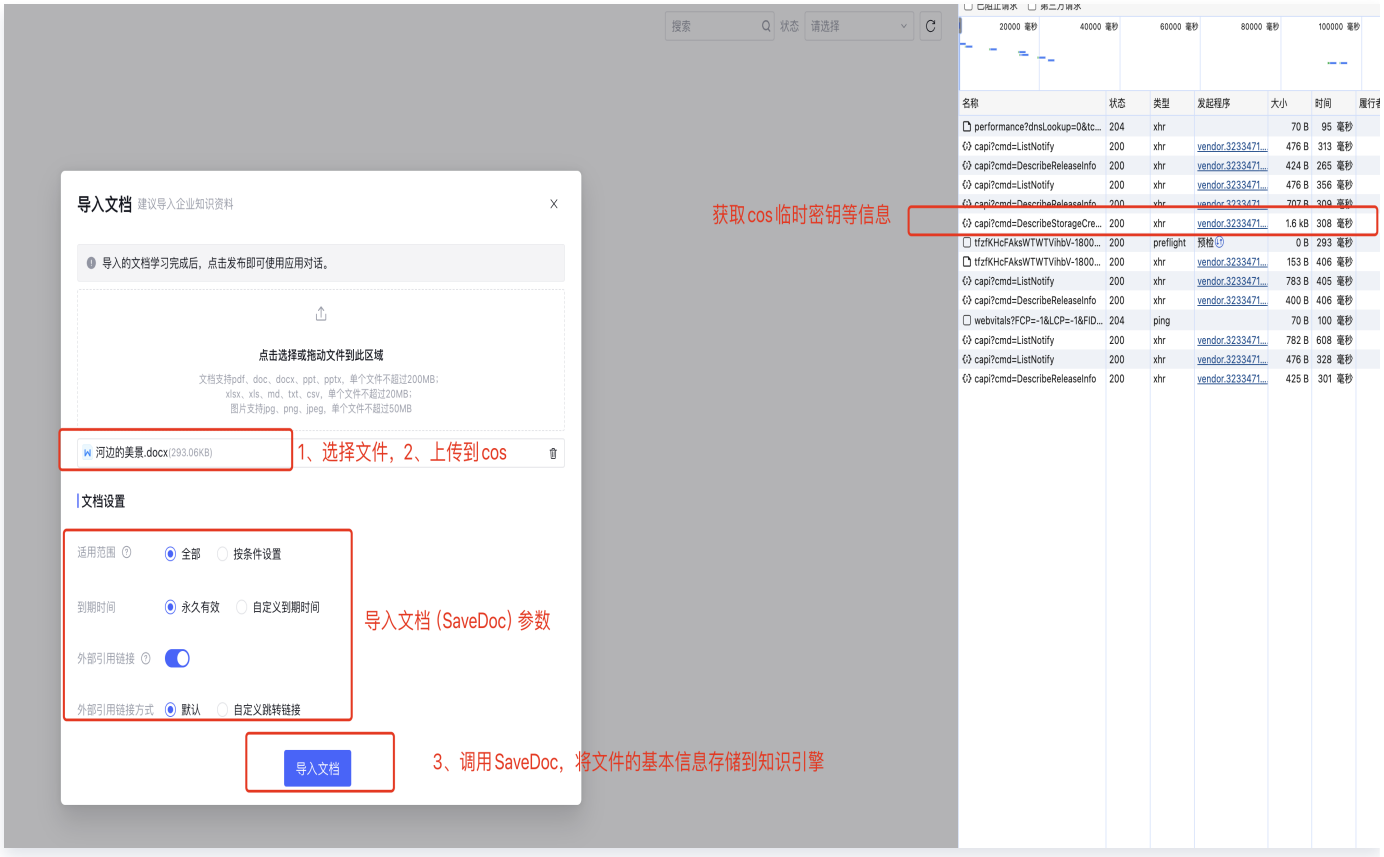
实时文档：实时文档主要是指对话界面中文档的上传。



步骤说明

离线上传文件需要三个步骤【其中前两步也适用于实时文档】：

1. 调用 DescribeStorageCredential 接口，获取临时密钥。
2. 通过临时密钥，通过 COS 的 SDK 上传到 COS。
3. 调用 SaveDoc 接口，将文件的基础信息存储到智能体开发平台中。如下图所示：



1. 获取临时密钥

参考：获取文件上传临时密钥 。

1.1 请求参数增加 FileType，FileType 为正常的文件名类型后缀，例如 xlsx、pdf、 docx、png 等。

- 填写 FileType 字段，获取的密钥只有上传权限，返回参数使用 UploadPath。
- 不填写 FileType 字段，获取的密钥只有下载权限。
- **TypeKey 区分是离线文件还是实时文件。**
- BotBizID 为必填选项。
- 每个文件上传都需要获取不同的临时密钥，并且临时密钥具有有效期。

