Conversational AI Agent v0.3 CN接入指导书

Conversational AI Agent是声网提供的新一代AI Agent解决方案，帮助客户以简化与智能体进行实时交互。

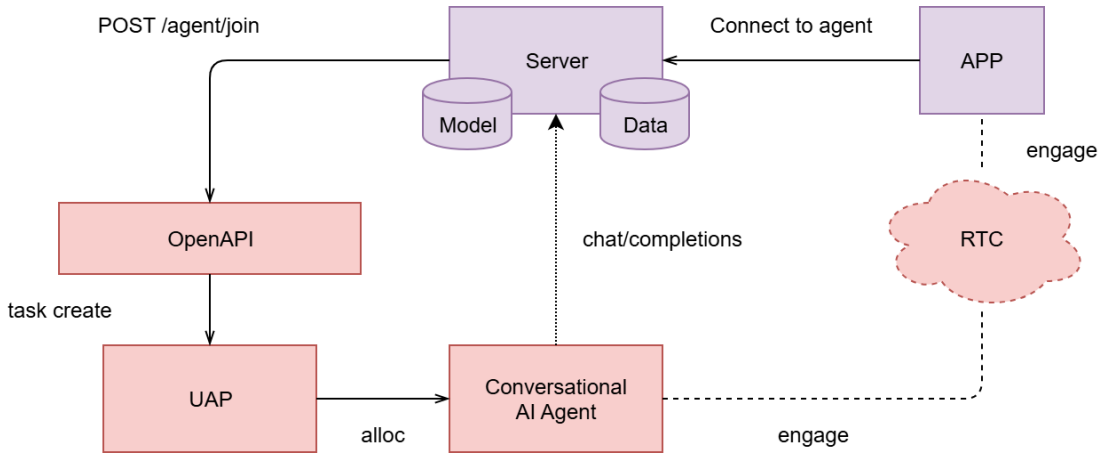
当前本服务只是Private Beta阶段，最终协议和API可能存在微调尽情谅解；

目前需要手工开通服务能力，具体开通请联系Mao Yujie

与上一个版本 Conversational AI Agent v0.2 CN接入指导书的差异使用红色标识

方案简介

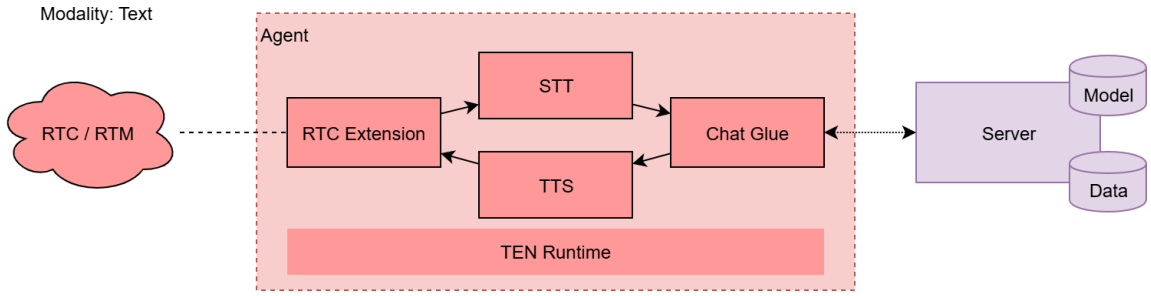
Conversational AI Agent整体架构如下：



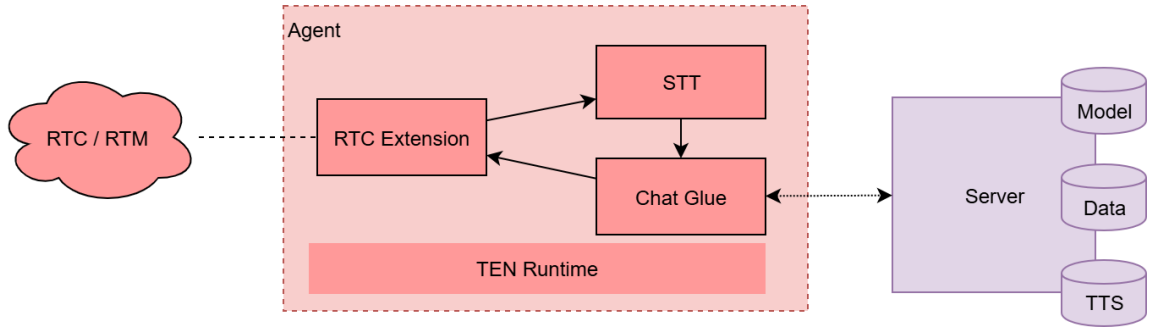
通过声网提供的RESTful API，客户可以启动一个AI Agent加入指定的RTC频道与频道中的用户进行1对1 RTC交流

