

语音合成CosyVoice Python SDK

更新时间：2025-11-20 10:30:41

[产品详情](#)

[我的收藏](#)

本文介绍语音合成CosyVoice Python SDK的参数和接口细节。

用户指南：关于模型介绍和选型建议请参见[实时语音合成-CosyVoice/Sambert](#)。

前提条件

- 已开通服务并[获取API Key](#)。请[配置API Key到环境变量](#)，而非硬编码在代码中，防范因代码泄露导致的安全风险。

说明

当您需要为第三方应用或用户提供临时访问权限，或者希望严格控制敏感数据访问、删除等高风险操作时，建议使用[临时鉴权Token](#)。

与长期有效的 API Key 相比，临时鉴权 Token 具备时效性短（60秒）、安全性高的特点，适用于临时调用场景，能有效降低API Key泄露的风险。

使用方式：在代码中，将原本用于鉴权的 API Key 替换为获取到的临时鉴权 Token 即可。

- [安装最新版DashScope SDK](#)。

模型与价格

模型名称	单价	免费额度
cosyvoice-v3-plus	2元/万字符	2025年11月15日0点前开通阿里云百炼：2000字符2025年11月15日0点后开通阿里云百炼：1万字符有效期：阿里云百炼开通后90天内
cosyvoice-v3-flash	1元/万字符	
cosyvoice-v2	2元/万字符	

模型名称 cosyvoice-v1	付单价	免费额度

语音合成文本限制与格式规范

文本长度限制

- 非流式调用（[同步调用](#)或[异步调用](#)）：单次发送文本长度不得超过 2000 字符。
- [流式调用](#)：单次发送文本长度不得超过 2000 字符，且累计发送文本总长度不得超过 20 万字符。

字符计算规则

- 汉字（包括简/繁体汉字、日文汉字和韩文汉字）按2个字符计算，其他所有字符（如标点符号、字母、数字、日韩文假名/谚文等）均按 1个字符计算
- 计算文本长度时，不包含SSML 标签内容
- 示例：
 - "你好" → 2(你)+2(好)=4字符
 - "中A文123" → 2(中)+1(A)+2(文)+1(1)+1(2)+1(3)=8字符
 - "中文。" → 2(中)+2(文)+1(。) =5字符
 - "中 文。" → 2(中)+1(空格)+2(文)+1(。) =6字符
 - "你好" → 2(你)+2(好)=4字符

编码格式

需采用UTF-8编码。

数学表达式支持说明

当前数学表达式解析功能仅适用于 `cosyvoice-v2`、`cosyvoice-v3-flash` 和 `cosyvoice-v3-plus` 模型，支持识别中小学常见的数学表达式，包括但不限于基础运算、代数、几何等内容。

详情请参见[LaTeX 公式转语音](#)。

SSML标记语言支持说明

当前SSML（Speech Synthesis Markup Language，语音合成标记语言）功能仅 `cosyvoice-v2` 模型的部分音色可用，使用时需满足以下条件：

- 使用[DashScope SDK](#) 1.23.4 或更高版本
- 仅支持[同步调用](#)和[异步调用](#)（即[SpeechSynthesizer](#)类的 `call` 方法），不支持[流式调用](#)（即[SpeechSynthesizer](#)类的 `streaming_call` 方法）
- 使用方法与普通语音合成一致：将包含SSML的文本传入[SpeechSynthesizer](#)类的 `call` 方法即可

