

**527meeting**

---

## **TRTC 接口设计说明书**

**文档版本：v1.1.0**

**发布日期：2023年03月08日**

## 版权声明

本手册的所有内容，其著作权归属南京特临信息科技有限公司所有，未经本公司许可，不得以任何方式仿制、拷贝、眷抄或转译。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 修改记录

日期	作者	版本	参考版本	备注
2023-03-08	江水	1.1.0		

# 目录

<b>1. 创建 TRTCCloud 实例.....</b>	<b>1</b>
1.1. 概述.....	1
1.2. 参数.....	1
1.3. 示例.....	1
<b>2. API 调整为异步的接口.....</b>	<b>2</b>
2.1. 概述.....	2
2.2. 接口.....	2
2.3. 示例.....	2
<b>3. 屏幕分享(startScreenCapture) .....</b>	<b>3</b>
3.1. 概述 .....	3
3.2. 示例 .....	3
<b>4. Electron Demo.....</b>	<b>3</b>
4.1. 概述 .....	3
4.2. 附件 .....	3
<b>5. Web Demo .....</b>	<b>4</b>
5.1. 概述 .....	4
5.2. 附件 .....	4
<b>6. 创建会议 .....</b>	<b>4</b>
6.1. 概述 .....	4
6.2. 请求格式 .....	4
6.3. 参数 .....	4
6.4. 返回参数 .....	5
6.5. 传参格式 .....	5
6.6. 示例 .....	5
6.7. 附件 .....	6
<b>7. TRTCCloud 接口支持情况.....</b>	<b>7</b>
7.1. 创建 TRTC 对象 .....	7
7.2. 设置回调 .....	7
7.3. 创建与销毁 TRTCCloud 单例 .....	7
7.4. 房间相关接口函数 .....	7
7.5. CDN 相关接口函数 .....	8
7.6. 视频相关接口函数 .....	8
7.7. 音频相关接口函数 .....	9
7.8. 摄像头相关接口函数 .....	10
7.9. 音频设备相关接口函数 .....	10
7.10. 美颜相关接口函数 .....	10

7.11. 辅流相关接口函数 .....	11
7.12. 自定义消息发送 .....	11
7.13. 背景混音相关接口函数 .....	12
7.14. 音效相关接口函数 .....	12
7.15. 设备和网络测试 .....	13
7.16. LOG 相关接口函数 .....	13
7.17. 弃用接口函数 .....	14
<b>8. 视频通话回调接口支持情况 .....</b>	<b>14</b>
8.1. 错误事件和警告事件 .....	14
8.2. 房间事件回调 .....	14
8.3. 成员事件回调 .....	14
8.4. 统计和质量回调 .....	15
8.5. 服务器事件回调 .....	15
8.6. 硬件设备事件回调 .....	15
8.7. 自定义消息的接收回调 .....	16
8.8. CDN 旁路转推回调 .....	16
8.9. 系统音量采集回调 .....	16
8.10. 音效回调 .....	16
8.11. 屏幕分享回调 .....	16
8.12. 截图回调 .....	17
8.13. 背景混音事件回调 .....	17
<b>9. 服务端 API 接口支持情况 .....</b>	<b>17</b>
9.1. 房间管理相关接口 .....	17
9.2. 混流转推相关接口 .....	17
9.3. 云端录制相关接口 .....	18
9.4. 用量统计相关接口 .....	18
9.5. 监控仪表查询相关接口 .....	18
9.6. 图片相关接口 .....	18
<b>10. 关键类型定义 .....</b>	<b>19</b>
10.1. 关键类型 .....	19
10.2. 枚举值 .....	19
<b>11. 错误码 .....</b>	<b>20</b>
11.1. 错误码表 .....	20
11.1.1. 基础错误码 .....	20
11.1.2. 进房相关错误码 .....	20
11.1.3. 退房相关错误码 .....	21
11.1.4. 设备（摄像头、麦克风、扬声器）相关错误码 .....	21
11.1.5. 屏幕分享相关错误码 .....	22
11.1.6. 编解码相关错误码 .....	22
11.1.7. 自定义采集相关错误码 .....	23
11.1.8. CDN 绑定和混流相关错误码 .....	23
11.1.9. 跨房连麦相关错误码 .....	23

11.2. 警告码表 .....	25
<b>12. 接口详情 .....</b>	<b>27</b>
12.1. 创建函数 (new TRTCCloud()) .....	27
12.1.1. 示例 .....	27
12.1.2. 方法 .....	27
12.2. 房间 .....	28
12.2.1. enterRoom .....	28
12.2.2. exitRoom() .....	30
12.2.3. switchRoom(params) .....	30
12.2.4. switchRole(role) .....	30
12.2.5. connectOtherRoom(params) .....	30
12.2.6. disconnectOtherRoom() .....	31
12.2.7. setDefaultStreamRecvMode(autoRecvAudio, autoRecvVideo) .....	31
12.3. 混流 CDN .....	32
12.3.1. Start-publishing(streamId, type) .....	32
12.3.2. stopPublishing() .....	32
12.3.3. startPublishCDNStream(param) .....	32
12.3.4. stopPublishCDNStream() .....	33
12.3.5. setMixTranscodingConfig(config) .....	33
12.4. 视频 .....	35
12.4.1. startLocalPreview(view) .....	35
12.4.2. stopLocalPreview() .....	36
12.4.3. muteLocalVideo(mute, streamType) .....	36
12.4.4. startRemoteView(userId, view, streamType) .....	36
12.4.5. stopRemoteView(userId, streamType) .....	36
12.4.6. stopAllRemoteView() .....	37
12.4.7. muteRemoteVideoStream(userId, mute, streamType) .....	37
12.4.8. muteAllRemoteVideoStreams(mute) .....	37
12.4.9. setVideoEncoderParam(params) .....	37
12.4.10. setNetworkQosParam(params) .....	38
12.4.11. setLocalRenderParams(params) .....	38
12.4.12. setLocalViewFillMode(mode) .....	39
12.4.13. setRemoteRenderParams(userId, streamType, params) .....	39
12.4.14. setRemoteViewFillMode(userId, mode) .....	40
12.4.15. setLocalViewRotation(rotation) .....	40
12.4.16. setRemoteViewRotation(userId, rotation) .....	40
12.4.17. setVideoEncoderRotation(rotation) .....	41
12.4.18. setLocalViewMirror(mirror) .....	41
12.4.19. setVideoEncoderMirror(mirror) .....	41
12.4.20. enableSmallVideoStream(enable, params) .....	41
12.4.21. setRemoteVideoStreamType(userId, streamType) .....	42
12.4.22. snapshotVideo(userId, streamType) .....	43
12.4.23. setPriorRemoteVideoStreamType(type) .....	43
12.4.24. startLocalRecording(options) .....	43

12.4.25. stopLocalRecording()	44
12.5. 音频	44
12.5.1. startLocalAudio(quality)	44
12.5.2. stopLocalAudio()	44
12.5.3. muteLocalAudio(mute)	44
12.5.4. muteRemoteAudio(userId, mute)	45
12.5.5. muteAllRemoteAudio(mute)	45
12.5.6. setRemoteAudioVolume(userId, volume)	45
12.5.7. setAudioCaptureVolume(volume)	45
12.5.8. getAudioCaptureVolume() → {Number}	45
12.5.9. setAudioPlayoutVolume(volume)	46
12.5.10. getAudioPlayoutVolume() → {Number}	46
12.5.11. enableAudioVolumeEvaluation(interval)	46
12.5.12. startAudioRecording(params) → {Number}	46
12.5.13. stopAudioRecording()	47
12.5.14. setAudioQuality(quality)	47
12.5.15. setMicVolumeOnMixing(volume)	47
12.6. 摄像头	48
12.6.1. getCameraDevicesList() → {Array.<TRTCDeviceInfo>}	48
12.6.2. setCurrentCameraDevice(deviceId)	48
12.6.3. getCurrentCameraDevice() → {TRTCDeviceInfo}	48
12.7. 音频设备	48
12.7.1. getMicDevicesList() → {Array.<TRTCDeviceInfo>}	48
12.7.2. getCurrentMicDevice() → {TRTCDeviceInfo}	49
12.7.3. setCurrentMicDevice(micId)	49
12.7.4. getCurrentMicDeviceVolume() → {Number}	49
12.7.5. setCurrentMicDeviceVolume(volume)	49
12.7.6. setCurrentMicDeviceMute(mute)	49
12.7.7. getCurrentMicDeviceMute() → {Boolean}	49
12.7.8. getSpeakerDevicesList() → {Array.<TRTCDeviceInfo>}	50
12.7.9. getCurrentSpeakerDevice() → {TRTCDeviceInfo}	50
12.7.10. setCurrentSpeakerDevice(speakerId)	50
12.7.11. getCurrentSpeakerVolume() → {Number}	50
12.7.12. setCurrentSpeakerVolume(volume)	50
12.7.13. setCurrentSpeakerDeviceMute(mute)	51
12.7.14. getCurrentSpeakerDeviceMute() → {Boolean}	51
12.8. 音头像前处理	51
12.8.1. setBeautyStyle(style, beauty, white, ruddiness)	51
12.8.2. setWaterMark(streamType, srcData, srcType, nWidth, nHeight, xOffset, yOffset, fWidthRatio)	51
12.9. 辅流（屏幕分享）	52
12.9.1. startRemoteSubStreamView(userId, view)	52
12.9.2. stopRemoteSubStreamView(userId)	52
12.9.3. setRemoteSubStreamViewFillMode(userId, mode)	53
12.9.4. setRemoteSubStreamViewRotation(userId, rotation)	53