注：目前Agent不支持多人沟通

Conversational AI Agent内部包含STT和TTS模块，会通过OpenAI兼容协议向客户提供的URL发起请求，并流式的获取其响应结果，具体内部实现如下：



对于STT和TTS，目前由声网提供，如果客户需要使用自己实现或集成的TTS，则可以选择使用Audio模态服务，具体实现如下：



在这个场景下，输入为文本，输出为音频。Chat Completion直接支持使用OpenAI在audio-preview中的音频协议，接受base64编码后的pcm片段

接入指南

Agent启动

Method：POST

URL：https://api.agora.io/cn/api/conversational-ai-agent/v1/projects/{appid}/join

启动参数：

- Name：Agent唯一标识，相同标识不可重复创建
- Properties：Agent具体参数，完整请求格式如下：

```
{
    "channel": "channel_name",
    "token": "token",
    "agent_rtc_uid": 1234,
    "remote_rtc_uid": 123,
    "audio_scenario": 9,
    "input_modalities": [
        "text"
    ],
    "output_modalities": [
        "text"
    ],
    "idle_timeout": 120,
    "custom_llm": {
        "url": "https://api.minimax.chat/v1/text/chatcompletion_v2",
        "token": "xxx",
        "prompt": "You are a helpful chatbot.",
        "model": "abab6.5s-chat",
        "greeting": "",
        "failure_message": "",
        "extra_metadata": {
            "userName": ""
        },
        "metadata_enabled": true
    },
    "tts": {
        "voice_id": "male-qn-qingse"
    },
    "asr": {
        "language": "zh-CN"
    },
    "vad": {
        "silence_duration_ms": 1000
    }
}
```

具体参数描述如下：

参数	类型	描述	默认参数	备注
token	string	rtc 频道 token	""	
channel	string	rtc 频道名	必填	
agent_rtc_uid	int32	rtc 用户id	必填	0：表示随机分配，但是Token也需要相应修改
remote_rtc_uid	int32	rtc订阅用户id	必填	0：表示订阅所有人，针对1v1场景需要明确指定
rtc_codec	**int32**	**rtc发流编码格式**	**-1**	**0：PCMU** **8：PCMA** **9：G722** **120：OPUS** **122：OPUSFB**
audio_scenario	**int32**	**音频模式**	**0**	**0：默认配置** **9：AI QOS（需要搭配AI QOS端侧版本）**
input_modalities	array[string]	custom llm的输入模态	["text"]	**支持["text"]，["text", "image"]**
output_modalities	array[string]	custom llm的输出模态	["text"]	支持["audio"],["text"],["text", "audio"]
idle_timeout	int	agent最大空闲时间（秒）	30	如果填写为0，则直到手工退出，agent才会停止
asr.language	string	语言	“zh-CN”	zh-CN（支持中英文混合），en-US
tts.voice_id	string	音色ID	"female-shaonv"	具体支持的音色请参考附录
custom_llm.url	string	custom llm回调地址	必填	
custom_llm.token	string	custom llm校验token	""	生产环境务必启用token
custom_llm.prompt	string	custom llm可能支持的Prompt	“”	会作为System Message传入
custom_llm.model	string	custom llm可能需要的Model ID	""	
custom_llm.max_history	**int32**	**custom llm中缓存的短期记忆条目数**	**10**	**0：不缓存任何短期记忆** **User和Assistant会单独记录条目**
custom_llm.greeting	string	agent问候语	""	如果填写，则频道内第一个rtc用户加入时会自动问候
custom_llm.failure_message	string	agent失败提示	""	如果填写，则在custom llm调用错误时会通过TTS返回
custom_llm.extra_metadata	dict	agent携带上下文信息	{}	如果填写，每次custom llm请求时会通过`metadata`参数传递给Server *第三方大模型服务不支持这个参数，仅在自建场景适用*
custom_llm.metadata_enabled	bool	是否需要携带上下文	false	
vad.interrupt_threshold	float	打断灵敏度	0.8	数值越大，灵敏度越高 0.0 - 1.0
vad.prefix_padding_ms	int	音频识别开始的Padding时长（毫秒）	300	
vad.silence_duration_ms	int	音频静默时长（毫秒）	500	
vad.threshold	float	vad识别灵敏度	0.3	数值越大，灵敏度越高 0.0 - 1.0