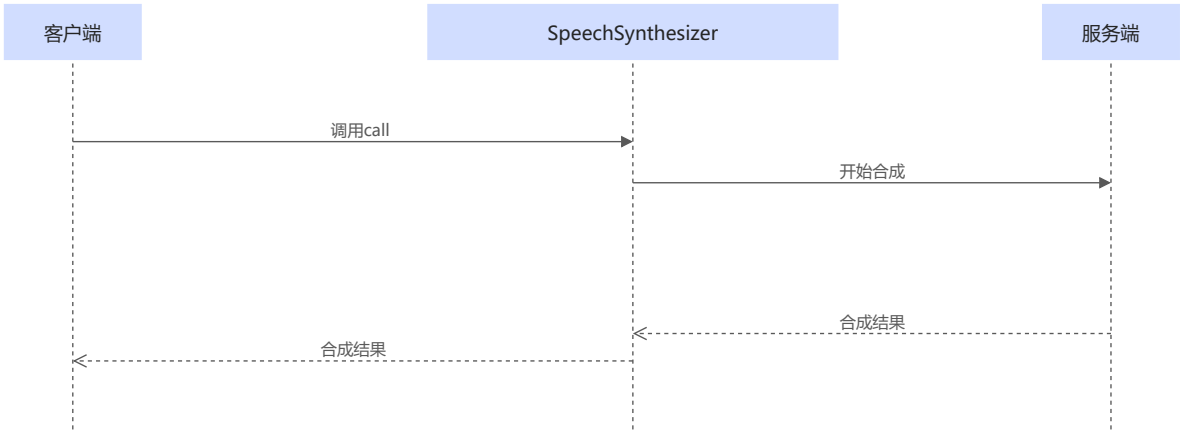
快速开始

[SpeechSynthesizer](#)类提供了语音合成的关键接口，支持以下几种调用方式：

- 同步调用：阻塞式，一次性发送完整文本，直接返回完整音频。适合短文本语音合成场景。
- 异步调用：非阻塞式，一次性发送完整文本，通过回调函数接收音频数据（可能分片）。适用于对实时性要求高的短文本语音合成场景。
- 流式调用：非阻塞式，可分多次发送文本片段，通过回调函数实时接收增量合成的音频流。适合实时性要求高的长文本语音合成场景。

同步调用

提交单个语音合成任务，无需调用回调函数，进行语音合成（无流式输出中间结果），最终一次性获取完整结果。



实例化SpeechSynthesizer类绑定请求参数，调用 call 方法进行合成并获取二进制音频数据。
发送的文本长度不得超过2000字符（详情请参见SpeechSynthesizer类的 call 方法）。

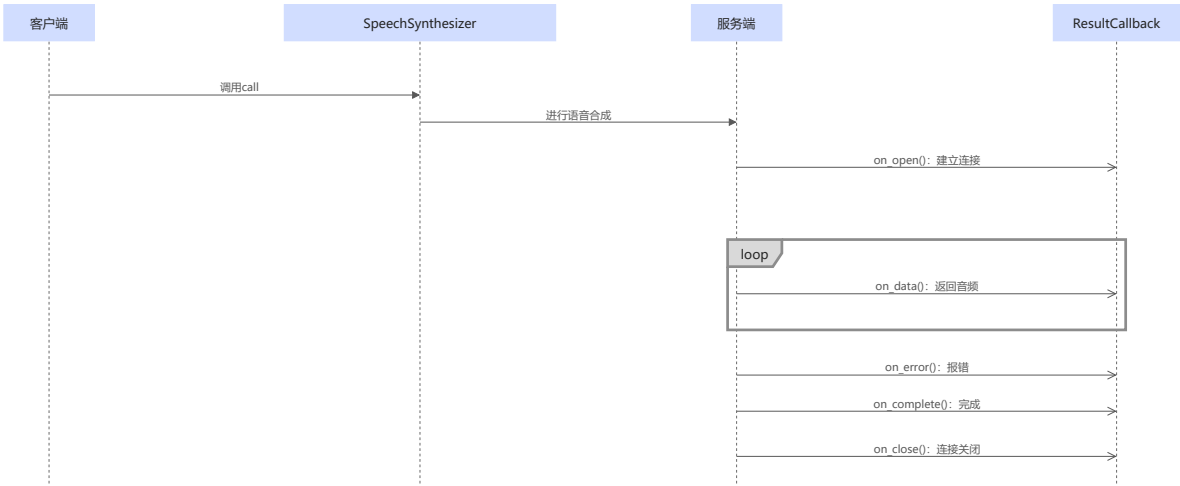
重要

每次调用 call 方法前，需要重新初始化 speechSynthesizer 实例。

点击查看完整示例

异步调用

提交单个语音合成任务，通过回调的方式流式输出中间结果，合成结果通过 ResultCallback 中的回调函数流式获取。



实例化SpeechSynthesizer类绑定请求参数和回调接口（ResultCallback），调用 call 方法进行合成并通过回调接口（ResultCallback）的 on_data 方法实时获取合成结果。

