在目录中筛选
多模态
视觉理解
视觉推理
文字提取
界面交互
音频理解
全模态
实时多模态
语音合成
实时语音合成-CosyVoice/Sambert
实时语音合成-通义千问
语音合成-通义千问
语音识别
实时语音识别-Fun-ASR/Gummy/Paraformer
实时语音识别-通义千问
录音文件识别-Fun-ASR/Paraformer/SenseVoice
录音文件识别-通义千问
语音翻译
实时语音翻译-Gummy
实时音视频翻译-通义千问

实时语音合成-通义千问

更新时间：2026-01-17 00:23:59

实时语音合成-通义千问提供低延迟、流式文本输入与流式音频输出能力，提供多种拟人音色，支持多语种/方言合成，可在同一音色下输出多语种，并能自适应调节语气，流畅处理复杂文本。

核心功能

- 实时生成高保真语音，支持中英等多语种自然发声
- 提供**声音复刻**（基于参考音频复刻音色）与**声音设计**（通过文本描述生成音色）两种音色定制方式，快速定制个性化音色
- 支持流式输入输出，低延迟响应实时交互场景
- 可调节语速、语调、音量与码率，精细控制语音表现
- 兼容主流音频格式，最高支持 48kHz 采样率输出

适用范围

支持的模型：

中国内地 | 国际

在中国内地部署模式下，接入点与数据存储均位于**北京地域**，模型推理计算资源仅限于中国内地。

调用以下模型时，请选择北京地域的 API Key：

- **通义千问** 3-TTS-VD-Realtime：qwen3-tts-vd-realtime-2025-12-16（快照版）
- **通义千问** 3-TTS-VC-Realtime：qwen3-tts-vc-realtime-2026-01-15（最新快照版）、qwen3-tts-vc-realtime-2025-11-27（快照版）
- **通义千问** 3-TTS-Flash-Realtime：qwen3-tts-flash-realtime（稳定版，当前等同 qwen3-tts-flash-realtime-2025-11-27）、qwen3-tts-flash-realtime-2025-11-27（最新快照版）、qwen3-tts-flash-realtime-2025-09-18（快照版）
- **通义千问**-TTS-Realtime：qwen-tts-realtime（稳定版，当前等同 qwen-tts-realtime-2025-07-15）、qwen-tts-realtime-latest（最新版，当前等同 qwen-tts-realtime-2025-07-15）、qwen-tts-realtime-2025-07-15（快照版）

更多信息请参见模型列表

模型选型

场景	推荐模型	理由	注意事项
品牌形象、专属声音、扩展系统音色等语音定制（基于文本描述）	qwen3-tts-vd-realtime-2025-12-16	支持声音设计，无需音频样本，通过文本描述创建定制化音色，适合从零开始设计品牌专属声音	不支持使用系统音色，不支持声音复刻
品牌形象、专属声音、扩展系统音色等语音定制（基于音频样本）	qwen3-tts-vc-realtime-2026-01-15	支持声音复刻，基于真实音频样本快速复刻音色，打造拟人化品牌声纹，确保音色高度还原与一致性	不支持使用系统音色，不支持声音设计
智能客服与对话机器人	qwen3-tts-flash-realtime-2025-11-27	支持流式输入输出，可调节语速音高，提供自然交互体验；多音频格式输出适配不同终端	仅支持系统音色，不支持声音复刻/设计
多语种内容播报	qwen3-tts-flash-realtime-2025-11-27	支持多种语言与中文方言，覆盖全球化内容分发需求	仅支持系统音色，不支持声音复刻/设计
有声阅读与内容生产	qwen3-tts-flash-realtime-2025-11-27	可调节音量、语速、音高，满足有声书、播客等内容精细化制作需求	仅支持系统音色，不支持声音复刻/设计
电商直播与短视频配音	qwen3-tts-flash-realtime-2025-11-27	支持 mp3/opus 压缩格式，适合带宽受限场景；可调节参数满足不同风格配音需求	仅支持系统音色，不支持声音复刻/设计