返回值：正常

StatusCode：200

属性	类型	描述

agent_id	string	Agent ID，实例唯一标识，用于后续删除
create_ts	int	创建时间戳
state	string	当前状态

返回值：错误

StatusCode：非200

属性	类型	描述
detail	string	具体错误描述
reason	string	错误类别

错误码参见后续章节

Agent停止

Method: POST

URL:https://api.agora.io/cn/api/conversational-ai-agent/v1/projects/{appid}/agents/{agentId}/leave

其中agentId填写join时返回的agent_id

Agent 变更

允许运行时对Agent的参数进行调整

Method: POST

URL:https://api.agora.io/cn/api/conversational-ai-agent/v1/projects/{appid}/agents/{agentId}/update

其中agentId填写join时返回的agent_id，允许变更的参数包括：

参数	类型	描述
token	string	rtc 频道 token
tts.voice_id	string	音色ID
custom_llm.url	string	custom llm回调地址
custom_llm.token	string	custom llm校验token
custom_llm.prompt	string	custom llm可能支持的Prompt
custom_llm.model	string	custom llm可能需要的Model ID
custom_llm.greeting	string	agent问候语
custom_llm.failure_message	string	agent失败提示
custom_llm.extra_metadata	dict	agent携带上下文信息
custom_llm.metadata_enabled	bool	是否需要携带上下文
vad.interrupt_theshold	float	打断灵敏度
vad.prefix_padding_ms	int	音频识别开始的Padding时长（毫秒）
vad.silence_duration_ms	int	音频静默时长（毫秒）
vad.threshold	float	vad识别灵敏度

Agent 状态查询

获取运行中Agent的实时状态

Method: GET

URL:https://api.agora.io/cn/api/conversational-ai-agent/v1/projects/{appid}/agents/{agentId}


```
{
    "message": "",
    "requestResponse": {
        "agent_name": "conversational-ai-agents",
        "name": "unique_name",
        "properties": {
            "agent_rtc_uid": 1234,
            "asr": {
                "language": "zh-CN"
            },
            "audio_scenario": 9,
            "channel": "channel_name",
            "custom_llm": {
                "model": "abab6.5s-chat",
                "greeting": "",
                "url": "https://api.minimax.chat/v1/text
/chatcompletion_v2",
                "token": "xxx"
            },
            "tts": {
                "voice_id": "female-shaonv"
            }
        }
    },
    "statusCode": "ok",
    "taskId": "1NT29X11GQSINF1ECVGZNU80BEIN56XF"
}
```


Agent 查询列表

获取满足条件的Agent列表

Method: GET

URL:https://api.agora.io/cn/api/conversational-ai-agent/v1/projects/{appid}/agents

```
{
    "data": {
        "count": 1,
        "list": [
            {
                "startTs": 1735035893,
                "stateName": "STATE_RUNNING",
                "taskId": "1NT29X11GQSINF1ECVGZNU80BEIN56XF"
            }
        ]
    },
    "meta": {
        "cursor": "",
        "total": 1
    },
    "status": "ok"
}
```


Glue Sample

目前请求需要客户侧自行准备Glue URL和Token

现有第三方大模型

当前Glue的协议兼容OpenAI，因此所有兼容OpenAI的大模型提供商可以直接被接入，以Minimax为例，可以通过如下方式访问：

```
curl --location 'https://api.agora.io/cn/api/conversational-ai-agent/v1
/projects/{appid}/join' \
--header 'Authorization: Basic ${key}'  \
--header 'Content-Type: application/json' \
--data '{"name":"unique_name","properties":{"channel":"test_channel",
"agent_rtc_uid":1234, "custom_llm": {"url":"https://api.minimax.chat/v1
/text/chatcompletion_v2", "token":"xxxxx", "model":"abab6.5s-chat",
"greeting":""}}}' -i
```