12.9.5. getScreenCaptureSources(thumbWidth, thumbHeight, iconWidth, iconHeight) → {Array.<TRTCScreenCaptureSourceInfo>}.....	53
12.9.6. selectScreenCaptureTarget(source, captureRect, property).....	54
12.9.7. startScreenCapture(view, type, params).....	55
12.9.8. pauseScreenCapture() .....	56
12.9.9. resumeScreenCapture().....	56
12.9.10. stopScreenCapture().....	56
12.9.11. setSubStreamEncoderParam(params).....	57
12.9.12. setSubStreamMixVolume(volume) .....	57
12.9.13. addExcludedShareWindow(win).....	58
12.9.14. removeExcludedShareWindow(win).....	58
12.10. 自定义消息发送 .....	58
12.10.1. removeAllExcludedShareWindow() .....	58
12.10.2. sendCustomCmdMsg(cmdId, msg, reliable, ordered) → {Boolean} .....	58
12.11. 背景混音.....	59
12.11.1. playBGM(path) .....	59
12.11.2. stopBGM() .....	59
12.11.3. pauseBGM().....	59
12.11.4. resumeBGM() .....	59
12.11.5. getBGMDuration(path) → {Number} .....	59
12.11.6. setBGMPosition(pos) .....	60
12.11.7. setBGMVolume(volume) .....	60
12.11.8. setBGMPLayoutVolume(volume) .....	60
12.11.9. setBGMPublishVolume(volume) .....	60
12.11.10. startSystemAudioLoopback(path) .....	61
12.11.11. stopSystemAudioLoopback().....	61
12.11.12. setSystemAudioLoopbackVolume(volume) .....	61
12.12. 音效 .....	61
12.12.1. startPlayMusic(musicParam, callbackMap) .....	61
12.12.2. stopPlayMusic(id) .....	62
12.12.3. pausePlayMusic(id) .....	62
12.12.4. resumePlayMusic(id) .....	62
12.12.5. getMusicDurationInMS(path) → {Number}.....	62
12.12.6. seekMusicToPosInTime(id, pts) .....	63
12.12.7. setAllMusicVolume(volume) .....	63
12.12.8. setMusicLayoutVolume(id, volume) .....	63
12.12.9. setMusicPublishVolume(id, volume).....	63
12.12.10. playAudioEffect(effect) .....	64
12.12.11. setAudioEffectVolume(effectId, volume) .....	64
12.12.12. stopAudioEffect(effectId) .....	64
12.12.13. stopAllAudioEffects() .....	64
12.12.14. setAllAudioEffectsVolume(volume) .....	65
12.12.15. pauseAudioEffect(effectId) .....	65
12.12.16. resumeAudioEffect(effectId) .....	65

12.13. 设备和网络测试 .....	65
12.13.1. startSpeedTest(params) .....	65
12.13.2. stopSpeedTest() .....	66
12.13.3. startCameraDeviceTest(view) .....	66
12.13.4. stopCameraDeviceTest() .....	66
12.13.5. startMicDeviceTest(interval) .....	66
12.13.6. stopMicDeviceTest() .....	67
12.13.7. startSpeakerDeviceTest(testAudioFilePath) .....	67
12.13.8. stopSpeakerDeviceTest() .....	67
12.14. LOG .....	67
12.14.1. getSDKVersion() → {String} .....	67
12.14.2. setLogLevel(level) .....	67
12.14.3. setConsoleEnabled(enabled) .....	68
12.14.4. setLogCompressEnabled(enabled) .....	68
12.14.5. setLogDirPath(path) .....	68
12.14.6. setLogCallback(callback) .....	68
12.14.7. callExperimentalAPI(jsonStr) .....	68
12.14.8. setRenderMode(mode) .....	69

## TRTCCloud API 和腾讯 TRTCCloud 接口的差异

TRTCCloud API 是基于腾讯 TRTCCloud 接口定义来编写的，快速替换腾讯原有接口的 API。本套接口可以在 electron 和 web 环境中均可使用，部分接口调用有些差别，后面会着重描述差异接口。

## 1. 创建 TRTCCloud 实例

### 1.1. 概述

构造函数

### 1.2. 参数

参数	类型	是否必选	参数说明
host	String	是	服务器地址
relayServer	String	否	Relay 服务器地址，域名+端口号（域名不要加 https）,默认服务地址和 host 一致
socket	Boolean	否	true 支持呼叫功能（和控制服务器保持 socket 连接），false 不支持,默认 true
nginx	Boolean	否	是否使用 nginx 和流媒体连接
transport	String	否	协议： udp 或 tcp,默认传 tcp
relay	String	否	relay 或 all, 默认传 relay

### 1.3. 示例

```
1. //引用类库
2. import TRTCCloud from './rtc_could';
```

#### 1) 全部参数实例

```
1. trtcCloud= new TRTCCloud({ mode: 'pc', host: 'https://gosun4.527meeting.com/app',
2. relayServer: "gosun4.527meeting.com:30001",relay: 'relay',transport: 'tcp' });
```

#### 2) 服务器指向，其它参数默认实例

```
1. trtcCloud = new TRTCCloud({
2.   host: 'https://gosun4.527meeting.com/app',
3. });
```

#### 3) 只调用本地接口,可以不传参数

```
1. trtcCloud = new TRTCCloud();
```

## 2. API 调整为异步的接口

### 2.1. 概述

下面方法均为异步调用造函数。

### 2.2. 接口

API	描述
getCameraDevicesList	获取摄像头设备列表。
setCurrentCameraDevice	设置要使用的摄像头。
getCurrentCameraDevice	获取当前使用的摄像头。
getMicDevicesList	获取麦克风设备列表。
getCurrentMicDevice	获取当前选择的麦克风。
setCurrentMicDevice	设置要使用的麦克风。
getSpeakerDevicesList	获取扬声器设备列表。
getCurrentSpeakerDevice	获取当前的扬声器设备。
setCurrentSpeakerDevice	设置要使用的扬声器。
getScreenCaptureSources	枚举可共享的窗口列表。 (web 端不能使用)

### 2.3. 示例

```

1.     async getCameraList() {
2.         let tmp = await trtcCloud.getCameraDevicesList();
3.         if (tmp.length === 1 && this.selectedCameraID === '') {
4.             this.selectedCameraID = tmp[0].deviceId;
5.         }
6.         for (var i = 0; i<tmp.length; i++) {
7.             this.cameraList.push({
8.                 text: tmp[i].deviceName,
9.                 value: tmp[i].deviceId,
10.            });
11.        }
12.        logger.log.apply(logger, ['getCameraList', ...tmp]);
13.    }

1.     trtcCloud.setCurrentCameraDevice(this.cameraId);

```

## 3. 屏幕分享(startScreenCapture)

### 3.1. 概述

屏幕分享 startScreenCapture 接口，在 web 端（浏览器）使用时，不要再调用 getScreenCaptureSources 接口，直接调用 startScreenCapture 方法，浏览器会弹出分享源的选择框。

### 3.2. 示例

```
1. if (window.require) {
2.   //Electron 端
3.   trtcCloud.getScreenCaptureSources(200, 160, 0, 0).then(function (sources) {
4.     let source = {
5.       sourceId: sources[0].id,
6.       sourceName: sources[0].name,
7.       type: 1,
8.     };
9.     startScreenShare(source)
10.   });
11. } else {
12.   //web 端
13.   trtcCloud.startScreenCapture(null, TRTCVideoStreamType.TRTCVideoStreamTypeSub, new TRTCVideoEnc
Param(
14.   TRTCVideoResolution.TRTCVideoResolution_1920_1080,
15.   TRTCVideoResolutionMode.TRTCVideoResolutionModeLandscape,
16.   24, 1600, 200,true
17. ));
18. }
```

## 4. Electron Demo

### 4.1. 概述

(1)解压 RTCSimpleDemo.zip  
(2)在终端中 cd RTCSimpleDemo,执行 npm install, 安装成功后, 再执行 npm start 就可以运行当前 demo 了, demo 要运行在公网上, 服务器地址指向的 gosun4.527meeting.com(公网地址)。

### 4.2. 附件

双击附件即可打开



RTCSimpleDem  
o.zip

## 5. Web Demo

### 5.1. 概述

- (1) 解压 api.zip
- (2) 在 api 文夹中, 启本地服务 (<http://localhost:8080> 或 <http://127.0.0.1:8080/>)
- (3) 在 Chrome 浏览器中访问 <http://127.0.0.1:8080/api/>
- (4) 如果是放到非本地服务上, 访问需要使用 https 访问。如 <https://gosun4.527meeting.com/api>,否则页面会报错。

### 5.2. 附件

双击附件即可打开



## 6. 创建会议

### 6.1. 概述

创建即时会议, 可以获取 roomId 或 meetingId。注: 非认证可调用

### 6.2. 请求格式

请求地址	<a href="https://域名/app/meeting">https://域名/app/meeting</a>
请求方式	POST
Content-Type	application/x-www-form-urlencoded

### 6.3. 参数

参数	类型	是否必选	参数说明
module	String	是	值: meeting
method	String	是	方法名: instantMeeting
orgId	number	是	公司唯一标识, 私有部署可传 2
appointmentName	String	是	会议名称
startTime	String	是	会议开始时间, 传入 UTC 时间格式 “yyyyMMddHHmmss”

endTime	String	是	会议结束时间，传入 UTC 时间格式 “yyyyMMddHHmmss”
type	number	是	房间类型 1 音频会议 2 视频会议 3 远程培训
definition	String	否	清晰度 1080p 720p 480p ,不传默认 480p
isPublic	Boolean	是	ture 公共会议室 false 私有会议室
numberOfParticipants	number	否	参会人数，默认 200 方

## 6.4. 返回参数

参数	类型	参数说明
meetingId	number	会议唯一标识
inviteCode	number	参会 code
roomId	number	对应的房间 id
result	number	0 成功 1 失败

## 6.5. 传参格式

key 是 params,value 才是 JSON 对象

## 6.6. 示例

```

1. function http(option) {
2.     //var input = $("#inputImg")[0];
3.     $.ajax({
4.         dataType: 'json',
5.         type: "POST",
6.         contentType: "application/x-www-form-urlencoded; charset=utf-8",
7.         url: 'https://gosun4.527meeting.com/app/meeting',
8.         data: {
9.             params: JSON.stringify(option)