发送的文本长度不得超过2000字符（详情请参见SpeechSynthesizer类的 call 方法）。

重要

每次调用 call 方法前，需要重新初始化 speechSynthesizer 实例。

点击查看完整示例

流式调用

在同一个语音合成任务中分多次提交文本，并通过回调的方式实时获取合成结果。

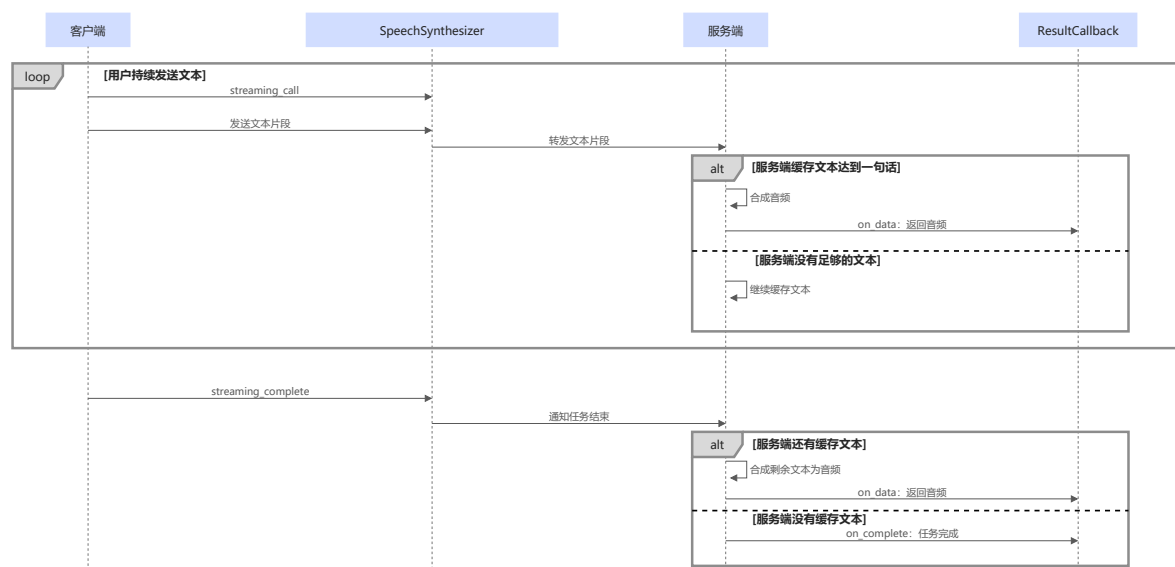
说明

- 流式输入时可多次调用 `streaming_call` 按顺序提交文本片段。服务端接收文本片段后自动进行分句：
 - 完整语句立即合成
 - 不完整语句缓存至完整后合成

调用 `streaming_complete` 时，服务端会强制合成所有已接收但未处理的文本片段（包括未完成的句子）。

- 发送文本片段的间隔不得超过23秒，否则触发“request timeout after 23 seconds”异常。若无待发送文本，需及时调用 `streaming_complete` 结束任务。

服务端强制设定23秒超时机制，客户端无法修改该配置。



1. 实例化SpeechSynthesizer类

实例化[SpeechSynthesizer类](#)绑定[请求参数](#)和[回调接口（ResultCallback）](#)。

2. 流式传输

多次调用[SpeechSynthesizer类](#)的 `streaming_call` 方法分片提交待合成文本，将待合成文本分段发送至服务端。

在发送文本的过程中，服务端会通过[回调接口（ResultCallback）](#)的 `on_data` 方法，将合成结果实时返回给客户端。

每次调用 `streaming_call` 方法发送的文本片段（即 `text`）长度不得超过2000字符，累计发送的文本总长度不得超过20万字符。

3. 结束处理

调用[SpeechSynthesizer类](#)的 `streaming_complete` 方法结束语音合成。

该方法会阻塞当前线程，直到[回调接口（ResultCallback）](#)的 `on_complete` 或者 `on_error` 回调触发后才会释放线程阻塞。

请务必确保调用该方法，否则可能会导致结尾部分的文本无法成功转换为语音。

[点击查看完整示例](#)

