大模型知识引擎

应用接口文档

腾讯云

【版权声明】

【商标声明】

【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

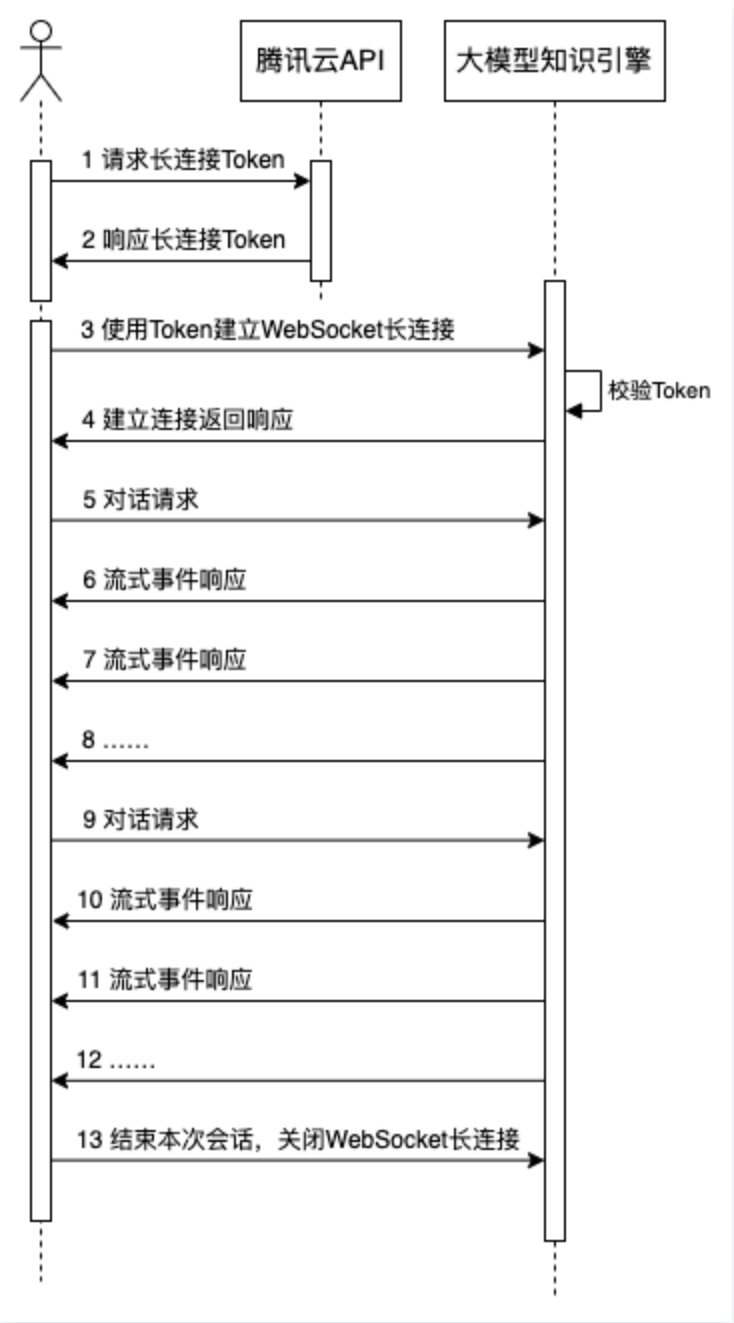
最近更新时间：2024-07-31 12:15:41

概述

在大模型知识引擎平台创建应用，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大模型知识引擎进行交互。
平台提供了两种常用的接入方式：WebSocket 与 HTTP SSE 的方式。