```

```
10.    },
11.    async: true, //默认是 true: 异步, false: 同步。
12.    error: function (data, status, headers) {},
13.    success: function (data) {
14.        document.getElementById('data').innerText = JSON.stringify(data);
15.    }
16. });
17. }
18. function dateToUtcStr(d) {
19.     var y = d.getUTCFullYear(),
20.         M = d.getUTCMonth() + 1,
21.         D = d.getUTCDate(),
22.         h = d.getUTCHours(),
23.         m = d.getUTCMinutes(),
24.         s = d.getUTCSeconds(),
25.         pad = function (x) {
26.             x = x + '';
27.             if (x.length === 1) {
28.                 return '0' + x;
29.             }
30.             return x;
31.         };
32.     return y + pad(M) + pad(D) + pad(h) + pad(m) + pad(s);
33. };
34. function send() {
35.     var todayStart = new Date(); //Utils.getDayStart(new Date());
36.     var tempStart = new Date(todayStart.getTime());
37.     var threeEnd = tempStart.setHours(tempStart.getHours() + 4);
38.     http({
39.         method: 'instantMeeting',
40.         orgId: 2,
41.         type: 2,
42.         startTime: dateToUtcStr(todayStart),
43.         endTime: dateToUtcStr(new Date(threeEnd)),
44.         isPublic: false,
45.         appointmentName: 'test abc'
46.     })
47. }
```

## 6.7. 附件

双击附件即可打开



instantMeeting.  
zip

## 7. TRTCCloud 接口支持情况

### 7.1. 创建 TRTC 对象

```

1. const TRTCCloud = require('trtc-electron-sdk').default;
2. // import TRTCCloud from 'trtc-electron-sdk';
3. this rtcCloud = new TRTCCloud();

```

从v7.9.348起，TRTC Electron SDK 增加了 trtc.d.ts 文件，方便使用 typescript 的开发者

```

1. import TRTCCloud from 'trtc-electron-sdk';
2. const rtcCloud: TRTCCloud = new TRTCCloud();
3. // 获取 SDK 版本号
4. rtcCloud.getSDKVersion();

```

### 7.2. 设置回调

```

1. subscribeEvents = (rtcCloud) => {
2.   rtcCloud.on('onError', (errcode, errmsg) => {
3.     console.info('trtc_demo: onError : ' + errcode + " msg" + errmsg);
4.   });
5.   rtcCloud.on('onEnterRoom', (elapsed) => {
6.     console.info('trtc_demo: onEnterRoom elapsed:' + elapsed);
7.   });
8.   rtcCloud.on('onExitRoom', (reason) => {
9.     console.info('onExitRoom: userenter reason:' + reason);
10.   });
11. };
12. subscribeEvents(this.rtcCloud);

```

### 7.3. 创建与销毁 TRTCCloud 单例

API	描述	是否支持
getTRTCShareInstance	用于动态加载 dll 时，创建 TRTCCloud 对象单例。	支持
destroyTRTCShareInstance	释放 TRTCCloud 单例对象并清理资源。	支持

### 7.4. 房间相关接口函数

API	描述	是否支持
enterRoom	进入房间，若房间不存在，系统将自动创建一个新房间。	支持
exitRoom	退出房间。	支持
switchRoom	切换房间。	待开发
switchRole	切换角色，仅适用于直播场景（TRTCAppSceneLIVE 和	

	TRTCAppSceneVoiceChatRoom)。	
connectOtherRoom	请求跨房连麦（主播跨房 PK）。	
disconnectOtherRoom	关闭跨房连麦（主播跨房 PK）。	
setDefaultStreamRecvMode	设置音视频数据接收模式（需要在进房前设置才能生效）。	

## 7.5. CDN 相关接口函数

API	描述	是否支持
startPublishing	开始向腾讯云的直播 CDN 推流。	待开发
stopPublishing	停止向腾讯云的直播 CDN 推流。	待开发
startPublishCDNStream	开始向非腾讯云的直播 CDN 转推。	待开发
stopPublishCDNStream	停止向非腾讯云的直播 CDN 推流。	待开发
setMixTranscodingConfig	设置云端的混流转码参数。	待开发

## 7.6. 视频相关接口函数

API	描述	是否支持
startLocalPreview	启动本地摄像头采集和预览。	支持
stopLocalPreview	停止本地摄像头采集和预览。	支持
muteLocalVideo	是否屏蔽自己的视频画面。	支持
startRemoteView	开始显示远端视频画面。	支持
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流。	支持
stopAllRemoteView	停止显示所有远端视频画面，同时不再拉取远端用户的视频数据流。	支持
muteRemoteVideoStream	暂停接收指定的远端视频流。	支持
muteAllRemoteVideoStreams	停止接收所有远端视频流。	支持
setVideoEncoderParam	设置视频编码器相关参数。	支持
setNetworkQosParam	设置网络流控相关参数。	支持
setLocalRenderParams	设置本地图像（主流）的渲染参数。	支持
setLocalViewFillMode	废弃接口：设置本地图像的渲染模式。	支持
setRemoteRenderParams	设置远端图像的渲染参数。	支持
setRemoteViewFillMode	废弃接口：设置远端图像的渲染模式。	支持

setLocalViewRotation	废弃接口：设置本地图像的顺时针旋转角度。	支持
setRemoteViewRotation	废弃接口：设置远端图像的顺时针旋转角度。	支持
setVideoEncoderRotation	设置视频编码输出的（即远端用户观看到的以及服务器录制下来的）画面方向。	待开发
setLocalViewMirror	废弃接口：设置本地摄像头预览画面的镜像模式。	支持
setVideoEncoderMirror	设置编码器输出的画面镜像模式。	待开发
enableSmallVideoStream	开启大小画面双路编码模式。默认大中小流	待开发
setRemoteVideoStreamType	选定观看指定 userId 的大画面或小画面。	支持
setPriorRemoteVideoStreamType	废弃接口：设定观看方优先选择的视频质量。	不支持
snapshotVideo	视频画面截图。	支持

## 7.7. 音频相关接口函数

API	描述	是否支持
startLocalAudio	开启本地音频的采集和上行。	支持
stopLocalAudio	关闭本地音频的采集和上行。	支持
muteLocalAudio	静音本地的音频。	支持
muteRemoteAudio	静音某一个用户的声音，同时不再拉取该远端用户的音频数据流。	支持
muteAllRemoteAudio	静音所有用户的声音，同时不再拉取远端用户的音频数据流。	支持
setAudioCaptureVolume	设置 SDK 采集音量。	不支持
getAudioCaptureVolume	获取 SDK 采集音量。	不支持
setAudioPlayoutVolume	设置 SDK 播放音量。	不支持
getAudioPlayoutVolume	获取 SDK 播放音量。	不支持
enableAudioVolumeEvaluation	启用或关闭音量大小提示。	不支持
startAudioRecording	开始录音。	待开发
stopAudioRecording	停止录音。	待开发
setAudioQuality	废弃接口：设置音频质量。	不支持
setRemoteAudioVolume	设置远程用户播放音量。	不支持

## 7.8. 摄像头相关接口函数

API	描述	是否支持
getCameraDevicesList	获取摄像头设备列表。	支持
setCurrentCameraDevice	设置要使用的摄像头。	支持
getCurrentCameraDevice	获取当前使用的摄像头。	支持

## 7.9. 音频设备相关接口函数

API	描述	是否支持
getMicDevicesList	获取麦克风设备列表。	支持
getCurrentMicDevice	获取当前选择的麦克风。	支持
setCurrentMicDevice	设置要使用的麦克风。	支持
getCurrentMicDeviceVolume	获取系统当前麦克风设备音量。	待开发
setCurrentMicDeviceVolume	设置系统当前麦克风设备的音量。	待开发
setCurrentMicDeviceMute	设置系统当前麦克风设备的静音状态。	支持
getCurrentMicDeviceMute	获取系统当前麦克风设备是否静音。	支持
getSpeakerDevicesList	获取扬声器设备列表。	支持
getCurrentSpeakerDevice	获取当前的扬声器设备。	支持
setCurrentSpeakerDevice	设置要使用的扬声器。	支持
getCurrentSpeakerVolume	获取系统当前扬声器设备音量。	待开发
setCurrentSpeakerVolume	设置系统当前扬声器设备音量。	待开发
setCurrentSpeakerDeviceMute	设置系统当前扬声器设备的静音状态。控制 audio 标签	支持
getCurrentSpeakerDeviceMute	获取系统当前扬声器设备是否静音。控制 audio 标签	支持

## 7.10. 美颜相关接口函数

API	描述	是否支持
setBeautyStyle	设置美颜、美白以红润效果级别。	不支持
setWaterMark	设置水印。	待开发

## 7.11. 辅流相关接口函数

API	描述	是否支持
startRemoteSubStreamView	废弃接口：开始渲染远端用户的辅流（屏幕分享）画面。	支持
stopRemoteSubStreamView	废弃接口：停止渲染远端用户的辅流（屏幕分享）画面。	支持
setRemoteSubStreamViewFillMode	废弃接口：设置辅流（屏幕分享）画面的渲染模式。	支持
setRemoteSubStreamViewRotation	废弃接口：设置辅流（屏幕分享）画面的顺时针旋转角度。	支持
getScreenCaptureSources	枚举可共享的窗口列表。（Electron 端支持）	支持
selectScreenCaptureTarget	设置屏幕共享参数，该方法在屏幕共享过程中也可以调用。	支持
startScreenCapture	启动屏幕分享。	支持
pauseScreenCapture	暂停屏幕分享。	待开发
resumeScreenCapture	恢复屏幕分享。	待开发
stopScreenCapture	停止屏幕分享。	支持
setSubStreamEncoderParam	设置辅流（屏幕分享）的编码器参数。	不支持
setSubStreamMixVolume	设置辅流（屏幕分享）的混音音量大小。	不支持
addExcludedShareWindow	将指定窗口加入屏幕分享的排除列表中，加入排除列表中的窗口不会被分享出去。	待开发
removeExcludedShareWindow	将指定窗口从屏幕分享的排除列表中移除。	待开发
removeAllExcludedShareWindow	将所有窗口从屏幕分享的排除列表中移除。	待开发

## 7.12. 自定义消息发送

API	描述	是否支持
sendCustomCmdMsg	发送自定义消息给房间内所有用户。	待开发
sendSEIMsg	将小数据量的自定义数据嵌入视频帧中。	待开发