自行实现Glue协议

整体协议与OpenAI一致，下面是Python的样例代码：

```
import os
import json
import traceback
import logging
import asyncio

from typing import List, Union, Dict, Optional
```

```
from pydantic import BaseModel, HttpUrl

from fastapi.security import HTTPBearer, HTTPAuthorizationCredentials
from fastapi.responses import StreamingResponse
from fastapi import Depends, FastAPI, HTTPException, Request

logger = logging.getLogger(__name__)

app = FastAPI(title="Chat Completion API",
              description="API for streaming chat completions with support
for text, image, and audio content",
              version="1.0.0")

class TextContent(BaseModel):
    type: str = "text"
    text: str

class ImageContent(BaseModel):
    type: str = "image"
    image_url: HttpUrl

class AudioContent(BaseModel):
    type: str = "input_audio"
    input_audio: Dict[str, str]

class ToolFunction(BaseModel):
    name: str
    description: Optional[str]
    parameters: Optional[Dict]
    strict: bool = False

class Tool(BaseModel):
    type: str = "function"
    function: ToolFunction

class ToolChoice(BaseModel):
    type: str = "function"
    function: Optional[Dict]

class ResponseFormat(BaseModel):
    type: str = "json_schema"
    json_schema: Optional[Dict[str, str]]

class SystemMessage(BaseModel):
    role: str = "system"
    content: Union[str, List[str]]

class UserMessage(BaseModel):
    role: str = "user"
    content: Union[str, List[Union[TextContent, ImageContent,
AudioContent]]]

class AssistantMessage(BaseModel):
```

```
    role: str = "assistant"
    content: Union[str, List[TextContent]] = None
    audio: Optional[Dict[str, str]] = None
    tool_calls: Optional[List[Dict]] = None

class ToolMessage(BaseModel):
    role: str = "tool"
    content: Union[str, List[str]]
    tool_call_id: str

class ChatCompletionRequest(BaseModel):
    context: Optional[Dict] = None
    model: Optional[str] = None
    messages: List[Union[SystemMessage, UserMessage, AssistantMessage,
ToolMessage]]
    response_format: Optional[ResponseFormat] = None
    modalities: List[str] = ["text"]
    audio: Optional[Dict[str, str]] = None
    tools: Optional[List[Tool]] = None
    tool_choice: Optional[Union[str, ToolChoice]] = "auto"
    parallel_tool_calls: bool = True
    stream: bool = True
    stream_options: Optional[Dict] = None

security = HTTPBearer()

def verify_token(credentials: HTTPAuthorizationCredentials = Depends
(security)):
    token = credentials.credentials
    if token != os.getenv("API_TOKEN"):
        logger.warning("Invalid or missing token")
        raise HTTPException(status_code=403, detail="Invalid or missing
token")

@app.post("/chat/completions", dependencies=[Depends(verify_token)])
async def create_chat_completion(request: ChatCompletionRequest, req:
Request):
    try:
        logger.debug(f"Received request: {request.json()}")
        #TODO Your logic here
                yield "data: [DONE]\n\n"
    except asyncio.CancelledError:
        logger.info("Request was cancelled")
        raise HTTPException(status_code=499, detail="Request was
cancelled")
    except Exception as e:
        traceback_str = ''.join(traceback.format_tb(e.__traceback__))
        error_message = f"{str(e)}\n{traceback_str}"
        logger.error(error_message)
        raise HTTPException(status_code=500, detail=error_message)

if __name__ == "__main__":
    import uvicorn
```

```
from fastapi.security import HTTPBearer, HTTPAuthorizationCredentials
from fastapi import Depends

uvicorn.run(app, host="0.0.0.0", port=8000)
```