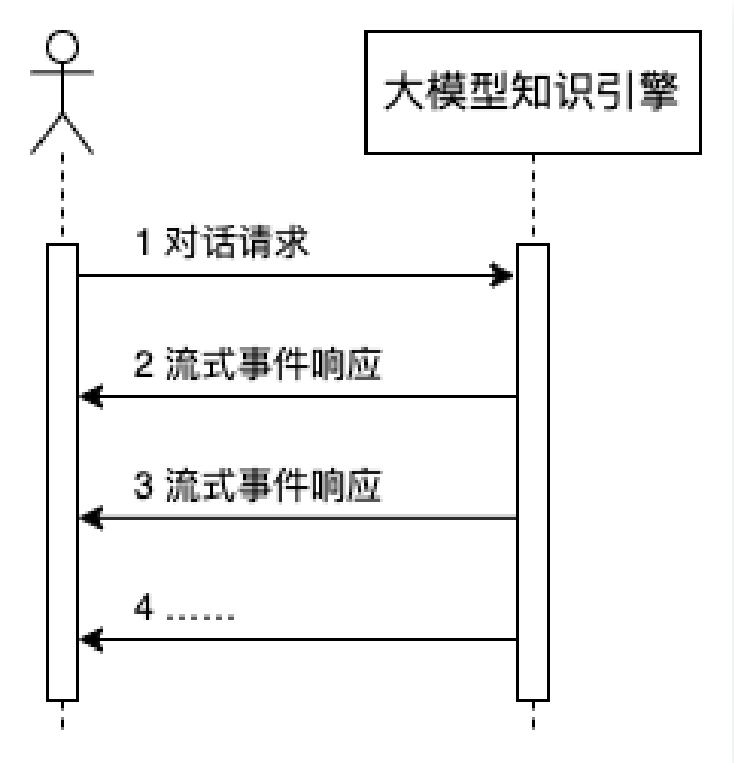
1. WebSocket 接入方式简介

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：



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。



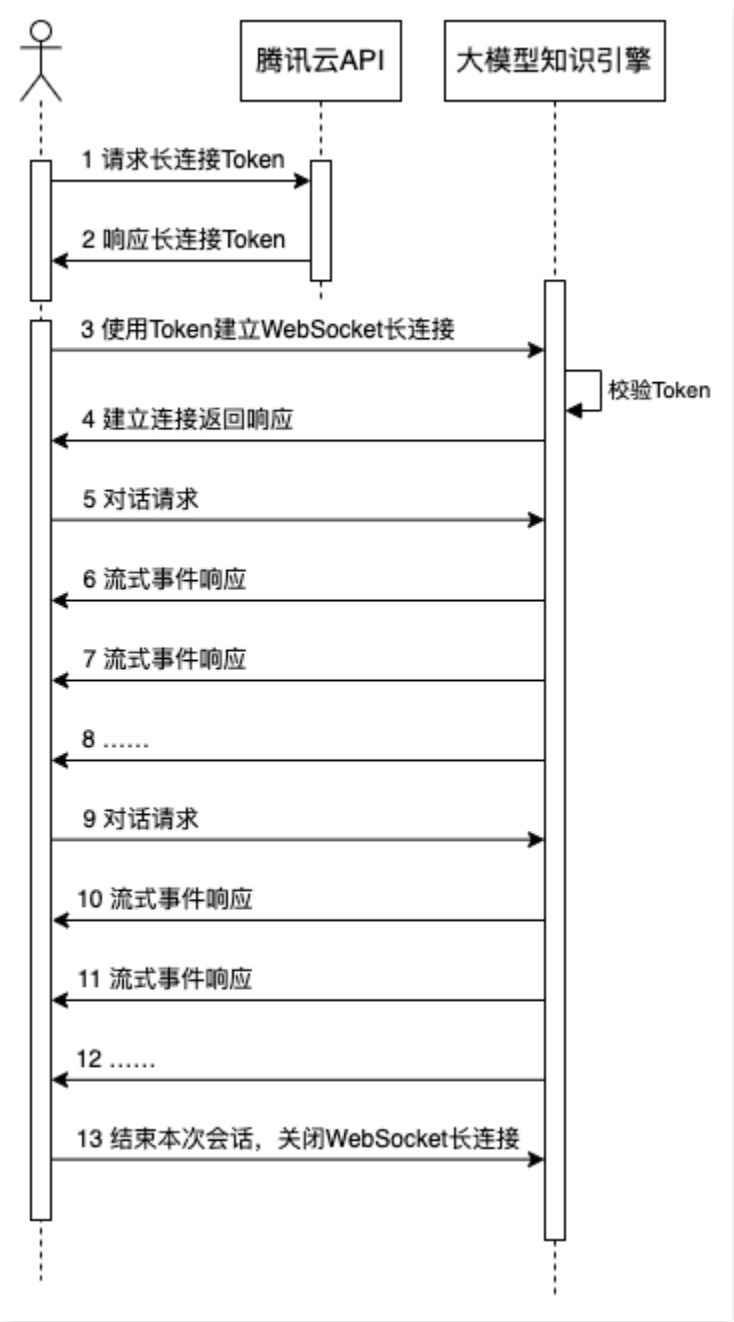
注意：

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：2024-09-18 18:24:21

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。
> - **如果需要携带标签信息，请在调用 GetWsToken 接口时传入。**