## 7.13. 背景混音相关接口函数

API	描述	是否支持
playBGM	废弃接口：启动播放背景音乐。	不支持
stopBGM	废弃接口：停止播放背景音乐。	不支持
pauseBGM	废弃接口：暂停播放背景音乐。	不支持
resumeBGM	废弃接口：继续播放背景音乐。	不支持
getBGMDuration	废弃接口：获取背景音乐文件总时长，单位毫秒。	不支持
setBGMPosition	废弃接口：设置背景音乐播放进度。	不支持
setBGMVolume	废弃接口：设置背景音乐播放音量的大小。	不支持
setBGMLayoutVolume	废弃接口：设置背景音乐本地播放音量的大小。	不支持
setBGMPublishVolume	废弃接口：设置背景音乐远端播放音量的大小。	不支持
startSystemAudioLoopback	打开系统声音采集。	待开发
stopSystemAudioLoopback	关闭系统声音采集。	待开发
setSystemAudioLoopbackVolume	设置系统声音采集的音量。	待开发
startPlayMusic	启动播放背景音乐。	待开发
stopPlayMusic	停止播放背景音乐。	待开发
pausePlayMusic	暂停播放背景音乐。	待开发
resumePlayMusic	恢复播放背景音乐。	待开发
getMusicDurationInMS	获取背景音乐文件总时长，单位毫秒。	待开发
seekMusicToPosInTime	设置背景音乐播放进度。	待开发
setAllMusicVolume	设置背景音乐的音量大小，播放背景音乐混音时使用，用来控制背景音音量大小。	待开发
setMusicLayoutVolume	设置背景音乐本地播放音量的大小。	待开发
setMusicPublishVolume	设置背景音乐远端播放音量的大小。	待开发

## 7.14. 音效相关接口函数

API	描述	是否支持

playAudioEffect	废弃接口：播放音效。	不支持
setAudioEffectVolume	废弃接口：设置音效音量。	不支持
stopAudioEffect	废弃接口：停止音效。	不支持
stopAllAudioEffects	废弃接口：停止所有音效。	不支持
setAllAudioEffectsVolume	废弃接口：设置所有音效的音量。	不支持
pauseAudioEffect	废弃接口：暂停音效。	不支持
resumeAudioEffect	废弃接口：恢复音效。	不支持

## 7.15. 设备和网络测试

API	描述	是否支持
startSpeedTest	开始进行网络测速（视频通话期间请勿测试，以免影响通话质量）。	待开发
stopSpeedTest	停止网络测速。	待开发
startCameraDeviceTest	开始进行摄像头测试。	待开发
stopCameraDeviceTest	停止摄像头测试。	待开发
startMicDeviceTest	开始进行麦克风测试。	待开发
stopMicDeviceTest	停止麦克风测试。	待开发
startSpeakerDeviceTest	开始进行扬声器测试。	待开发
stopSpeakerDeviceTest	停止扬声器测试。	待开发

## 7.16. LOG 相关接口函数

API	描述	是否支持
getSDKVersion	获取 SDK 版本信息。	支持
setLogLevel	设置 Log 输出级别。	支持
setConsoleEnabled	启用或禁用控制台日志打印。	支持
setLogCompressEnabled	启用或禁用 Log 的本地压缩。	待开发
setLogDirPath	设置日志保存路径。	待开发
setLogCallback	设置日志回调。	待开发
callExperimentalAPI	调用实验性 API 接口。	待开发

## 7.17. 弃用接口函数

API	描述
setMicVolumeOnMixing	从 v6.9 版本开始废弃。

# 8. 视频通话回调接口支持情况

## 8.1. 错误事件和警告事件

API	描述	是否支持
onError	错误回调：SDK 不可恢复的错误，必须监听，并分情况给用户适当的界面提示。	支持
onWarning	警告回调：用于告知您一些非严重性问题，例如出现了卡顿或可恢复的解码失败。	支持

## 8.2. 房间事件回调

API	描述	是否支持
onEnterRoom	已加入房间的回调。	支持
onExitRoom	退出房间的事件回调。	支持
onSwitchRole	切换角色的事件回调。	待开发
onConnectOtherRoom	请求跨房连麦（主播跨房 PK）的结果回调。	待开发
onDisconnectOtherRoom	关闭跨房连麦（主播跨房 PK）的结果回调。	待开发
onSwitchRoom	切换房间。	待开发

## 8.3. 成员事件回调

API	描述	是否支持
onRemoteUserEnterRoom	有用户加入当前房间。	支持
onRemoteUserLeaveRoom	有用户离开当前房间。	支持
onUserVideoAvailable	用户是否开启摄像头视频。	支持
onUserSubStreamAvailable	用户是否开启屏幕分享。	支持
onUserAudioAvailable	用户是否开启音频上行。	支持

onFirstVideoFrame	开始渲染本地或远程用户的首帧画面。	支持
onFirstAudioFrame	开始播放远程用户的首帧音频（本地声音暂不通知）。	待开发
onSendFirstLocalVideoFrame	首帧本地视频数据已经被送出。	待开发
onSendFirstLocalAudioFrame	首帧本地音频数据已经被送出。	待开发
onUserEnter	废弃接口：有主播加入当前房间。	不支持
onUserExit	废弃接口：有主播离开当前房间。	不支持

## 8.4. 统计和质量回调

API	描述	是否支持
onNetworkQuality	网络质量：该回调每 2 秒触发一次，统计当前网络的上行和下行质量。	待开发
onStatistics	技术指标统计回调。	待开发

## 8.5. 服务器事件回调

API	描述	是否支持
onConnectionLost	SDK 与服务器的连接断开。	待开发
onTryToReconnect	SDK 尝试重新连接到服务器。	待开发
onConnectionRecovery	SDK 与服务器的连接恢复。	待开发
onSpeedTest	废弃接口：服务器测速的回调，SDK 对多个服务器 IP 进行测速，每个 IP 的测速结果通过这个回调通知。	不支持
onSpeedTestResult	网速测试的结果回调。	待开发

## 8.6. 硬件设备事件回调

API	描述	是否支持
onCameraDidReady	摄像头准备就绪。	待开发
onMicDidReady	麦克风准备就绪。	待开发
onUserVoiceVolume	用于提示音量大小的回调，包括每个 userId 的音量和远端总音量；本地用户 userid 为"。	待开发
onDeviceChange	本地设备通断回调。	待开发
onTestMicVolume	麦克风测试音量回调。	待开发
onTestSpeakerVolume	扬声器测试音量回调。	待开发

onAudioDeviceCaptureVolumeChanged	当前音频采集设备音量变化回调。	待开发
onAudioDevicePlayoutVolumeChanged	当前音频播放设备音量变化回调。	待开发

## 8.7. 自定义消息的接回调

API	描述	是否支持
onRecvCustomCmdMsg	收到自定义消息回调。	待开发
onMissCustomCmdMsg	自定义消息丢失回调。	待开发
onRecvSEIMsg	收到 SEI 消息的回调。	待开发

## 8.8. CDN 旁路转推回调

API	描述	是否支持
onStartPublishing	开始向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 startPublishing() 接口。	待开发
onStopPublishing	停止向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 stopPublishing() 接口。	待开发
onStartPublishCDNStream	启动旁路推流到 CDN 完成的回调。	待开发
onStopPublishCDNStream	停止旁路推流到 CDN 完成的回调。	待开发
onSetMixTranscodingConfig	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 setMixTranscodingConfig() 接口。	待开发

## 8.9. 系统音量采集回调

API	描述	是否支持
onSystemAudioLoopbackError	系统音量采集状态的回调（仅在 Mac 上有效）。	待开发

## 8.10. 音效回调

API	描述	是否支持
onAudioEffectFinished	废弃接口：播放音效结束回调。	不支持

## 8.11. 屏幕分享回调

API	描述	是否支持
onScreenCaptureCovered	当屏幕分享窗口被遮挡无法正常捕获时，SDK 会通过此回调通知，可在此回调里通知用户移开遮挡窗口。	待开发

onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知。	待开发
onScreenCapturePaused	当屏幕分享暂停时，SDK 会通过此回调通知。	待开发
onScreenCaptureResumed	当屏幕分享恢复时，SDK 会通过此回调通知。	待开发
onScreenCaptureStopped	当屏幕分享停止时，SDK 会通过此回调通知。	待开发

## 8.12. 截图回调

API	描述	是否支持
onSnapshotComplete	截图完成时，SDK 会通过此回调通知。	支持

## 8.13. 背景混音事件回调

API	描述	是否支持
onPlayBGMBegin	废弃接口：开始播放背景音乐。	不支持
onPlayBGMPProgress	废弃接口：播放背景音乐的进度。	不支持
onPlayBGMComplete	废弃接口：播放背景音乐结束。	不支持

# 9. 服务端 API 接口支持情况

## 9.1. 房间管理相关接口

API	描述	是否支持
RemoveUser	移出用户（封装到 TRTCCould.js 中）	支持
DismissRoom	解散房间（封装到 TRTCCould.js 中）	支持
RemoveUserByStrRoomId	移出用户（字符串房间号）	待开发
DismissRoomByStrRoomId	解散房间（字符串房间号）	待开发

## 9.2. 混流转推相关接口

API	描述	是否支持
StartPublishCdnStream	启动转推任务	待开发
UpdatePublishCdnStream	更新转推任务	待开发
StopPublishCdnStream	停止转推任务	待开发
StartMCUMixTranscode	启动云端混流（旧）	待开发

况

StopMCUMixTranscode	结束云端混流（旧）	待开发
StartMCUMixTranscodeByStrRoomId	启动云端混流（字符串房间号）	待开发
StopMCUMixTranscodeByStrRoomId	结束云端混流（字符串房间号）	待开发

### 9.3. 云端录制相关接口

API	描述	是否支持
CreateCloudRecording	开始云端录制	待开发
DeleteCloudRecording	停止云端录制任务	待开发
DescribeCloudRecording	查询云端录制状态	待开发
ModifyCloudRecording	更新云端录制任务	待开发

### 9.4. 用量统计相关接口

API	描述	是否支持
DescribeMixTranscodingUsage	查询 TRTC 混流转码用量	待开发
DescribeRecordingUsage	查询 TRTC 录制用量	待开发
DescribeTrtcUsage	查询 TRTC 音视频用量	待开发
DescribeRecordStatistic	查询云端录制计费时长（旧）	待开发
DescribeRelayUsage	查询 TRTC 旁路转推用量	待开发

### 9.5. 监控仪表查询相关接口

API	描述	是否支持
DescribeTRTCRealTimeScaleMetricData	查询 TRTC 实时监控规模指标	待开发
DescribeTRTCRealTimeQualityMetricData	查询 TRTC 实时监控质量指标	待开发
DescribeTRTCMarketScaleMetricData	查询 TRTC 数据大盘规模指标	待开发
DescribeTRTCMarketQualityMetricData	查询 TRTC 数据大盘质量指标	待开发