更多说明请参见模型功能特性对比

快速开始

运行代码前，需要获取并配置 API Key。如果通过 SDK 调用，还需要安装最新版 DashScope SDK。

使用系统音色进行语音合成 | 使用声音复刻音色进行语音合成 | 使用声音设计音色进行语音合成

以下示例演示如何使用系统音色（参见音色列表）进行语音合成。

使用DashScope SDK | 使用WebSocket API

Python | Java

server commit模式 | commit模式

```python
import os
import base64
import threading
import time
import dashscope
from dashscope.audio.qwen_tts_realtime import *

qwen_tts_realtime: QwenTtsRealtime = None
text_to_synthesize = [
```

本页导读
- 核心功能
- 适用范围
- 模型选型
- 快速开始
- 交互流程
- API参考
- 模型功能特性对比
- 支持的音色

```
    '对吧~我就特别喜欢这种超市，',
    '尤其是过年的时候',
    '去逛超市',
    '就会觉得',
    '超级超级开心！',
    '想买好多好多的东西呢！'
]

DO_VIDEO_TEST = False

def init_dashscope_api_key():
    """
        Set your DashScope API-key. More information:
        https://github.com/aliyun/alibabacloud-bailian-speech-demo/blob/master/PREREQUISITES.md
    """

    # 新加坡和北京地域的API Key不同。获取API Key: https://help.aliyun.com/zh/model-studio/get-api-key
    if 'DASHSCOPE_API_KEY' in os.environ:
        dashscope.api_key = os.environ[
            'DASHSCOPE_API_KEY']  # load API-key from environment variable DASHSCOPE_API_KEY
    else:
        dashscope.api_key = 'your-dashscope-api-key'  # set API-key manually



class MyCallback(QwenTtsRealtimeCallback):
    def __init__(self):
        self.complete_event = threading.Event()
        self.file = open('result_24k.pcm', 'wb')

    def on_open(self) -> None:
        print('connection opened, init player')

    def on_close(self, close_status_code, close_msg) -> None:
        self.file.close()
        print('connection closed with code: {}, msg: {}, destroy player'.format(close_status_code, close_msg))

    def on_event(self, response: str) -> None:
        try:
            global qwen_tts_realtime
            type = response['type']
            if 'session.created' == type:
                print('start session: {}'.format(response['session']['id']))
            if 'response.audio.delta' == type:
                recv_audio_b64 = response['delta']
                self.file.write(base64.b64decode(recv_audio_b64))
            if 'response.done' == type:
                print(f'response {qwen_tts_realtime.get_last_response_id()} done')
            if 'session.finished' == type:
                print('session finished')
                self.complete_event.set()
        except Exception as e:
            print('[Error] {}'.format(e))
            return

    def wait_for_finished(self):
        self.complete_event.wait()


if __name__  == '__main__':
    init_dashscope_api_key()

    print('Initializing ...')

    callback = MyCallback()

    qwen_tts_realtime = QwenTtsRealtime(
        model='qwen3-tts-flash-realtime',
        callback=callback,
        # 以下为北京地域url，若使用新加坡地域的模型，需将url替换为：wss://dashscope-intl.aliyuncs.com/api-ws/v1/realtime
        url='wss://dashscope.aliyuncs.com/api-ws/v1/realtime'
        )

    qwen_tts_realtime.connect()
    qwen_tts_realtime.update_session(
        voice = 'Cherry',
        response_format = AudioFormat.PCM_24000HZ_MONO_16BIT,
        mode = 'server_commit'
    )
    for text_chunk in text_to_synthesize:
        print(f'send texd: {text_chunk}')
        qwen_tts_realtime.append_text(text_chunk)
        time.sleep(0.1)
    qwen_tts_realtime.finish()
    callback.wait_for_finished()
    print('[Metric] session: {}, first audio delay: {}'.format(
                    qwen_tts_realtime.get_session_id(),
                    qwen_tts_realtime.get_first_audio_delay(),
                    ))
```

更多示例代码请参见 github。

交互流程

server_commit 模式　commit 模式

将 `session.update` 事件的 `session.mode` 设为 `"server_commit"` 以启用该模式，服务端会智能处理文本分段和合成时机。

交互流程如下：

1. 客户端发送 `session.update` 事件，服务端响应 `session.created` 与 `session.updated` 事件。
2. 客户端发送 `input_text_buffer.append` 事件追加文本至服务端缓冲区。
3. 服务端智能处理文本分段和合成时机，并返回 `response.created` 、 `response.output_item.added` 、 `response.content_part.added` 、 `response.audio.delta` 事件。
4. 服务端响应完成后响应 `response.audio.done` 、 `response.content_part.done` 、 `response.output_item.done` 、 `response.done` 。
5. 服务端响应 `session.finished` 来结束会话。