> ⚠ **注意：**
> 请注意 TypeKey 取值不要混用，offline 为离线文档上传，realtime 用在实时文档上传；为空默认为 offline。

1.2 请求参数增加 IsPublic（区分公有场景还是私有场景）

示例：

公有场景： `/public/12332323/21321321321/image/1.png`

私有场景： `/corp/12332323/12332323/doc/1.pdf`

> **说明：**
> IsPublic 用于上传文件或图片时选择场景。离线文档上传时，IsPublic 为 false。实时文档上传时，如果是图片， IsPublic 为 true；如果是其他类型的文件，IsPublic 为 false。

接口返回的下面参数用于后续步骤中。

参数名	说明
Bucket	存储桶位置 [离线文档和实时文档此处也不同]
TmpSecretId	临时 secretID
TmpSecretKey	临时 secretKey
Token	用于上传的 token
UploadPath	后续步骤中用到，配合上传到 cos

接口返回示例：


```
{
    "code" : 0,
    "data" : {
        "Response" : {
            "Bucket" : "qidian-qbot-test-1251316161",
            "CorpUin" : "0",
            "Credentials" : {
                "TmpSecretId" : "********************************",
                "TmpSecretKey" : "********************************",
                "Token" : "********************************"
            },
            "ExpiredTime" : 1722234043,
            "FilePath" : "",
            "ImagePath" : "",
            "Region" : "ap-guangzhou",
            "RequestId" : "771ca2ee-03a7-487f-b77e-ccbb240f3cb8",
            "StartTime" : 1722233443,
```

```
            "Type" :  "cos",
            "UploadPath" :  "/corp/17468272416xxxxxxxxxx6.txt"
        }
    },
    "message" :  "OK",
    "reqId" :  "749c46ae-8070-457f-84be-5d0daa5cce05"
}
```


2. 调用腾讯云提供的 cos 存储接口，将文件存储到智能体开发平台 cos 中

参考：COS SDK 概览

> **说明：**
> 1. 需要用到步骤1中的 TmpSecretId,TmpSecretKey,Token,UploadPath 。
> 2. 文件上传到 cos[putObject] 不能直接从 API-Explorer 上拷贝代码，**下面两个在通过代码上传文件的时候，代码有差异，具体可以参考下面代码示例(demo)：**
>
>   API-Explorer 使用的是固定密钥。
>
>   通过代码上传使用的是临时密钥。
> 3. 需要注意，COS API 上传的时候对应的 key 为 获取临时密钥返回的 UploadPath 。