### 9.6. 图片相关接口

API	描述	是否支持
CreatePicture	上传图片	待开发

DeletePicture	删除图片	待开发
DescribePicture	查询图片	待开发
ModifyPicture	修改图片	待开发

## 10. 关键类型定义

### 10.1. 关键类型

类名	描述
TRTCParams	进房相关参数。
TRTCVideoEncParam	视频编码参数。
TRTCNetworkQosParam	网络流控相关参数。
TRTCQualityInfo	视频质量。
TRTCVolumeInfo	音量大小。
TRTCSpeedTestResult	网络测速结果。
TRTCMixUser	云端混流中每一路子画面的位置信息。
TRTCTranscodingConfig	云端混流（转码）配置。
TRTCPublishCDNParam	CDN 旁路推流参数。
TRTCAudioRecordingParams	录音参数。
TRTCLocalStatistics	自己本地的音视频统计信息。
TRTCRemoteStatistics	远端成员的音视频统计信息。
TRTCStatistics	统计数据。

### 10.2. 枚举值

枚举	描述
TRTCVideoResolution	视频分辨率。
TRTCVideoResolutionMode	视频分辨率模式。
TRTCVideoStreamType	视频流类型。
TRTCQuality	画质级别。
TRTCVideoFillMode	视频画面填充模式。
TRTCBeautyStyle	美颜（磨皮）算法。
TRTCAppScene	应用场景。

TRTCRoleType	角色，仅适用于直播场景 (TRTCAppSceneLIVE)。
TRTCQosControlMode	流控模式。
TRTCVideoQosPreference	画质偏好。
TRTCDeviceState	设备操作。
TRTCDeviceType	设备类型。
TRTCWaterMarkSrcType	水印图片的源类型。

## 11. 错误码

### 11.1. 错误码表

#### 11.1.1. 基础错误码

符号	值	含义
ERR_NULL	0	无错误

#### 11.1.2. 进房相关错误码

TRTCCloud.enterRoom() 在进房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.onEnterRoom() 和 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_ROOM_ENTER_FAIL	-3301	进入房间失败
ERR_ENTER_ROOM_PARAM_NULL	-3316	进房参数为空，请检查 TRTCCloud.enterRoom(): 接口调用是否传入有效的 param
ERR_SDK_APPID_INVALID	-3317	进房参数 sdkAppId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_USER_ID_INVALID	-3319	进房参数 userID 不正确
ERR_USER_SIG_INVALID	-3320	进房参数 userSig 不正确
ERR_ROOM_REQUEST_ENTER_ROOM_TIM EOUT	-3308	请求进房超时，请检查网络
ERR_SERVER_INFO_PRIVILEGE_FLAG_ER ROR	-100006	权限位校验失败，请检查 privateMapKey 是否正确
ERR_SERVER_INFO_SERVICE_SUSPENDE	-100013	服务不可用。请检查：套餐包剩余分钟数是否

D		大于 0，腾讯云账号是否欠费
ERR_SERVER_INFO_ECDH_GET_TINYID	-100018	userSig 校验失败，请检查 userSig 是否正确

### 11.1.3. 退房相关错误码

TRTCCloud.exitRoom() 在退房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_QUIT_ROOM_TIME_OUT	-3325	请求退房超时

### 11.1.4. 设备（摄像头、麦克风、扬声器）相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOTAUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOTAUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac

		设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败

### 11.1.5. 屏幕分享相关错误码

您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要 5.0 以上的系统，在 iOS 平台，需要 11.0 以上的系统
ERR_SERVER_CENTER_NO_PRIVILEGE_PUSH_SUB_VIDEO	-102015	没有权限上行辅路
ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO	-102016	其他用户正在上行辅路
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止

### 11.1.6. 编解码相关错误码

您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出
ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率

### 11.1.7. 自定义采集相关错误码

您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
<code>ERR_PIXEL_FORMAT_UNSUPPORTED</code>	-1327	设置的 pixel format 不支持
<code>ERR_BUFFER_TYPE_UNSUPPORTED</code>	-1328	设置的 buffer type 不支持

### 11.1.8. CDN 绑定和混流相关错误码

您可以通过回调函数 `TRTCCloudDelegate.onStartPublishing()` 和 `TRTCCloudDelegate.onSetMixTranscodingConfig()` 捕获相关通知。

符号	值	含义
<code>ERR_PUBLISH_CDN_STREAM_REQUEST_TIMEOUT_OUT</code>	-3321	旁路转推请求超时
<code>ERR_CLOUD_MIX_TRANSCODING_REQUEST_TIMEOUT_OUT</code>	-3322	云端混流请求超时
<code>ERR_PUBLISH_CDN_STREAM_SERVER_FAILED</code>	-3323	旁路转推回包异常
<code>ERR_ROOM_REQUEST_STARTUBLISHING_TIMEOUT</code>	-3333	开始向腾讯云的直播 CDN 推流信令超时
<code>ERR_ROOM_REQUEST_STARTUBLISHING_ERROR</code>	-3334	开始向腾讯云的直播 CDN 推流信令异常
<code>ERR_ROOM_REQUEST_STOPUBLISHING_TIMEOUT</code>	-3335	停止向腾讯云的直播 CDN 推流信令超时
<code>ERR_ROOM_REQUEST_STOPUBLISHING_ERROR</code>	-3336	停止向腾讯云的直播 CDN 推流信令异常

### 11.1.9. 跨房连麦相关错误码

`TRTCCloud.ConnectOtherRoom()` 在跨房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.onConnectOtherRoom()` 捕获相关通知。

符号	值	含义
<code>ERR_ROOM_REQUEST_CONN_ROOM_TIMEOUT</code>	-3326	请求连麦超时
<code>ERR_ROOM_REQUEST_DISCONNECT_ROOM_</code>	-3327	请求退出连麦超时

TIMEOUT		
ERR_ROOM_REQUEST_CONN_ROOM_INV_ALID_PARAM	-3328	无效参数
ERR_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 switchRole() 到主播
ERR_SERVER_CENTER_CONN_ROOM_NOT_SUPPORT	-102031	不支持跨房间连麦
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_NUM	-102032	达到跨房间连麦上限
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_RETRY_TIMES	-102033	跨房间连麦重试次数耗尽
ERR_SERVER_CENTER_CONN_ROOM_REQ_TIMEOUT	-102034	跨房间连麦请求超时
ERR_SERVER_CENTER_CONN_ROOM_REQ	-102035	跨房间连麦请求格式错误
ERR_SERVER_CENTER_CONN_ROOM_NO_SIG	-102036	跨房间连麦无签名
ERR_SERVER_CENTER_CONN_ROOM_DECRYPT_SIG	-102037	跨房间连麦签名解密失败
ERR_SERVER_CENTER_CONN_ROOM_NO_KEY	-102038	未找到跨房间连麦签名解密密钥
ERR_SERVER_CENTER_CONN_ROOM_PARSE_SIG	-102039	跨房间连麦签名解析错误
ERR_SERVER_CENTER_CONN_ROOM_INV_ALID_SIG_TIME	-102040	跨房间连麦签名时间戳错误
ERR_SERVER_CENTER_CONN_ROOM_SIG_GROUPID	-102041	跨房间连麦签名内房间号不匹配
ERR_SERVER_CENTER_CONN_ROOM_NOT_CONNECTED	-102042	跨房间连麦签名内用户名不匹配
ERR_SERVER_CENTER_CONN_ROOM_USE_R_NOT_CONNECTED	-102043	本用户未发起连麦
ERR_SERVER_CENTER_CONN_ROOM_FAILED	-102044	跨房间连麦失败

ERR_SERVER_CENTER_CONN_ROOM_CAN CEL_FAILED	-102045	取消跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CO NNED_ROOM_NOT_EXIST	-102046	被连麦房间不存在
ERR_SERVER_CENTER_CONN_ROOM_CO NNED_REACH_MAX_ROOM	-102047	被连麦房间达到连麦上限
ERR_SERVER_CENTER_CONN_ROOM_CO NNED_USER_NOT_EXIST	-102048	被连麦用户不存在
ERR_SERVER_CENTER_CONN_ROOM_CO NNED_USER_DELETED	-102049	被连麦用户已被删除
ERR_SERVER_CENTER_CONN_ROOM_CO NNED_USER_FULL	-102050	被连麦用户达到资源上限
ERR_SERVER_CENTER_CONN_ROOM_INV ALID_SEQ	-102051	连麦请求序号错乱

## 11.2. 警告码表

警告码无需特别关注，您可以按需选择是否对当前用户进行相应的提示。

符号	值	含义
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风

ED		
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用）
WARNING_VIDEO_FRAME_DECODE_FAIL	2101	当前视频帧解码失败
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_VIDEO_PLAY_LAG	2105	当前视频播放出现卡顿
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_START_CAPTURE_IGNORED	4000	已经在采集，启动采集被忽略
WARNING_AUDIO_RECORDING_WRITE FAIL	7001	音频录制写入文件失败
WARNING_ROOM_DISCONNECT	5101	网络断开连接
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，忽略上行音视频数据
WARNING_NET_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻
WARNING_RTMP_SERVER_RECONNECT	1102	直播推流，网络断连，已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_LIVE_STREAM_SERVER_RECONNECT	2103	直播拉流，网络断连，已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_RECV_DATA_LAG	2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
WARNING_RTMP_DNS_FAIL	3001	直播，DNS 解析失败
WARNING_RTMP_SEVER_CONN_FAIL	3002	直播，服务器连接失败
WARNING_RTMP_SHAKE_FAIL	3003	直播，与 RTMP 服务器握手失败
WARNING_RTMP_SERVER_BREAK_CONNECTION	3004	直播，服务器主动断开
WARNING_RTMP_READ_WRITE_FAIL	3005	直播，RTMP 读/写失败，将会断开连接