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}}]
```

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)	是	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）
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。可参考 实时文档解析 。
custom_variables	map[string]string	否	自定义参数的值。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。
system_role	string(2000)	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。

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	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
is_evil	bool	是否命中敏感内容
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
knowledge	Object 数组	命中的知识
option_cards	string 数组	选项卡，任务流程专有
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
task_flow	Object	任务流程调试信息

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

task_flow 任务流程调试信息的数据结构：

名称	类型	说明
task_flow_name	string	任务流程名称
task_flow_id	string	任务流程 ID
query_rewrite	string	问题改写结果
hit_intent	string	命中的意图
slot_info	map[string]Object	运行时收集的槽位信息

api_response	map[string]Object	API 节点的返回信息
type	int	任务流程回复类型 0：任务流程回复 1：任务流程静默回复 2：任务流程拉回话术 3：任务流程自定义回复

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	uint64	参考来源 ID，可调用 获取来源详情列表 接口查看参考来源详情。 **说明：** 该 id 字段对应 DescribeRefer 中的 ReferBizIds 字段
type	uint32	参考来源类型 1: 问答 2: 文档片段
url	string	参考来源链接（仅参考来源类型为文档片段时使用）
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

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

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
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
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

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

> ⚠ **注意：**
> 获取 Token 接口需要后端调用腾讯云 SDK。

JS 版本

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：2024-09-18 18:28:21

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

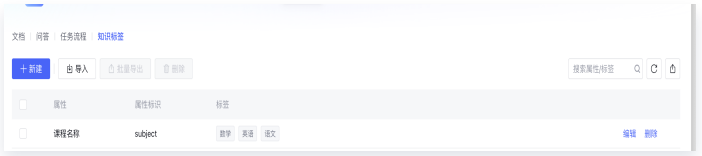
请求地址： `https://wss.lke.cloud.tencent.com/v1/qbot/chat/sse`
请求方式： POST

> **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)	是	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(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。可参考 实时文档解析 。
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	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）
visitor_labels	Object 数组	否	知识标签（用于知识库中知识的检索过滤）；知识标签结构为: {"name": "subject", "values": ["语文", "数学"]} 知识标签里面定义了相关的属性，属性标识，标签。  文档适用范围选择了“语文”，“数学”标签。  **注意：** visitor_labels 中的 name 字段对应上图中的属性标识。
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。 **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	自定义参数的值。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。

system_role	string(2000)	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。

visitor_labels 知识标签列表的数据结构：

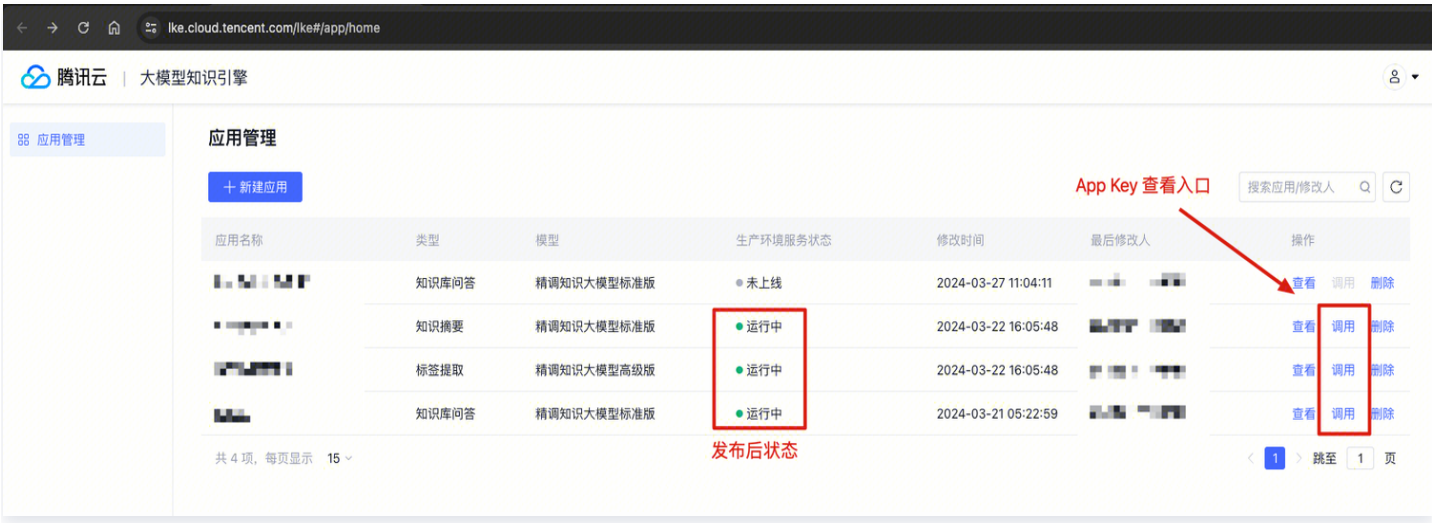
名称	类型	说明
name	string	知识标签名
values	string 数组	知识标签值

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。

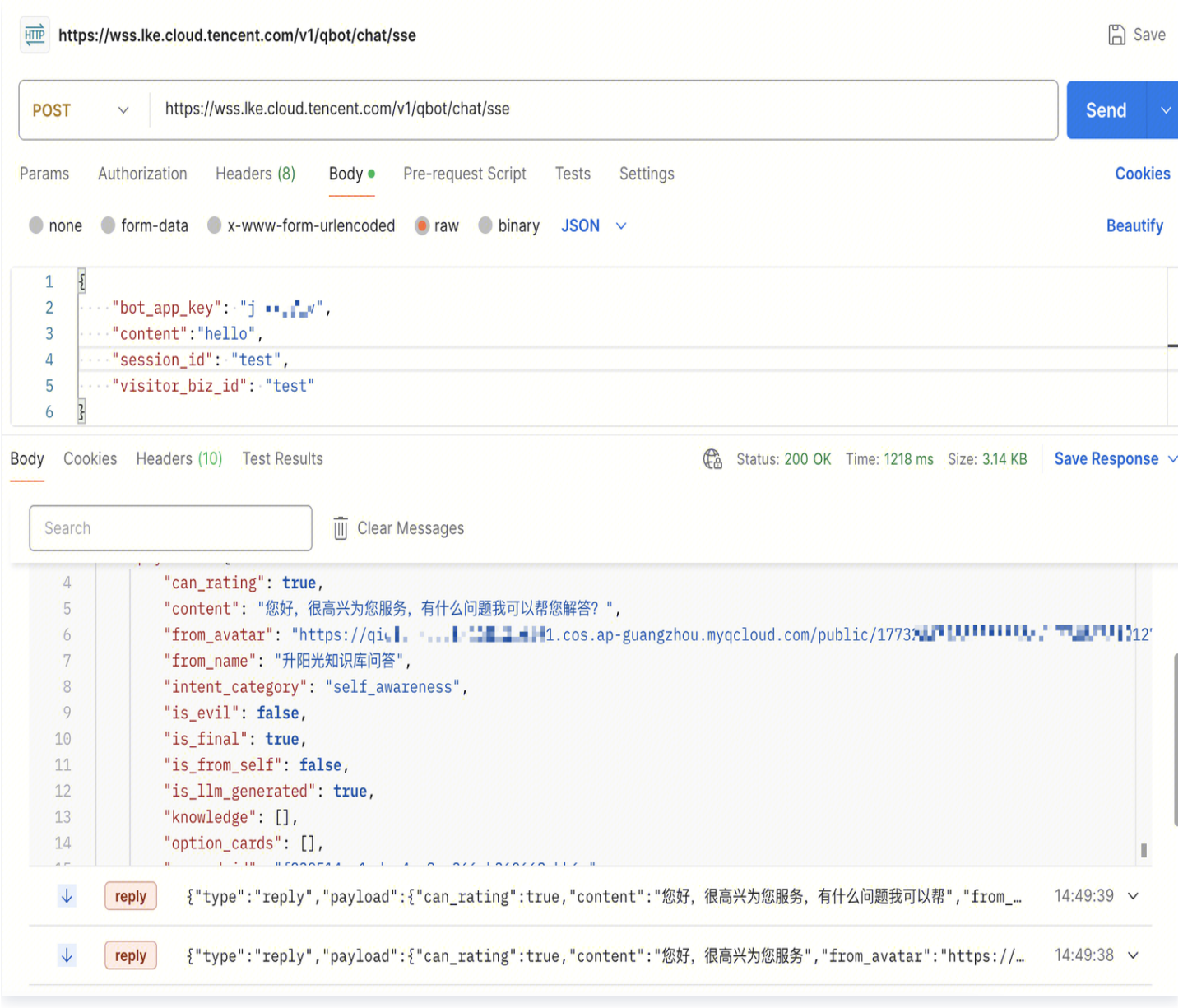


在**调用信息**窗口可以看到 AppKey，单击**复制**即可。

1.3 curl 调用示例