请求参数

请求参数通过[SpeechSynthesizer类](#)的构造方法进行设置。

参数	类型	默认值	是否必须	说明
model	str	-	是	指定模型。不同版本的模型编码方式一致，但使用时须确保模型（model）与音色（voice）匹配：每个版本的模型只能使用本版本的默认音色或专属音色。
voice	str	-	是	指定语音合成所使用的音色。支持默认音色和专属音色： 默认音色 ：参见 音色列表 ， 专属音色 ：通过 声音复刻 功能定制。使用复刻音色时，请确保声音复刻与语音合成使用同一账号。详细操作步骤邀请参见 CosyVoice声音复刻API 。⚠️ 使用声音复刻系列模型合成语音时，仅能使用该模型复刻生成的专属音色，不能使用默认音色。 **⚠️ 使用专属音色合成语音时，语音合成模型（`model`）必须与声音复刻模型（`target_model`）相同。**
format	enum	mp3	否	指定音频编码格式及采样率。若未指定format，则合成音频采样率为22.05kHz，格式为mp3。 说明 默认采样率代表当前音色的最佳采样率，缺省条件下默认按照该采样率输出，同时支持降采样或升采样。可指定的音频编码格式及采样率如下：所有模型均支持的音频编码格式及采样率：AudioFormat.WAV_8000HZ_MONO_16BIT，代表音频格式为wav，采样率为8kHzAudioFormat.WAV_16000HZ_MONO_16BIT，代表音频格式为wav，采样率为16kHzAudioFormat.WAV_22050HZ_MONO_16BIT，代表音频格式为wav，采样率为22.05kHzAudioFormat.WAV_24000HZ_MONO_16BIT，代表音频格式为wav，采样率为24kHzAudioFormat.WAV_44100HZ_MONO_16BIT，代表音频格式为wav，采样率为44.1kHzAudioFormat.WAV_48000HZ_MONO_16BIT，代表音频格式为wav，采样率为48kHzAudioFormat.MP3_8000HZ_MONO_128KBPS，代表音频格式为mp3，采样率为8kHzAudioFormat.MP3_16000HZ_MONO_128KBPS，代表音频格式为mp3，采样率为16kHzAudioFormat.MP3_22050HZ_MONO_256KBPS，代表音频格式为mp3，采样率为22.05kHzAudioFormat.MP3_24000HZ_MONO_256KBPS，代表音频格式为mp3，采样率为24kHzAudioFormat.MP3_44100HZ_MONO_256KBPS，代表音频格式为mp3，采样率为44.1kHzAudioFormat.MP3_48000HZ_MONO_256KBPS，代表音频格式为mp3，采样率为48kHzAudioFormat.PCM_8000HZ_MONO_16BIT，代表音频格式为pcm，采样率为8kHzAudioFormat.PCM_16000HZ_MONO_16BIT，代表音频格式为pcm，采样率为16kHzAudioFormat.PCM_22050HZ_MONO_16BIT，代表音频格式为pcm，采样率为22.05kHzAudioFormat.PCM_24000HZ_MONO_16BIT，代表音频格式为pcm，采样率为24kHzAudioFormat.PCM_44100HZ_MONO_16BIT，代表音频格式为pcm，采样率为44.1kHzAudioFormat.PCM_48000HZ_MONO_16BIT，代表音频格式为pcm，采样率为48kHz除cosyvoice-v1外，其他模型支持的音频编码格式及采样率：音频格式为opus时，支持通过bit_rate参数调整码率。仅对1.24.0及之后版本的DashScope适用。AudioFormat.OGG_OPUS_8KHZ_MONO_32KBPS，代表音频格式为opus，采样率为8kHz，码率为32kbpsAudioFormat.OGG_OPUS_16KHZ_MONO_16KBPS，代表音频格式为opus，采样率为16kHz，码率为16kbpsAudioFormat.OGG_OPUS_16KHZ_MONO_32KBPS，代表音频格式为opus，采样率为16kHz，码率为32kbpsAudioFormat.OGG_OPUS_16KHZ_MONO_64KBPS，代表音频格式为opus，采样率为16kHz，码率为64kbpsAudioFormat.OGG_OPUS_24KHZ_MONO_16KBPS，代表音频格式为opus，采样率为24kHz，码率为16kbpsAudioFormat.OGG_OPUS_24KHZ_MONO_32KBPS，代表音频格式为opus，采样率为24kHz，码率为32kbpsAudioFormat.OGG_OPUS_24KHZ_MONO_64KBPS，代表音频格式为opus，采样率为24kHz，码率为64kbpsAudioFormat.OGG_OPUS_48KHZ_MONO_16KBPS，代表音频格式为opus，采样率为48kHz，码率为16kbpsAudioFormat.OGG_OPUS_48KHZ_MONO_32KBPS，代表音频格式为opus，采样率为48kHz，码率为32kbpsAudioFormat.OGG_OPUS_48KHZ_MONO_64KBPS，代表音频格式为opus，采样率为48kHz，码率为64kbps