WARNING_RTMP_WRITE_FAIL	3006	直播, RTMP 写失败 (SDK 内部错误码, 不会对外抛出)
WARNING_RTMP_READ_FAIL	3007	直播, RTMP 读失败 (SDK 内部错误码, 不会对外抛出)
WARNING_RTMP_NO_DATA	3008	直播, 超过 30s 没有数据发送, 主动断开连接
WARNING_PLAY_LIVE_STREAM_INFO_CONNECT_FAIL	3009	直播, connect 服务器调用失败 (SDK 内部错误码, 不会对外抛出)
WARNING_NO_STEAM_SOURCE_FAIL	3010	直播, 连接失败, 该流地址无视频 (SDK 内部错误码, 不会对外抛出)
WARNING_ROOM_RECONNECT	5102	网络断连, 已启动自动重连
WARNING_ROOM_NET_BUSY	5103	网络状况不佳: 上行带宽太小, 上传数据受阻

## 12. 接口详情

### 12.1. 创建函数 (new TRTCCloud())

#### 12.1.1. 示例

```

1. // 创建/使用/销毁 TRTCCloud 对象的示例代码:
2. import TRTCCloud from 'trtc-electron-sdk';
3. this rtcCloud = new TRTCCloud();
4. // 获取 SDK 版本号
5. let version = this rtcCloud.getSDKVersion();

```

#### 12.1.2. 方法

(1) (static) getTRTCShareInstance() → {TRTCCloud}

创建 TRTCCloud 对象单例

Returns:

Type **TRTCCloud**

(2) (static) destroyTRTCShareInstance()

释放 TRTCCloud 对象并清理资源

## (3) getConfigObject() → {TRTCCConfig}

获取 TRTC 配置对象

可以通过 TRTC 配置对象 TRTCCConfig 打开 debug 模式

示例：

```
1. // Enable 'debug' mode
2. const TRTCCloud = require('trtc-electron-sdk');
3. const rtcCloud = new TRTCCloud();
4. rtcCloud.getConfigObject().setDebugMode(true);
```

Returns:

Type **TRTCCConfig**

## (4) destroy()

清理资源

## 12.2. 房间

### 12.2.1. enterRoom

概述：进入房间。

调用接口后，您会收到来自 TRTCCallback 中的 onEnterRoom(result) 回调：

- 如果加入成功，result 会是一个正数 (result > 0)，表示加入房间的时间消耗，单位是毫秒 (ms)。
- 如果加入失败，result 会是一个负数 (result < 0)，表示进房失败的错误码。

进房失败的错误码含义请参见错误码。

参数 scene 的枚举值如下：

枚举	描述
TRTCAppSceneVideoCall	视频通话场景，支持 720P、1080P 高清画质，单个房间最多支持 300 人同时在线，最高支持 50 人同时发言。 适合：[1 对 1 视频通话]、[300 人视频会议]、[在线问诊]、[视频聊天]、[远程面试]等。
TRTCAppSceneAudioCall	语音通话场景，支持 48kHz，支持双声道。单个房间最多支持 300 人同时在线，最高支持 50 人同时发言。 适合：[1 对 1 语音通话]、[300 人语音会议]、[语音聊天]、[语音会议]、[在线狼人杀]等。
TRTCAppSceneLIVE	视频互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于 300ms；支持十万级别观众同时播放，播放延时低至 1000ms。 适合：[视频低延时直播]、[十万人互动课堂]、[视频直播 PK]、[视频相亲房]、[互动课堂]、[远程培训]、[超大型会议]等。
TRTCAppSceneVoiceChatRoo	语音互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于 300ms；

m	支持十万级别观众同时播放，播放延时低至 1000ms。 适合：[语音低延时直播]、[语音直播连麦]、[语聊房]、[K 歌房]、[FM 电台]等。
---	---

注意：

- 当 scene 选择为 TRTCAppSceneLIVE 或 TRTCAppSceneVoiceChatRoom 时，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。
- 不管进房是否成功，enterRoom 都必须与 exitRoom 配对使用，在调用 exitRoom 前再次调用 enterRoom 函数会导致不可预期的错误问题。

参数：

Name	Type	Description																																	
params	TRTCParams	进房参数																																	
		Properties																																	
		<table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>sdkAppId</td><td>Number</td><td>应用标识（必填）</td></tr> <tr> <td>userId</td><td>String</td><td>用户标识（必填）</td></tr> <tr> <td>userSig</td><td>String</td><td>用户签名（必填）</td></tr> <tr> <td>roomId</td><td>Number</td><td>房间号码, roomId 和 strRoomId 必须填一个, 若您选用 strRoomId, 则 roomId 需要填写为 0。</td></tr> <tr> <td>strRoomId</td><td>String</td><td>字符串房间号码 [选填], 在同一个房间内的用户可以看到彼此并进行视频通话, roomId 和 strRoomId 必须填一个。若两者都填, 则优先选择 roomId。</td></tr> <tr> <td>role</td><td>TRTCRoleType</td><td>直播场景下的角色, 默认值: 主播 TRTCRoleAnchor: 主播, 可以上行视频和音频, 一个房间里最多支持 50 个主播同时上行音视频。 TRTCRoleAudience: 观众, 只能观看, 不能上行视频和音频, 一个房间里的观众人数没有上限。</td></tr> <tr> <td>privateMapKey</td><td>String</td><td>房间签名（非必填）</td></tr> <tr> <td>businessInfo</td><td>String</td><td>业务数据（非必填）</td></tr> <tr> <td>streamId</td><td>String</td><td>自定义 CDN 播放地址（非必填）</td></tr> <tr> <td>userDefineRecordId</td><td>String</td><td>设置云端录制完成后的回调消息中的 "userdefinerecordid" 字段内容, 便于您更方便的识别录制回调。</td></tr> </tbody> </table>	Name	Type	Description	sdkAppId	Number	应用标识（必填）	userId	String	用户标识（必填）	userSig	String	用户签名（必填）	roomId	Number	房间号码, roomId 和 strRoomId 必须填一个, 若您选用 strRoomId, 则 roomId 需要填写为 0。	strRoomId	String	字符串房间号码 [选填], 在同一个房间内的用户可以看到彼此并进行视频通话, roomId 和 strRoomId 必须填一个。若两者都填, 则优先选择 roomId。	role	TRTCRoleType	直播场景下的角色, 默认值: 主播 TRTCRoleAnchor: 主播, 可以上行视频和音频, 一个房间里最多支持 50 个主播同时上行音视频。 TRTCRoleAudience: 观众, 只能观看, 不能上行视频和音频, 一个房间里的观众人数没有上限。	privateMapKey	String	房间签名（非必填）	businessInfo	String	业务数据（非必填）	streamId	String	自定义 CDN 播放地址（非必填）	userDefineRecordId	String	设置云端录制完成后的回调消息中的 "userdefinerecordid" 字段内容, 便于您更方便的识别录制回调。
Name	Type	Description																																	
sdkAppId	Number	应用标识（必填）																																	
userId	String	用户标识（必填）																																	
userSig	String	用户签名（必填）																																	
roomId	Number	房间号码, roomId 和 strRoomId 必须填一个, 若您选用 strRoomId, 则 roomId 需要填写为 0。																																	
strRoomId	String	字符串房间号码 [选填], 在同一个房间内的用户可以看到彼此并进行视频通话, roomId 和 strRoomId 必须填一个。若两者都填, 则优先选择 roomId。																																	
role	TRTCRoleType	直播场景下的角色, 默认值: 主播 TRTCRoleAnchor: 主播, 可以上行视频和音频, 一个房间里最多支持 50 个主播同时上行音视频。 TRTCRoleAudience: 观众, 只能观看, 不能上行视频和音频, 一个房间里的观众人数没有上限。																																	
privateMapKey	String	房间签名（非必填）																																	
businessInfo	String	业务数据（非必填）																																	
streamId	String	自定义 CDN 播放地址（非必填）																																	
userDefineRecordId	String	设置云端录制完成后的回调消息中的 "userdefinerecordid" 字段内容, 便于您更方便的识别录制回调。																																	

scene	TRTCAppScene	应用场景，目前支持视频通话（VideoCall）、在线直播（Live）、语音通话（AudioCall）、语音聊天室（VoiceChatRoom）四种场景。
-------	--------------	---

### 12.2.2. exitRoom()

概述：退出房间

调用 `exitRoom()` 接口会执行退出房间的相关逻辑，例如释放音视频设备资源和编解码器资源。待资源释放完毕，SDK 会通过 `TRTCCallback` 中的 `onExitRoom()` 回调通知您。

如果您要再次调用 `enterRoom()` 或者切换到其它的音视频 SDK，请等待 `onExitRoom()` 回调到来后再执行相关操作，否则可能会遇到如摄像头、麦克风设备被强占等各种异常问题。

### 12.2.3. switchRoom(params)

概述：切换房间

调用该接口后，用户会先退出原来的房间并快速进入 `TRTCSwitchRoomParam` 中指定的新房间：相比于直接调用 `exitRoom + enterRoom` 的方式，`switchRoom` 接口对主播更加友好，因为 `switchRoom` 不会停止主播端视频的采集和预览。

参数：

Name	Type	Description
params	TRTCSwitchRoomParam	房间切换参数，请参考 <code>TRTCSwitchRoomParam</code> 接口调用结果会通过 <code>onSwitchRoom(errCode, errMsg)</code> 事件回调通知给您。

### 12.2.4. switchRole(role)

概述：切换角色，仅适用于直播场景（`TRTCAppSceneLIVE` 和 `TRTCAppSceneVoiceChatRoom`）

在直播场景下，一个用户可能需要在“观众”和“主播”之间来回切换。您可以在进房前通过 `TRTCPParams` 中的 `role` 字段确定角色，也可以通过 `switchRole` 在进房后切换角色。

参数：

Name	Type	Description
role	TRTCRoleType	目标角色，默认为主播 <code>TRTCRoleAnchor</code> : 主播，可以上行视频和音频，一个房间里最多支持 50 个主播同时上行音视频。 <code>TRTCRoleAudience</code> : 观众，只能观看，不能上行视频和音频，一个房间里观众人数没有上限。

### 12.2.5. connectOtherRoom(params)

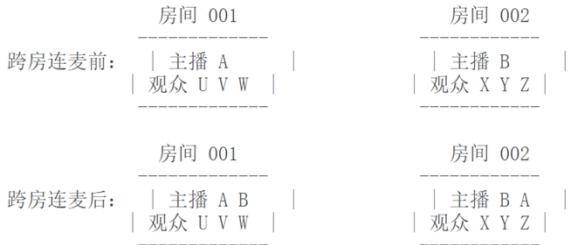
概述：请求跨房连麦（主播跨房 PK）

TRTC 中两个不同音视频房间中的主播，可以通过“跨房连麦”功能拉通连麦通话功能。使用此功能时，两个主播无需退出各自原来的直播间即可进行“连麦 PK”。

例如：当房间“001”中的主播 A 通过 `connectOtherRoom()` 跟房间“002”中的主播 B 拉通跨房连麦后，

房间“001”中的用户都会收到主播 B 的 onUserEnter(B) 回调和 onUserVideoAvailable(B,true) 回调。房间“002”中的用户都会收到主播 A 的 onUserEnter(A) 回调和 onUserVideoAvailable(A,true) 回调。