视觉模态理解

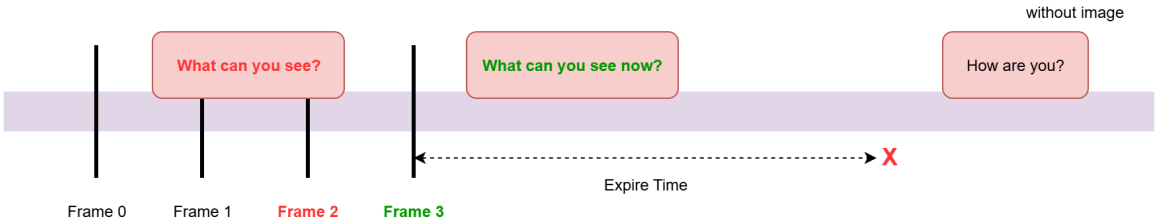
在输入模态中增加image，则会要求大模型支持视觉模态，具体协议参考OpenAI，具体message如下：

```
{
    "role": "user",
    "content": [{
        "type": "image_url",
        "image_url": {
            "url": "jpeg_base64encoded"
        }
    }]
}
```

Agent会每秒记录一个RTC视频帧，当需要询问大模型时携带该视频帧一同发给大模型，需要确保Custom LLM大模型支持视觉模态，可能包括如下场景：

- 持续发送视频，则每个问答都会携带视频进行问答
- 当需要视觉配合回答问题，则在回答过程发送对应的视频帧
    - 可以选择将静态图片通过RTC发送，并在不需要视觉时，不再发送

Agent会在停止接受视频帧一定时间后忘记之前的缓存，具体逻辑如下：



端侧字幕获取

目前整套方案支持通过Data Stream获取字幕信息，不同的SDK的获取方式参考文档：

- iOS:https://doc.shengwang.cn/api-ref/rtc/ios/API/toc_datastream#callback_irtcengineeventhandler_onstreammessage
- Android:https://doc.shengwang.cn/api-ref/rtc/android/API/toc_datastream#onStreamMessage

字幕会在Agent和人说话过程返回，主要包括如下标识：

- is_final：文字是否还会变化，用于人的ASR结果（**永远为True**）
- stream_id：字幕对应的用户，Agent标识为0，非零则使用对应用户的int uid
- message_id：用于唯一标识一个消息
- data_type：默认为“transcribe”
- text_ts： 字幕生成的毫秒时间
- text：具体的字幕内容

发送时，会自动将超过1k数据包进行如下编码：{message_id}|{id}|{total}|{chunked_data}，其中data为base64编码后的json内容

原始内容

```
{
        "is_final": true,
        "stream_id": 0,
        "message_id": "abcdefgh",
        "data_type": "transcribe",
        "text_ts": 1111111111,
        "text": "Once upon a time, in a small village nestled between rolling hills, there was a mysterious forest known as the Whispering Woods. The villagers often spoke of the woods in hushed tones, for it was said that the trees could speak, sharing secrets of the past and visions of the future. However, few dared to enter, fearing the unknown.One day, a curious girl named Elara decided to explore the woods. She had always been fascinated by the tales told by the village elders. With a heart full of courage and a sense of adventure, she set off at dawn, leaving behind the familiar sounds of her village.As Elara stepped into the Whispering Woods, she was greeted by a gentle breeze that rustled the leaves above. It felt as if the trees were welcoming her. The deeper she ventured, the more alive the forest seemed. She could hear faint whispers, like a soft melody carried by the wind.“Who’s there?” Elara called out, her voice echoing in the stillness. To her surprise, the whispers grew louder, forming words that danced around her.“Seek the heart of the woods, and you shall find what you seek,” the trees murmured in unison.Intrigued, Elara followed the sound, her heart racing with anticipation. After walking for what felt like hours, she came upon a clearing bathed in sunlight. In the center stood an ancient tree, its trunk wide and gnarled, with branches that reached towards the sky. It was the oldest tree in the forest, known as the Heartwood.Approaching the Heartwood, Elara placed her hand on its rough bark. Suddenly, she felt a warmth spreading through her fingers, and the whispers transformed into a clear voice. “You have shown great courage, young one. What is it that you seek?”Elara took a deep breath. “I seek knowledge and understanding. I want to know my purpose in this world.” The Heartwood paused, and Elara felt the air around her shimmer. “Your purpose is not a single path, but a tapestry woven from your choices, your kindness, and your dreams. Embrace your journey, and you will find your way.”With that, the whispers faded, and a sense of peace enveloped Elara. She realized that her journey was just beginning. Thanking the Heartwood, she made her way back through the woods, feeling lighter and more connected to the world around her.When she returned to the village, Elara shared her experience with the villagers. Inspired by her bravery, others began to explore the Whispering Woods, discovering their own paths and purposes.From that day on, the village thrived, with each person finding their unique place in the world. And the Whispering Woods remained a sanctuary of wisdom, welcoming all who sought its secrets.And so, the story of Elara and the Whispering Woods became a cherished tale, reminding everyone that the journey of self-discovery is filled with wonder and possibility."
}
```