生命周期	客户端事件	服务器事件
会话初始化	session.update 会话配置	session.created 会话已创建 session.updated 会话配置已更新
用户文本输入	input_text_buffer.append 添加文本到服务端 input_text_buffer.commit 立即合成服务端缓存的文本 session.finish 通知服务端不再有文本输入	input_text_buffer.committed 服务端收到提交的文本
服务器音频输出	无	response.created 服务端开始生成响应 response.output_item.added 响应时有新的输出内容 response.content_part.added 新的输出内容添加到 assistant message response.audio.delta 模型增量生成的音频 response.content_part.done Assistant message 的文本或音频内容流式输出完成 response.output_item.done Assistant message 的整个输出项流式传输完成 response.audio.done 音频生成完成 response.done 响应完成

API 参考

实时语音合成-通义千问 API 参考

声音复刻-API 参考

声音设计-API 参考

模型功能特性对比

功能/特性	qwen3-tts-vd-realtime-2025-12-16	qwen3-tts-vc-realtime-2026-01-15、qwen3-tts-vc-realtime-2025-11-27	qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27、qwen3-tts-flash-realtime-2025-09-18	qwen-tts-realtime、qwen-tts-realtime-latest、qwen-tts-realtime-2025-07-15
支持语言	中文、英文、西班牙语、俄语、意大利语、法语、韩语、日语、德语、葡萄牙语		中文（普通话、北京、上海、四川、南京、陕西、闽南、天津、粤语，因音色而异）、英文、西班牙语、俄语、意大利语、法语、韩语、日语、德语、葡萄牙语	中文、英文
音频格式	pcm、wav、mp3、opus			pcm
音频采样率	8kHz、16kHz、24kHz、48kHz			24kHz
声音复刻	⊗	✔	⊗	
声音设计	✔	⊗		
SSML	⊗			
LaTeX	⊗			
音量调节	✔			⊗
语速调节	✔			⊗
语调（音高）调节	✔			⊗
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
			qwen3-tts-flash-realtime、	

限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API			
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符		中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API			
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符		中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API			
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符		中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000

接入方式	Java/Python SDK、WebSocket API		
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型
码率调节	✔		⊗
时间戳	⊗		
设置情感	⊗		
流式输入	✔		
流式输出	✔		
限流	每分钟调用次数（RPM）：180	qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API		
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型
码率调节	✔		⊗
时间戳	⊗		
设置情感	⊗		
流式输入	✔		
流式输出	✔		
限流	每分钟调用次数（RPM）：180	qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API		
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型
码率调节	✔		⊗
时间戳	⊗		
设置情感	⊗		
流式输入	✔		
流式输出	✔		
限流	每分钟调用次数（RPM）：180	qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API		
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API			
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符		中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API			
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符		中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗
时间戳	⊗			
设置情感	⊗			
流式输入	✔			
流式输出	✔			
限流	每分钟调用次数（RPM）：180		qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API			
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符		中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型	
码率调节	✔			⊗

时间戳	⊗		
设置情感	⊗		
流式输入	✔		
流式输出	✔		
限流 **码率调节**	每分钟调用次数（RPM）：180 ✔	qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）： ⊗
时间戳	⊗		
设置情感	⊗		
流式输入	✔		
流式输出	✔		
限流	每分钟调用次数（RPM）：180	qwen3-tts-flash-realtime、qwen3-tts-flash-realtime-2025-11-27 每分钟调用次数（RPM）：180 qwen3-tts-flash-realtime-2025-09-18 每分钟调用次数（RPM）：10	每分钟调用次数（RPM）：10 每分钟消耗 Token 数（TPM）：100,000
接入方式	Java/Python SDK、WebSocket API		
价格	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地：1 元/万字符 国际：0.954101 元/万字符	中国内地： • 输入成本：0.0024 元/千 Token • 输出成本：0.012 元/千 Token

支持的音色

不同模型支持的音色有所差异，使用时将请求参数 `voice` 设置为音色列表中 voice **参数**列对应的值。

`voice` 参数	详情	支持语种	支持模型