volume	int	50	否	合成音频的音量，取值范围：0-100。 重要 该字段在不同版本的DashScope SDK中有所不同：1.20.10及以后版本的SDK：volume1.20.10以前版本的SDK：volumn
speech_rate	float	1.0	否	合成音频的语速，取值范围：0.5-2。0.5：表示默认语速的0.5倍速。1：表示默认语速。默认语速是指模型默认输出的合成语速，语速会因音色不同而略有不同。约每秒钟4个字。2：表示默认语速的2倍速。
pitch_rate	float	1.0	否	合成音频的语调，取值范围：0.5-2。
bit_rate	int	32	否	指定音频的 码率 ，取值范围：6-510kbps。码率越大，音质越好，音频文件体积越大。仅在音频格式（format）为opus时可用。cosyvoice-v1模型不支持该参数。 说明 bit_rate需要通过additional_params参数进行设置：synthesizer = SpeechSynthesizer(model="cosyvoice-v2", voice="longxiaochn_v2", format=AudioFormat.OGG_OPUS_16KHZ_MONO_16KBPS, additional_params={"bit_rate": 32})
word_timestamp_enabled	bool	False	否	是否开启子级别时间戳，默认关闭。仅cosyvoice-v2支持该功能。时间戳结果仅能通过回调接口获取 说明 word_timestamp_enabled需要通过additional_params参数进行设置：synthesizer = SpeechSynthesizer(model="cosyvoice-v2", voice="longxiaochn_v2", callback=callback, # 时间戳结果仅能通过回调接口获取 additional_params={"word_timestamp_enabled": True}) 点击查看完整示例代码
seed	int	0	否	生成时使用的随机数种子，使合成的效果产生变化。默认值0。取值范围：0-65535。cosyvoice-v1不支持该功能。
language_hints	list[str]	-	否	提供语言提示，仅cosyvoice-v3-flash、cosyvoice-v3-plus支持该功能。在语音合成中有如下作用：指定 TN（Text Normalization，文本规范化）处理所用的语言，影响数字、缩写、符号等的朗读方式（仅中文、英文生效）。取值范围：zh：中文en：英文指定语音合成的目标语言（仅限复刻音色），帮助提升合成效果准确性，对英文、法语、德语、日语、韩语、俄语生效（无需填写中文）。韩和声音复刻时使用的languageHints/language_hints一致。取值范围：en：英文fr：法语de：德语ja：日语ko：韩语ru：俄语若设置的语言提示与文本内容明显不符（如为中文文本设置en），将忽略此提示，并依据文本内容自动检测语言。 注意 ：此参数为数组，但当前版本仅处理第一个元素，因此建议只传入一个值。
instruction	String	-	否	设置提示词。仅cosyvoice-v3-flash、cosyvoice-v3-plus支持该功能。在语音合成中有如下作用：指定小语种（仅限复刻音色）格式：“你会用<小语种>说出来。”（注意，结尾一定不要遗漏句号，使用时将<小语种>“替换为具体的小语种，例如替换为德语”。示例：“你会用德语说出来。”支持的语种：法语、德语、日语、韩语、俄语。指定方言（仅限复刻音色）格式：“请用<方言>表达。”（注意，结尾一定不要遗漏句号，使用时将<方言>“替换为具体的方言，例如替换为广东话”。示例：“请用广东话表达。”支持的方言：广东话、东北话、甘肃话、贵州话、河南话、湖北话、江西话、闽南话、宁夏话、山西话、陕西话、山东话、上海话、四川话、天津话、云南话。指定情感、场景、角色或身份等：仅部分默认音色支持该功能，且因音色而异，详情请参见 音色列表
callback	ResultCallback	-	否	回调接口（ResultCallback） 。