base64之后

ewoJImlzX2ZpbmFsIjogdHJ1ZSwKCSJzdHJlYW1faWQiOiAwLAoJIm1lc3NhZ2VfaWQiOiAiYWJ
jZGVmZ2giLAoJImRhdGFfdHlwZSI6ICJ0cmFuc2NyaWJlIiwKCSJ0ZXh0X3RzIjogMTExMTExMT
ExMSwKCSJ0ZXh0IjogIk9uY2UgdXBvbiBhIHRpbWUsIGluIGEgc21hbGwgdmlsbGFnZSBuZXN0b
GVkIGJldHdlZW4gcm9sbGluZyBoaWxscywgdGhlcmUgd2FzIGEgbXlzdGVyaW91cyBmb3Jlc3Qg
a25vd24gYXMgdGhlIFdoaXNwZXJpbmcgV29vZHMuIFRoZSB2aWxsYWdlcnMgb2Z0ZW4gc3Bva2U
gb2YgdGhlIHdvb2RzIGluIGh1c2hlZCB0b25lcywgZm9yIGl0IHdhcyBzYWlkIHRoYXQgdGhlIH
RyZWVzIGNvdWxkIHNwZWFrLCBzaGFyaW5nIHNlY3JldHMgb2YgdGhlIHBhc3QgYW5kIHZpc2lvb
nMgb2YgdGhlIGZ1dHVyZS4gSG93ZXZlciwgZmV3IGRhcmVkIHRvIGVudGVyLCBmZWFyaW5nIHRo
ZSB1bmtub3duLk9uZSBkYXksIGEgY3VyaW91cyBnaXJsIG5hbWVkIEVsYXJhIGRlY2lkZWQgdG8
gZXhwbG9yZSB0aGUgd29vZHMuIFNoZSBoYWQgYWx3YXlzIGJlZW4gZmFzY2luYXRlZCBieSB0aG
UgdGFsZXMgdG9sZCBieSB0aGUgdmlsbGFnZSBlbGRlcnMuIFdpdGggYSBoZWFydCBmdWxsIG9mI
GNvdXJhZ2UgYW5kIGEgc2Vuc2Ugb2YgYWR2ZW50dXJlLCBzaGUgc2V0IG9mZiBhdCBkYXduLCBs
ZWF2aW5nIGJlaGluZCB0aGUgZmFtaWxpYXIgc291bmRzIG9mIGhlciB2aWxsYWdlLkFzIEVsYXJ
hIHN0ZXBwZWQgaW50byB0aGUgV2hpc3BlcmluZyBXb29kcywgc2hlIHdhcyBncmVldGVkIGJ5IG
EgZ2VudGxlIGJyZWV6ZSB0aGF0IHJ1c3RsZWQgdGhlIGxlYXZlcyBhYm92ZS4gSXQgZmVsdCBhc
yBpZiB0aGUgdHJlZXMgd2VyZSB3ZWxjb21pbmcgaGVyLiBUaGUgZGVlcGVyIHNoZSB2ZW50dXJl
ZCwgdGhlIG1vcmUgYWxpdmUgdGhlIGZvcmVzdCBzZWVtZWQuIFNoZSBjb3VsZCBoZWFyIGZhaW5
0IHdoaXNwZXJzLCBsaWtlIGEgc29mdCBtZWxvZHkgY2FycmllZCBieSB0aGUgd2luZC7igJxXaG
/igJlzIHRoZXJlP+KAnSBFbGFyYSBjYWxsZWQgb3V0LCBoZXIgdm9pY2UgZWNob2luZyBpbiB0a
GUgc3RpbGxuZXNzLiBUbyBoZXIgc3VycHJpc2UsIHRoZSB3aGlzcGVycyBncmV3IGxvdWRlciwg
Zm9ybWluZyB3b3JkcyB0aGF0IGRhbmNlZCBhcm91bmQgaGVyLuKAnFNlZWsgdGhlIGhlYXJ0IG9
mIHRoZSB3b29kcywgYW5kIHlvdSBzaGFsbCBmaW5kIHdoYXQgeW91IHNlZWss4oCdIHRoZSB0cm
VlcyBtdXJtdXJlZCBpbiB1bmlzb24uSW50cmlndWVkLCBFbGFyYSBmb2xsb3dlZCB0aGUgc291b
mQsIGhlciBoZWFydCByYWNpbmcgd2l0aCBhbnRpY2lwYXRpb24uIEFmdGVyIHdhbGtpbmcgZm9y
IHdoYXQgZmVsdCBsaWtlIGhvdXJzLCBzaGUgY2FtZSB1cG9uIGEgY2xlYXJpbmcgYmF0aGVkIGl
uIHN1bmxpZ2h0LiBJbiB0aGUgY2VudGVyIHN0b29kIGFuIGFuY2llbnQgdHJlZSwgaXRzIHRydW