简言之，跨房连麦的本质，就是把两个不同房间中的主播相互分享，让每个房间里的观众都能看到两个主播。



考虑到后续扩展字段的兼容性问题，跨房连麦的参数暂时采用了 JSON 格式的字符串，要求至少包含两个字段：

- roomId：房间“001”中的主播 A 要跟房间“002”中的主播 B 连麦，主播 A 调用 connectOtherRoom() 时 roomId 应指定为“002”。
- userId：房间“001”中的主播 A 要跟房间“002”中的主播 B 连麦，主播 A 调用 connectOtherRoom() 时 userId 应指定为 B 的 userId。

跨房连麦的请求结果会通过 TRTCCallback 中的 onConnectOtherRoom 回调通知给您。

示例：

1. let json = JSON.stringify({roomId: 2, userId: "userB"});
2. rtcCloud.connectOtherRoom(json);

参数：

Name	Type	Description
params	String	JSON 字符串连麦参数，roomId 代表目标房间号，userId 代表目标用户 ID。

### 12.2.6. disconnectOtherRoom()

概述：关闭跨房连麦（主播跨房 PK）

跨房连麦的退出结果会通过 TRTCCallback 中的 onDisconnectOtherRoom 回调通知给您。

### 12.2.7. setDefaultStreamRecvMode(autoRecvAudio, autoRecvVideo)

概述：设置音视频数据接收模式（需要在进房前设置才能生效）

为实现进房秒开的绝佳体验，SDK 默认进房后自动接收音视频。即在您进房成功的同时，您将立刻收到远端所有用户的音视频数据。若您没有调用 startRemoteView，视频数据将自动超时取消。若您主要用于语音聊天等没有自动接收视频数据需求的场景，您可以根据实际需求选择接收模式。

注意：需要在进房前设置才能生效。

参数：

Name	Type	Description
autoRecvAudio	Boolean	true：自动接收音频数据；false：需要调用 muteRemoteAudio 进行请求或取消。默认值：true
autoRecvVideo	Boolean	true : 自动接收视频数据； false : 需要调用 startRemoteView/stopRemoteView 进行请求或取消。默认值：true

## 12.3. 混流 CDN

### 12.3.1. Start-publishing(streamId, type)

概述：开始向腾讯云的直播 CDN 推流

该接口会指定当前用户的音视频流在腾讯云 CDN 所对应的 StreamId，进而可以指定当前用户的 CDN 播放地址。

例如：如果我们采用如下代码设置当前用户的主画面 StreamId 为 user\_stream\_001，那么该用户主画面对应的 CDN 播放地址为：“[http://yourdomain/live/user\\_stream\\_001.flv](http://yourdomain/live/user_stream_001.flv)”，其中 yourdomain 为您自己备案的播放域名，您可以在直播控制台 配置您的播放域名，腾讯云不提供默认的播放域名。

您也可以在设置 enterRoom 的参数 TRTCParams 时指定 streamId，而且我们更推荐您采用这种方案。

注意：您需要先在实时音视频 控制台 中的功能配置页开启“启动自动旁路直播”才能生效。

示例：

1. let trtcCloud = TRTCCloud.getTRTCShareInstance();
2. trtcCloud.enterRoom(params, TRTCAppScene.TRTCAppSceneLIVE);
3. trtcCloud.startLocalPreview(view);
4. trtcCloud.startLocalAudio(TRTCAudioQuality.TRTCAudioQualityDefault);
5. trtcCloud.startPublishing("user\_stream\_001", TRTCVideoStreamType.TRTCVideoStreamTypeBig);

参数：

Name	Type	Description
streamId	String	自定义流 ID。
type	TRTCVideoStream Type	仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub。

### 12.3.2. stopPublishing()

概述：停止向腾讯云的直播 CDN 推流。

### 12.3.3. startPublishCDNStream(param)

概述：开始向非腾讯云的直播 CDN 转推

该接口跟 startPublishing() 类似，但 startPublishCDNStream() 支持向非腾讯云的直播 CDN 转推。

注意：

- 使用 startPublishing() 绑定腾讯云直播 CDN 不收取额外的费用。
- 使用 startPublishCDNStream() 绑定非腾讯云直播 CDN 需要收取转推费用，且需要通过工单联系我们开通。

参数：

Name	Type	Description		
params	TRTCPublishCDN Param	转推参数		
		Properties		
		Name	Type	Description

	appId	Number	腾讯云直播服务的 AppID
	bizId	Number	腾讯云直播服务的 bizid
	url	String	指定该路音视频流在第三方直播服务商的推流地址，推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。
	streamId	String	需要转推的 streamId，默认值：空值。如果不填写，则默认转推调用者的旁路流。

#### 12.3.4. stopPublishCDNStream()

概述：停止向非腾讯云的直播 CDN 推流。

#### 12.3.5. setMixTranscodingConfig(config)

概述：设置云端的混流转码参数。

如果您在实时音视频 控制台 中的功能配置页开启了“启动自动旁路直播”功能，房间里的每一路画面都会有一个默认的直播 CDN 地址。

一个直播间中可能有不止一位主播，而且每个主播都有自己的画面和声音，但对于 CDN 观众来说，他们只需要一路直播流，所以您需要将多路音视频流混成一路标准的直播流，这就需要混流转码。

当您调用 setMixTranscodingConfig() 接口时，SDK 会向腾讯云的转码服务器发送一条指令，目的是将房间里的多路音视频流混合为一路，您可以通过 mixUsers 参数来调整每一路画面的位置，以及是否只混合声音，也可以通过 videoWidth、videoHeight、videoBitrate 等参数控制混合音视频流的编码参数。

```
【画面1】=> 解码 =====> \
  【画面2】=> 解码 => 画面混合 => 编码 => 【混合后的画面】
  / 
【画面3】=> 解码 =====> /
  【声音1】=> 解码 =====> \
    【声音2】=> 解码 => 声音混合 => 编码 => 【混合后的声音】
    / 
  【声音3】=> 解码 =====> /
```

注意：混流转码为收费功能，调用接口将产生云端混流转码费用，详见 云端混流转码计费说明。

Name	Type	Description
params	TRTCPublishCDNParam	请参考 trtc_define.js 中关于 TRTCTranscodingConfig 的介绍，如果传入 null 取消云端混流转码。
	Properties	
Name	Type	Description
mode	TRTCTranscodeMode	转码 config 模式

		ngConfigMode		
	appId	Number	腾讯云直播 AppID	
	bizId	Number	腾讯云直播 bizid	
	videoWidth	Number	最终转码后的视频分辨率的宽度 (px)	
	videoHeight	Number	最终转码后的视频分辨率的高度 (px)	
	videoBitrate	Number	最终转码后的视频分辨率的码率 (kbps)	
	videoFrameRate	Number	最终转码后的视频分辨率的帧率 (FPS)	
	videoGOP	Number	最终转码后的视频分辨率的关键帧间隔 (也被称为 GOP) , 单位秒	
	audioSampleRate	Number	最终转码后的音频采样率	
	audioBitrate	Number	最终转码后的音频码率, 单位: kbps	
	audioChannels	Number	最终转码后的音频声道数	
	backgroundColor	String	混合后画面的底色颜色, 格式为十六进制数字, 比如: “0x61B9F1” 代表 RGB 分别为(97,158,241)	
	treamId	String	输出到 CDN 上的直播流 ID	
	mixUsersArray	Array.<TRTCMixUser>	每一路子画面的位置信息	
	treamId	String	输出到 CDN 上的直播流 ID	
		Properties		
		Name	Type	Description
		userId	String	参与混流的 userId

				roomId	String	参与混流的 roomId，跨房流传入的实际 roomId。当前房间流，请传入空字符串 (roomId = '')。
				rect	Rect	图层位置坐标以及大小，左上角为坐标原点(0,0) (绝对像素值)
					Properties	
				Name	Type	Description
				left	Number	图层位置的左坐标
				top	Number	图层位置的上坐标
				right	Number	图层位置的右坐标
				bottom	Number	图层位置的下坐标
				Name	Type	Description
				zOrder	Number	图层层次 (1 - 15) 不可重复
				pureAudio	Boolean	是否纯音频
				streamType	TRTCVideoStreamType	TRTCVideoStreamTypeBig: 主路画面，一般用于摄像头； TRTCVideoStreamTypeSub: 辅路画面，一般用于屏幕分享。

## 12.4. 视频

### 12.4.1. startLocalPreview(view)

概述：启动本地摄像头采集和预览

这个接口会启动默认的摄像头，可以通过 `setCurrentCameraDevice()` 接口选用其它摄像头 当开始渲染首帧摄像头画面时，您会收到 `TRTCCallback` 中的 `onFirstVideoFrame(null)` 回调。

参数：

Name	Type	Description
view	HTMLElement	承载预览画面的 DOM

### 12.4.2. stopLocalPreview()

**概述：**停止本地摄像头采集和预览

### 12.4.3. muteLocalVideo(mute, streamType)

**概述：**暂停/恢复发布本地的视频流

该接口可以暂停（或恢复）发布本地的视频画面，暂停之后，同一房间中的其他用户将无法继续看到自己画面。该接口在指定 `TRTCVideoStreamTypeBig` 时等效于 `start/stopLocalPreview` 这两个接口，但具有更好的响应速度。因为 `start/stopLocalPreview` 需要打开和关闭摄像头，而打开和关闭摄像头都是硬件设备相关的操作，非常耗时。相比之下，`muteLocalVideo` 只需要在软件层面对数据流进行暂停或者放行即可，因此效率更高，也更适合需要频繁打开关闭的场景。当暂停/恢复发布指定 `TRTCVideoStreamTypeBig` 后，同一房间中的其他用户将会收到 `onUserVideoAvailable` 回调通知。当暂停/恢复发布指定 `TRTCVideoStreamTypeSub` 后，同一房间中的其他用户将会收到 `onUserSubStreamAvailable` 回调通知。