关键接口

SpeechSynthesizer 类

SpeechSynthesizer 通过“from dashscope.audio.tts_v2 import *”方式引入，提供语音合成的关键接口。

方法	参数	返回值	描述
def call(self, text: str, timeout_millis=None)	text: 待合成文本 timeout_millis: 阻塞线程的超时时间，单位为毫秒。不设置或值为0时不生效	没有指定 ResultCallback时返回二进制音频数据，否则返回None	将整段文本（无论是纯文本还是包含 SSML 的文本）转换为语音。在创建 SpeechSynthesizer 实例时，存在以下两种情况：没有指定ResultCallback：call方法会阻塞当前线程直到语音合成完成并返回二进制音频数据。使用方法请参见 回调调用 。指定了ResultCallback：call方法会立刻返回None，并通过 回调接口（ResultCallback） 的on_data方法返回语音合成的结果。使用方法请参见 异步调用 。 重要 每次调用call方法前，需要重新初始化SpeechSynthesizer实例。

方法	参数	返回值	描述
<code>def streaming_call(self, text: str)</code>	<code>text</code> : 待合成文本片段	无	流式发送待合成文本（不支持包含SSML的文本）。您可以多次调用该接口，将待合成文本分多次发送给服务端。合成结果通过 回调接口（ResultCallback） 的on_data方法获取。使用方法请参见 流式调用 。
<code>def streaming_complete(self, complete_timeout_millis: int)</code>	<code>complete_timeout_millis</code> : 等待时间，单位为毫秒	无	结束流式语音合成。该方法阻塞当前线程N毫秒（具体时长由complete_timeout_millis决定），直到任务结束。如果completeTimeoutMillis设置为0，则无限期等待。默认情况下，如果等待时间超过10分钟，则停止等待。使用方法请参见 流式调用 。 重要 在 流式调用 时，请务必确保调用该方法，否则可能会出现合成语音缺失的问题。
<code>def get_last_request_id(self)</code>	无	上一个任务的request_id	获取上一个任务的request_id。
<code>def get_first_package_delay(self)</code>	无	首包延迟	获取首包延迟（一般在500ms左右）。首包延迟是开始发送文本和接收第一个音频包之间的时间，单位为毫秒。在任务完成后使用。首次发送文本时需建立 WebSocket 连接，因此首包延迟会包含连接建立的耗时。
<code>def get_response(self)</code>	无	最后一次报文	获取最后一次报文（为JSON格式的数据），可以用于获取task-failed报错。

回调接口（ResultCallback）

[异步调用](#)或[流式调用](#)时，服务端会通过回调的方式，将关键流程信息和数据返回给客户端。您需要实现回调方法，处理服务端返回的信息或者数据。

通过“`from dashscope.audio.tts_v2 import *`”方式引入。

点击查看示例

方法	参数	返回值	描述
<code>def on_open(self) -> None</code>	无	无	当和服务端建立连接完成后，该方法立刻被回调。
<code>def on_event(self, message: str) -> None</code>	<code>message</code> : 服务端返回的信息	无	当服务有回复时会被回调。 <code>message</code> 为JSON字符串，解析可获取Task ID（ <code>task_id</code> 参数）、本次请求中计费的有效字符数（ <code>characters</code> 参数）等信息。
<code>def on_complete(self) -> None</code>	无	无	当所有合成数据全部返回（语音合成完成）后被回调。
<code>def on_error(self, message) -> None</code>	<code>message</code> : 异常信息	无	发生异常时该方法被回调。
<code>def on_data(self, data: bytes) -> None</code>	<code>data</code> : 服务器返回的二进制音频数据	无	当服务器有合成音频返回时被回调。您可以将二进制音频数据合成为一个完整的音频文件后使用播放器播放，也可以通过支持流式播放的播放器实时播放。 重要 流式语音合成中，对于mp3/opus等压缩格式，音频分段传输需使用流式播放器，不可逐帧播放，避免解码失败。支持流式播放的播放器：ffmpeg、pyaudio (Python)、AudioFormat (Java)、MediaSource (Javascript)等。将音频数据合成完整的音频文件时，应以追加模式写入同一文件。流式语音合成的wav/mp3 格式音频仅首帧包含头信息，后续帧为纯音频数据。
<code>def on_close(self) -> None</code>	无	无	当服务已经关闭连接后被回调。

响应结果

服务器返回二进制音频数据：

- [同步调用](#)：对[SpeechSynthesizer](#)类的 `call` 方法返回的二进制音频数据进行处理。
- [异步调用](#)或[流式调用](#)：对[回调接口 \(ResultCallback\)](#) 的 `on_data` 方法的参数（bytes类型数据）进行处理。