5rIHdpZGUgYW5kIGduYXJsZWQsIHdpdGggYnJhbmNoZXMgdGhhdCByZWFjaGVkIHRvd2FyZHMgd
GhlIHNreS4gSXQgd2FzIHRoZSoldGVzdCB0cmVlIGluIHRoZSBmb3Jlc3QsIGtub3duIGFzIHB0
aGUgSGVhcnR3b29kLkFwcHJvYWNoaW5nIHRoZSBIZWFydHdvb2QsIEVsYXJhIHBsYWNlZCBoZXI
gaGFuZCBvbiBpdHMgcm91Z2ggYmFyay4gU3VkZGVubHksIHNoZSBmZWx0IGEgd2FybXRoIHNwcm
VhZGluZyB0aHJvdWdoIGhlciBmaW5nZXJzLCBhbmQgdGhlIHdoaXNwZXJzIHRyYW5zZm9ybWVkI
GludG8gYSBjbGVhciB2b2ljZS4g4oCcWW91IGhhdmUgc2hvd24gZ3JlYXQgY291cmFnZSwgeW91
bmcgb25lLiBXaGF0IGlzIGl0IHRoYXQgeW91IHNlZWs
/4oCdRWxhcmEgdG9vayBhIGRlZXAgYnJlYXRoLiDigJxJIHNlZWsga25vd2xlZGdlIGFuZCB1bm
RlcnN0YW5kaW5nLiBJIHdhbnQgdG8ga25vdyBteSBwdXJwb3NlIGluIHRoaXMgd29ybGQu4oCdV
GhlIEhlYXJ0d29vZCBwYXVzZWQsIGFuZCBFbGFyYSBmZWx0IHRoZSBhaXIgYXJvdW5kIGhlciBz
aGltbWVyLiDigJxZb3VyIHB1cnBvc2UgaXMgbm90IGEgc2luZ2xlIHBhdGgsIGJ1dCBhIHRhcGV
zdHJ5IHdvdmVuIGZyb20geW91ciBjaG9pY2VzLCB5b3VyIGtpbmRuZXNzLCBhbmQgeW91ciBkcm
VhbXMuIEVtYnJhY2UgeW91ciBqb3VybmV5LCBhbmQgeW91IHdpbGwgZmluZCB5b3VyIHdheS7ig
J1XaXRoIHRoYXQsIHRoZSB3aGlzcGVycyBmYWRlZCwgYW5kIEElc2Vuc2Ugb2YgcGVhY2UgZW52
ZWxvcGVkIEVsYXJhLiBTaGUgcmVhbGl6ZWQgdGhhdCBoZXIgam91cm5leSB3YXMganVzdCBiZWd
pbm5pbmcuIFRoYW5raW5nIHRoZSBIZWFydHdvb2QsIHNoZSBtYWRlIGhlciB3YXkgYmFjayB0aH
JvdWdoIHRoZSB3b29kcywgZmVlbGluZyBsaWdodGVyIGFuZCBtb3JlIGNvbm5lY3RlZCB0byB0a
GUgd29ybGQgYXJvdW5kIGhlci5XaGVuIHNoZSByZXR1cm5lZCB0byB0aGUgdmlsbGFnZSwgRWxh
cmEgc2hhcmVkIGhlciBleHBlcmllbmNlIHdpdGggdGhlIHZpbGxhZ2Vycy4gSW5zcGlyZWQgYnk
gaGVyIGJyYXZlcnksIG90aGVycyBiZWdhbiB0byBleHBsb3JlIHRoZSBXaGlzcGVyaW5nIFdvb2
RzLCBkaXNjb3ZlcmluZyB0aGVpciBvd24gcGF0aHMgYW5kIHB1cnBvc2VzLkZyb20gdGhhdCBkY
Xkgb24sIHRoZSB2aWxsYWdlIHRocml2ZWQsIHdpdGggZWFjaCBwZXJzb24gZmluZGluZyB0aGVp
ciB1bmlxdWUgcGxhY2UgaW4gdGhlIHdvcmxkLiBBbmQgdGhlIFdoaXNwZXJpbmcgV29vZHMgcmV
tYWluZWQgYSBzYW5jdHVhcnkgb2Ygd2lzZG9tLCB3ZWxjb21pbmcgYWxsIHdobyBzb3VnaHQgaX
RzIHNlY3JldHMuQW5kIHNvLCB0aGUgc3Rvcnkgb2YgRWxhcmEgYW5kIHRoZSBXaGlzcGVyaW5nI

```
Fdvb2RzIGJlY2FtZSBhIGNoZXJpc2hlZCB0YWxlLCByZW1pbmRpbmcgZXZlcnlvbmUgdGhhdCB0
aGUgam91cm5leSBvZiBzZWxmLWRpc2NvdmVyeSBpcyBmaWxsZWQgd2l0aCB3b25kZXIgYW5kIHB
vc3NpYmlsaXR5LiIKfQ==
```