```
curl -XPOST -vvv --no-buffer --location
'https://wss.lke.cloud.tencent.com/v1/qbot/chat/sse' \
--header 'Content-Type: application/json' \
--data '{
    "content": "消息内容",
    "bot_app_key": "<your appkey>",
    "visitor_biz_id": "<your visitor id>",
    "session_id": "<your session_id>",
    "visitor_labels": []
}'
```

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信息，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	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）
is_evil	bool	是否被敏感词打击
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
knowledge	Object 数组	命中的知识
option_cards	string 数组	选项卡，任务流程专有
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
task_flow	Object	任务流程调试信息

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

task_flow 任务流调试信息的数据结构：

名称	类型	说明
task_flow_name	string	任务流程名称
task_flow_id	string	任务流程 ID
query_rewrite	string	问题改写结果
hit_intent	string	命中的意图
slot_info	map[string]Object	运行时收集的槽位信息
api_response	map[string]Object	API 节点的返回信息
type	int	任务流程回复类型 0：任务流程回复 1：任务流程静默回复 2：任务流程拉回话术

		3：任务流程自定义回复

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	uint64	参考来源 ID，可调用 获取来源详情列表 接口查看参考来源详情 **说明：** 该 id 字段对应 DescribeRefer 中的 ReferBizIds 字段
type	uint32	参考来源类型 1: 问答 2: 文档片段
url	string	参考来源链接（仅参考来源类型为文档片段时使用）
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

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

2.5 返回示例

```
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":"我是大模型知识
引擎，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。

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
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
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

4. 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。
JS 版本

4.1 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。

实时文档解析

最近更新时间：2024-09-23 14:08:11

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 PutObject 接口 上传到 cos。
3. 调用实时文档解析接口，获取 doc_id。