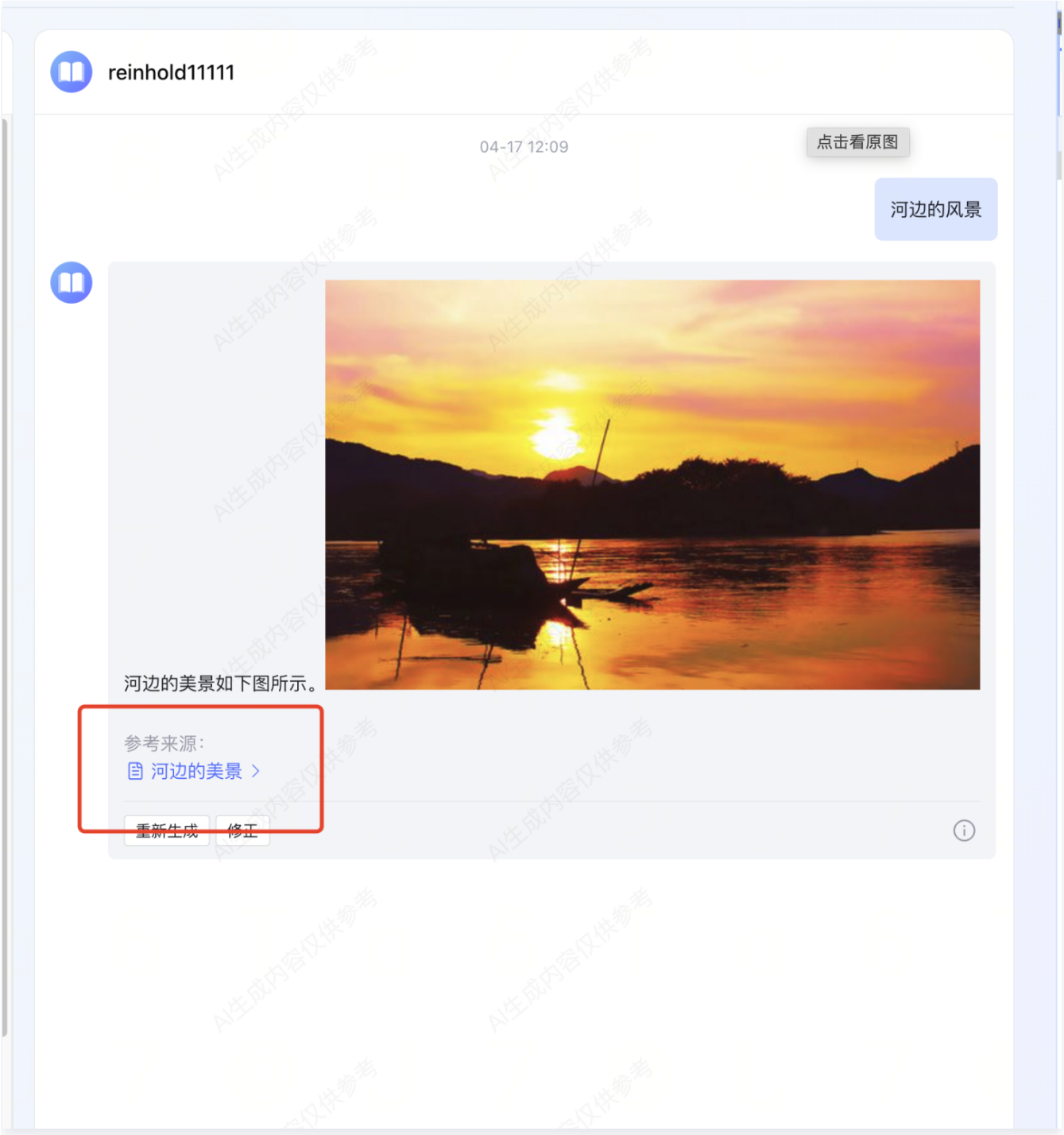
3. 调用 SaveDoc，保存元数据

参考：保存文档

几个重要字段说明， 其他字段请参考 api 文档。

字段	说明
IsRefer	为 true 的时候，会在用户端对话的过程中，命中了知识信息会给出相关的链接，效果如下图
CosUrl	步骤1中获取到的 uploadPath
CosHash	步骤2返回头中的 "X-Cos-Hash-Crc64ecma"
Etag	步骤2返回头中的 "Etag"
Opt	文档操作类型： 1：批量导入（批量导入问答对） 2：文档导入（正常导入单个文档） **当该值为1且导入的是 excel 文件时，会校验 excel 表头是否符合预期**

打开外部链接命中知识效果：



代码 demo

offline_upload_and_save_doc_20250427.zip

腾讯云智能体开发平台操作 cos 指南

最近更新时间：2025-05-27 17:47:32

核心差异说明

腾讯云智能体开发平台（TCADP）内置的 COS 服务与客户自购 COS 的主要区别如下：

1. 存储归属
   - TCADP 的数据存储在平台的 COS 中，无需单独购买存储服务，费用按字符计费。
   - 不支持将文件上传至客户自购的 COS 中。
2. 权限控制
   - 平台通过临时密钥实现细粒度权限管理，避免越权操作影响其他用户数据。

> **说明：**
> 使用腾讯云智能体开发平台 TCADP 产品，不需要购买 cos，不支持将文件上传到自己购买的 cos 中。

操作流程对比

操作步骤	腾讯云智能体开发平台内置COS	客户自购COS
1. 密钥获取	调用 DescribeStorageCredential 接口获取临时密钥	使用购买时获得的固定密钥（SecretId/Key）
2. 客户端初始化	通过临时密钥创建低权限 CosClient	通过固定密钥创建 CosClient
3. 数据操作	调用COS API（需临时密钥权限匹配）	直接调用COS API

> **注意：**
> 临时密钥的参数组合错误会导致操作失败（如403权限不足）。

临时密钥参数配置指南

以下为常见场景的参数组合示例，请根据实际需求填写（标灰字段为必填）：

场景	参数组合	备注
知识库上传文档（离线）	{"BotBizId":"182693390xxxx", "FileType":"pdf", "IsPublic":false}	参考 离线文档上传

对话中上传文档（实时）	{"BotBizId":"182693390xxxx", "FileType":"pdf", **"IsPublic":false**, "TypeKey":"realtime"}	参考 实时文档解析
对话中上传图片	{"BotBizId":"182693390xxxx", "FileType":"jpeg", **"IsPublic":true,** "TypeKey":"realtime"}	图片必须设为 IsPublic:true，参考 实时文档解析
文件下载	{"BotBizId":"182693390xxxx", **"IsPublic":false**}	无需指定文件类型

常见问题

? 为什么需要临时密钥?

- 平台通过动态权限隔离用户数据，保障安全性。

? 参数填错怎么办?

- 检查场景是否匹配（如上传图片需 IsPublic:true），或在对应操作页面通过抓包工具核对接口入参。

? 能否跨场景混用参数?

- 禁止混用。例如：离线文档上传不可使用 TypeKey:"realtime"。