分包后（假设分包为5段）

```
abcdefgh|0|5|ewoJImlzX2Zpb...
abcdefgh|1|5|ZWF2aW5nI...
abcdefgh|2|5|BoZWFydCByYW...
abcdefgh|3|5|G8ga25vdyBt...
abcdefgh|4|5|d24gcGF0aHMgY...
```

附录

音色列表：

- 青涩青年音色：male-qn-qingse
- 精英青年音色：male-qn-jingying
- 霸道青年音色：male-qn-badao
- 青年大学生音色：male-qn-daxuesheng
- 少女音色：female-shaonv
- 御姐音色：female-yujie
- 成熟女性音色：female-chengshu
- 甜美女性音色：female-tianmei
- 男性主持人：presenter_male
- 女性主持人：presenter_female
- 男性有声书1：audiobook_male_1
- 男性有声书2：audiobook_male_2
- 女性有声书1：audiobook_female_1
- 女性有声书2：audiobook_female_2
- 青涩青年音色-beta：male-qn-qingse-jingpin
- 精英青年音色-beta：male-qn-jingying-jingpin
- 霸道青年音色-beta：male-qn-badao-jingpin
- 青年大学生音色-beta：male-qn-daxuesheng-jingpin
- 少女音色-beta：female-shaonv-jingpin
- 御姐音色-beta：female-yujie-jingpin
- 成熟女性音色-beta：female-chengshu-jingpin
- 甜美女性音色-beta：female-tianmei-jingpin
- 聪明男童：clever_boy
- 可爱男童：cute_boy
- 萌萌女童：lovely_girl
- 卡通猪小琪：cartoon_pig
- 俊朗男友：junlang_nanyou
- 甜心小玲：tianxin_xiaoling
- 俏皮萌妹：qiaopi_mengmei
- 妩媚御姐：wumei_yujie
- 嗲嗲学妹：diadia_xuemei
- 淡雅学姐：danya_xuejie
- 霸道少爷：badao_shaoye
- 冷淡学长：lengdan_xiongzhang
- 纯真学弟：chunzhen_xuedi

错误码

错误码	描述
400	请求参数异常
403	未授权访问

409	Agent冲突
422	访问超限
502	网关异常
503	Agent启动错误
504	超时异常