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

1. 请求参数

请放到 HTTP Body 中, 以 JSON 的形式发送，具体如下：

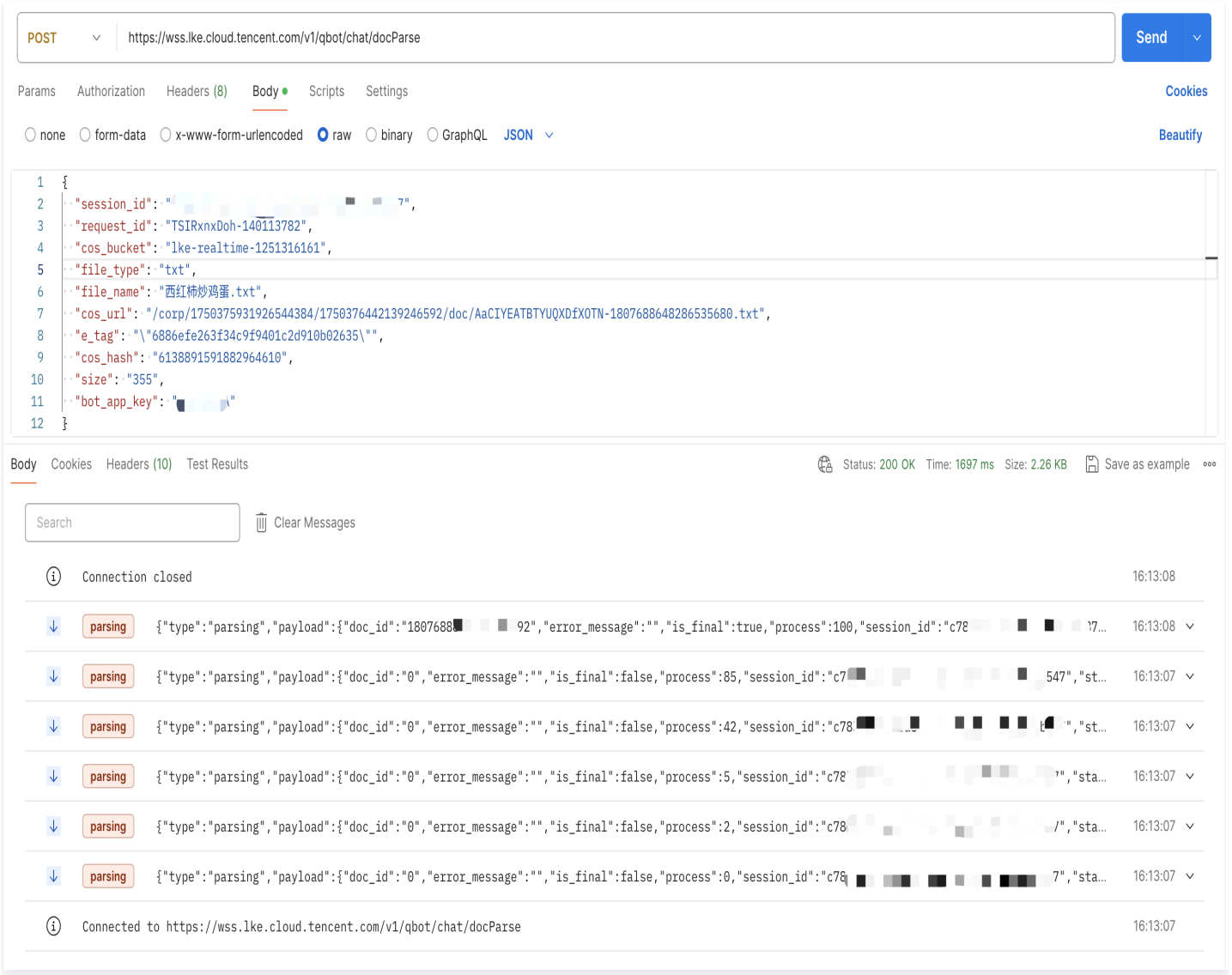
名称	类型	是否必填	说明
session_id	string(64)	是	会话 ID，用于标识一个会话（外部系统提供，建议不同的用户端会话传入**不同的**session_id，否则同一个应用下的不同用户的消息记录会串掉） **参数长度：2-64个字符** **校验规则： ^[a-zA-Z0-9_-]{2,64}$ ，一般可以用 uuid 来生成该值** **uuid 示例：1b9c0b03-dc83-47ac-8394-b366e3ea67ef** **注意：** 文档解析的 session_id 要与会话的保持一致。如果一次多轮会话涉及多次上传文档，该 session_id 也要保持一致，会话时会校验。
bot_app_key	string	是	应用密钥（运营提供）
request_id	string	是	请求的唯一 ID，建议使用 UUID 保证唯一性。
cos_bucket	string	是	cos 桶，使用 lke-realtime-1251316161
file_type	string	是	文件类型(md|txt|docx|pdf|xlsx)

			示例值：md|txt|docx|pdf|xlsx
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/23432432/233432/doc/zeSOHIBsdawkmlMMXOp-1796022574489010176.docx
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
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 示例值："58e88ad7665f11c4f66eba0eada383a5"
size	string	是	uint64 类型转为 string，上传文档的大小

1.1 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": "txt",
  "file_name": "西红柿炒鸡蛋.txt",
  "cos_url":
"/corp/17503759319265443 84/17503764421392465 92/doc/AaCIYEATBTYUQXDfXOTN-
1807688648286535680.txt",
  "e_tag": "\"6886efe263f34c9f9401c2d910b02635\"",
  "cos_hash": "6138891591882964610",
  "size": "355",
  "bot_app_key": "<your appkey>"
}'
```

1.2 postman 调用示例



2. 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

2.1 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. 实时文档解析 demo：

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。

离线文档上传

最近更新时间：2024-07-29 16:26:31

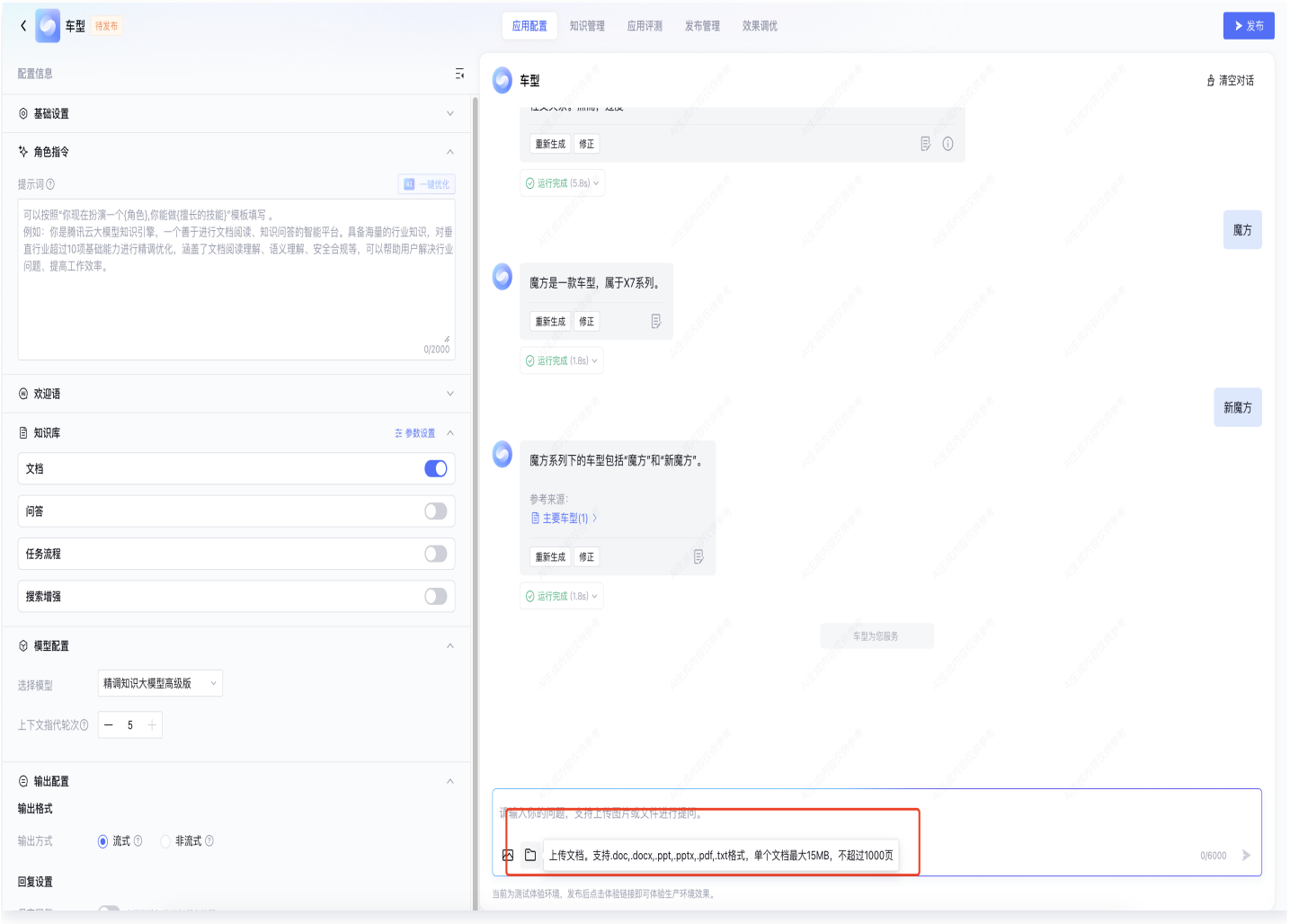
说明

1. 客户已经购买过腾讯云的 cos，用在客户自己的其他场景，由客户自行管理该 cos。
2. 客户购买【试用】知识引擎产品，需要做文件的上传下载访问等,这个时候文件所归属的 cos 是知识引擎团队内部申请的 cos, 这个 cos 是由知识引擎产品团队维护的，包括文件存储位置，文件访问权限等；暂不支持使用知识引擎产品的时候将文件保存到客户自行购买的 cos 地址上，cos 桶地址等相关信息在调用临时密钥接口【DescribeStorageCredential】，会给出来。
3. 使用接口调用 SaveDoc 接口的时候，有三个步骤：获取临时密钥，将文件上传到知识引擎的 cos ,调用 SaveDoc 保存；不能一步到位。

名词解释

离线文档：离线文档主要是指知识引擎【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。
离线文档

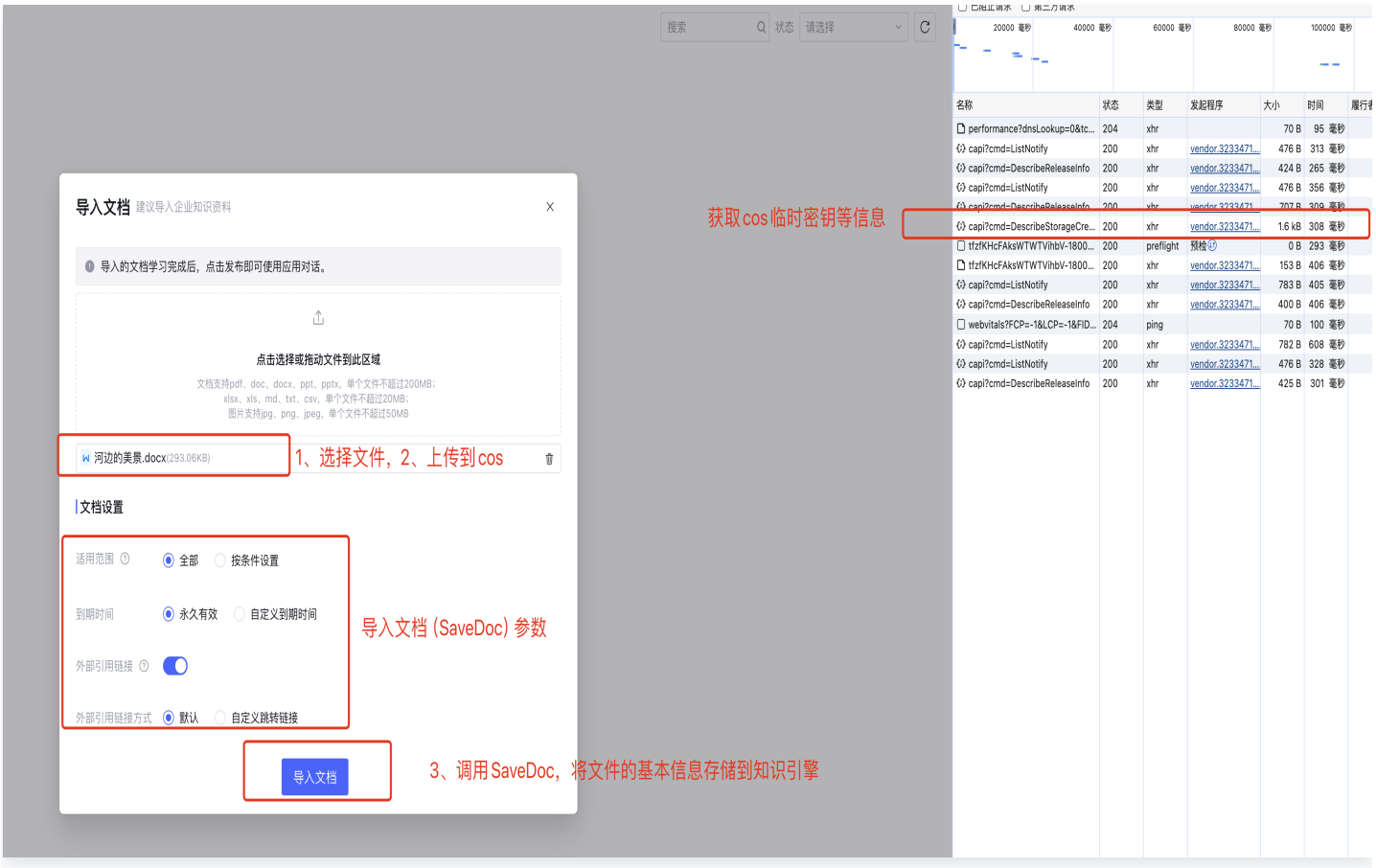
实时文档



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. 获取临时密钥。
2. 将文件上传到知识引擎给定的 cos 中。
3. 调用 SaveDoc 接口，将文件的基础信息存储到知识引擎中。

如下图：



三个步骤的参考：

1、 获取临时密钥

参考： **获取文件上传临时密钥** 。

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。
   1.1 填写 FileType 字段，获取的密钥只有上传权限，返回参数使用 UploadPath。
   1.2 不填写 FileType 字段，获取的密钥只有下载权限。
   1.3 **TypeKey 区分是离线文件还是实时文件**。
   1.4 BotBizID 为必填选项。
   1.5 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。

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：**
> 图片一般为 IsPublic: true; 从体验端，用户端上传图片的时候 IsPublic: true。

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" : "AKID92kj37NMnrgxxxxxxxx9qqk5zVp7",
                "TmpSecretKey" : "FKF/LyKCAXe2rxxxxxxxxxfVEipKg=",
                "Token" : "GLiqrSyvV6K4gGo1FiXAlRxxxxxxxxxxxxNJeg"
            },
            "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，将文件存储到知识引擎 cos 中

参考：PUT Object

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。

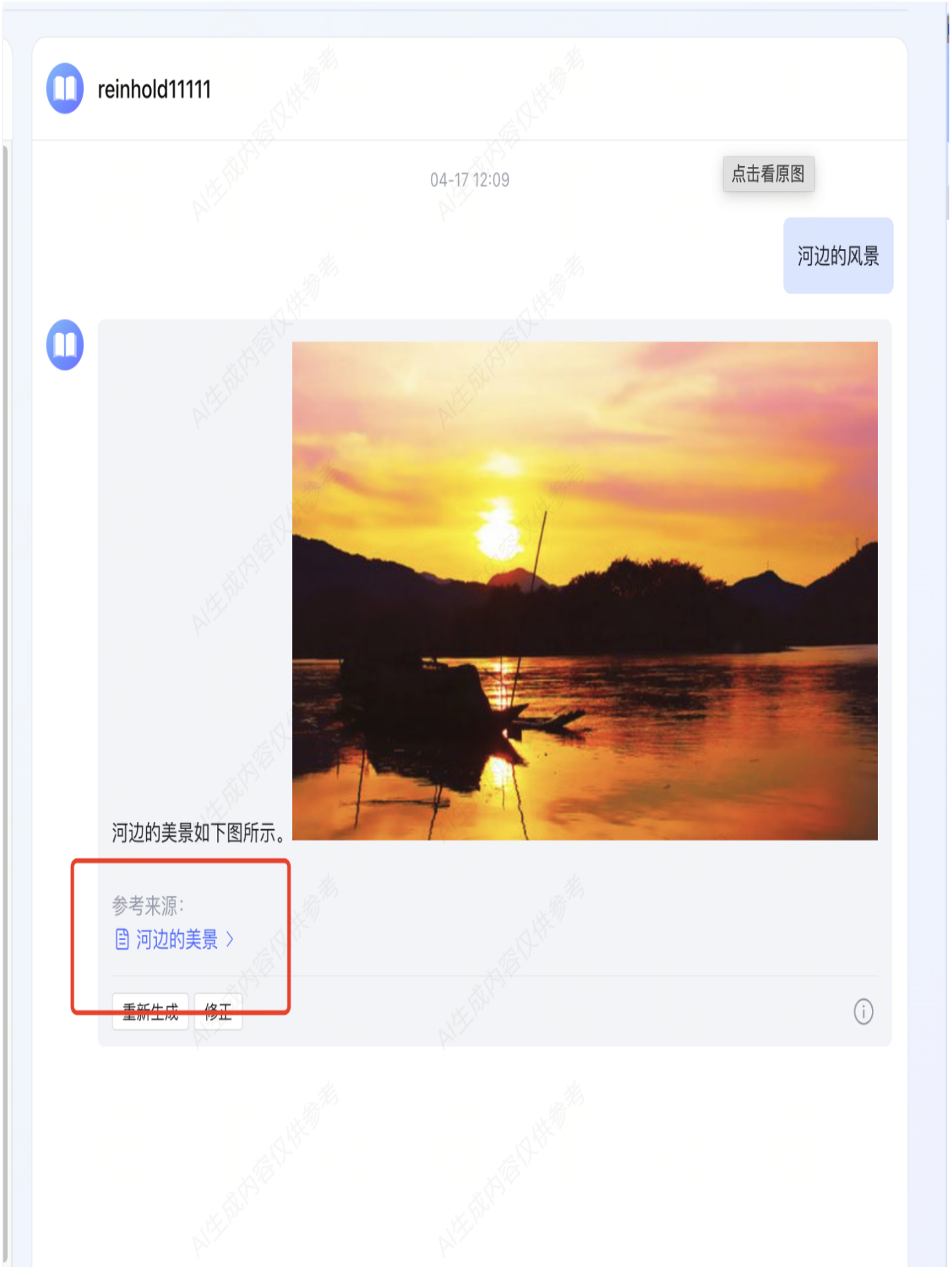
3、调用 SaveDoc，保存元数据

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：


reinhold11111
04-17 12:09
点击看原图
河边的风景
河边的美景如下图所示。
参考来源:
河边的美景 >
重新生成
修正

离线文档上传代码 demo:

offline_upload_and_save_doc_20240729.zip