**参数：**

Name	Type	Description
mute	Boolean	true：屏蔽；false：开启，默认值：false
streamType	TRTCVideoStreamType	要暂停/恢复的视频流类型，仅支持 <code>TRTCVideoStreamTypeBig</code> 和 <code>TRTCVideoStreamTypeSub</code>

### 12.4.4. startRemoteView(userId, view, streamType)

**概述：**开始显示远端视频画面

在收到 SDK 的 `onUserVideoAvailable(userId, 1)` 通知时，可以获知该远程用户开启了视频，此后调用 `startRemoteView(userId)` 接口加载该用户的远程画面时，可以用 `loading` 动画优化加载过程中的等待体验。待该用户的首帧画面开始显示时，您会收到 `onFirstVideoFrame(userId)` 事件回调。

**参数：**

Name	Type	Description
userId	String	对方的用户标识
view	HTMLElement	承载预览画面的 DOM
streamType	TRTCVideoStreamType	视频流类型

### 12.4.5. stopRemoteView(userId, streamType)

**概述：**停止显示远端视频画面，同时不再拉取该远端用户的视频数据流

调用此接口后，SDK 会停止接收该用户的远程视频流，同时会清理相关的视频显示资源。

**参数：**

Name	Type	Description
userId	String	对方的用户标识
streamType	TRTCVideoStreamType	视频流类型

### 12.4.6. stopAllRemoteView()

概述：停止显示所有远端视频画面，同时不再拉取该远端用户的视频数据流

注意：如果有屏幕分享的画面在显示，则屏幕分享的画面也会一并被关闭。

### 12.4.7. muteRemoteVideoStream(userId, mute, streamType)

概述：暂停接收指定的远端视频流

该接口仅停止接收远程用户的视频流，但并不释放显示资源，所以视频画面会冻屏在 mute 前的最后一帧。

参数：

Name	Type	Description
userId	String	对方的用户标识
mute	Boolean	是否停止接收
streamType	TRTCVideoStream Type	视频流类型

### 12.4.8. muteAllRemoteVideoStreams(mute)

概述：停止接收所有远端视频流

Name	Type	Description
mute	Boolean	是否停止接收

### 12.4.9. setVideoEncoderParam(params)

概述：设置视频编码器相关参数

该设置决定了远端用户看到的画面质量（同时也是云端录制出的视频文件的画面质量）

参数：

Name	Type	Description
params	TRTCVideoEncoderParam	视频编码参数
		Properties
Name	Type	Description
videoResolution	TRTCVideoResolution	视频分辨率
resMode	TRTCVideoResolutionMode	分辨率模式（横屏分辨率 - 竖屏分辨率） TRTCVideoResolutionModeLandscape: 横屏分辨率 TRTCVideoResolutionModePortrait : 竖屏分辨率

			率
	videoFps	Number	视频采集帧率
	videoBitrate	Number	视频上行码率

#### 12.4.10. setNetworkQosParam(params)

概述：设置网络流控相关参数

该设置决定了 SDK 在各种网络环境下的调控策略（例如弱网下是“保清晰”还是“保流畅”）

参数：

Name	Type	Description		
params	TRTCNetworkQoSParam	网络流控参数		
		Properties		
Name	Type	Description		
preference	TRTCVideoQosPreference	弱网下是“保清晰”还是“保流畅” TRTCVideoQosPreferenceSmooth：弱网下保流畅，在遭遇弱网环境时首先确保声音的流畅和优先发送，画面会变得模糊且会有较多马赛克，但可以保持流畅不卡顿。 TRTCVideoQosPreferenceClear：弱网下保清晰在遭遇弱网环境时，画面会尽可能保持清晰，但可能会更容易出现卡顿。		
controlMode	TRTCQosControlMode	流控模式（云端控制 - 客户端控制） TRTCQosControlModeClient：客户端控制（用于 SDK 开发内部调试，客户请勿使用） TRTCQosControlModeServer：云端控制（默认）		

#### 12.4.11. setLocalRenderParams(params)

概述：设置本地图像（主流）的渲染参数

参数：

Name	Type	Description		
params	TRTCRenderParams	本地图像的参数		
		Properties		
Name	Type	Description		

		rotation	TRTCVideoRotation	视频画面旋转方向
		fillMode	TRTCVideoFillMode	视频画面填充模式
		mirrorType	TRTCVideoMirrorType	画面渲染镜像类型

#### 12.4.12. setLocalViewFillMode(mode)

废弃接口：设置本地图像的渲染模式

Deprecated:从 TRTCSDK 8.0 后该接口已被废弃，请使用 setLocalRenderParams 接口替代。

参数：

Name	Type	Description
mode	TRTCVideoFillMode	<p>填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTCVideoFillMode_Fit</p> <p>TRTCVideoFillMode_Fill: 图像铺满屏幕，超出显示视窗的视频部分将被截掉，所以画面显示可能不完整。</p> <p>TRTCVideoFillMode_Fit: 图像长边填满屏幕，短边区域会被填充黑色，但画面的内容肯定是完整的。</p>

#### 12.4.13. setRemoteRenderParams(userId, streamType, params)

概述：设置远端画面的渲染模式

参数：

Name	Type	Description
userId	String	用户 ID
streamType	TRTCVideoStreamType	视频流类型
params	TRTCRenderParams	本地画面渲染参数
		Properties
Name	Type	Description
rotation	TRTCVideoRotation	视频画面旋转方向
fillMode	TRTCVideoFillMode	视频画面填充模式

		mirrorType	TRTCVideoMirrorType	画面渲染镜像类型
--	--	------------	---------------------	----------

#### 12.4.14. setRemoteViewFillMode(userId, mode)

废弃接口：设置远端图像的渲染模式

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setRemoteRenderParams 接口替代。

参数：

Name	Type	Description
userId	String	用户 ID
mode	TRTCVideoFillMode	<p>填充（画面可能会被拉伸裁剪）或适应（画面可能不会有黑边），默认值：TRTCVideoFillMode_Fit</p> <p>TRTCVideoFillMode_Fill：图像铺满屏幕，超出显示视窗的视频部分将被截掉，所以画面显示可能不完整。</p> <p>TRTCVideoFillMode_Fit：图像长边填满屏幕，短边区域会被填充黑色，但画面的内容肯定是完整的。</p>

#### 12.4.15. setLocalViewRotation(rotation)

废弃接口：设置本地图像的顺时针旋转角度

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setLocalRenderParams 接口替代。

参数：

Name	Type	Description
rotation	TRTCVideoRotation	<p>支持 TRTCVideoRotation90 、 TRTCVideoRotation180 、 TRTCVideoRotation270 旋转角度，默认值：TRTCVideoRotation0</p> <p>TRTCVideoRotation0 : 顺时针旋转 0 度</p> <p>TRTCVideoRotation90 : 顺时针旋转 90 度</p> <p>TRTCVideoRotation180: 顺时针旋转 180 度</p> <p>TRTCVideoRotation270: 顺时针旋转 270 度</p>

#### 12.4.16. setRemoteViewRotation(userId, rotation)

废弃接口：设置远端图像的顺时针旋转角度

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setRemoteRenderParams 接口替代。

参数：

Name	Type	Description
userId	String	用户 ID
rotation	TRTCVideoRotation	支持 TRTCVideoRotation90 、 TRTCVideoRotation180 、

	n	TRTCVideoRotation270 旋转角度，默认值：TRTCVideoRotation0 TRTCVideoRotation0 : 顺时针旋转 0 度 TRTCVideoRotation90 : 顺时针旋转 90 度 TRTCVideoRotation180: 顺时针旋转 180 度 TRTCVideoRotation270: 顺时针旋转 270 度
--	---	--

#### 12.4.17. setVideoEncoderRotation(rotation)

设置视频编码输出的（也就是远端用户观看到的，以及服务器录制下来的）画面方向

参数：

Name	Type	Description
rotation	TRTCVideoRotation	目前支持 TRTCVideoRotation0 和 TRTCVideoRotation180 两个旋转角度，默认值：TRTCVideoRotation0 TRTCVideoRotation0 : 顺时针旋转 0 度 TRTCVideoRotation180: 顺时针旋转 180 度

#### 12.4.18. setLocalViewMirror(mirror)

废弃接口：设置本地摄像头预览画面的镜像模式

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setLocalRenderParams 接口替代。

参数：

Name	Type	Description
mirror	Boolean	镜像模式, windows 默认值: false(非镜像模式), mac 默认值: true(镜像模式)

#### 12.4.19. setVideoEncoderMirror(mirror)

设置编码器输出的画面镜像模式

该接口不改变本地摄像头的预览画面，但会改变另一端用户看到的（以及服务器录制的）画面效果。

参数：

Name	Type	Description
mirror	Boolean	是否开启远端镜像, true: 远端画面镜像; false: 远端画面非镜像。默认值: false

#### 12.4.20. enableSmallVideoStream(enable, params)

概述：开启大小画面双路编码模式

如果当前用户是房间中的主要角色（例如主播、老师、主持人等），并且使用 PC 或者 Mac 环境，可以开启该模式。开启该模式后，当前用户会同时输出【高清】和【低清】两路视频流（但只有一路音频流）。对于开启该模式的当前用户，会占用更多的网络带宽，并且会更加消耗 CPU 计算资源。

对于同一房间的远程观众而言：

- 如果用户的下行网络很好，可以选择观看【高清】画面
- 如果用户的下行网络较差，可以选择观看【低清】画面

参数：

Name	Type	Description	
enable	Boolean	是否开启小画面编码，默认值：false	
params	TRTCVideoEncParam	小流的视频参数	
		Properties	
	Name	Type	
	videoResolution	TRTCVideoResolution	视频分辨率
	resMode	TRTCVideoResolutionMode	分辨率模式（横屏分辨率 - 竖屏分辨率） TRTCVideoResolutionModeLandscape: 横屏分辨率 TRTCVideoResolutionModePortrait : 竖屏分辨率
	videoFps	Number	视频采集帧率
	videoBitrate	Number	视频上行码率
	minVideoBitrate	Number	视频最小码率

#### 12.4.21. setRemoteVideoStreamType(userId, streamType)

选定观看指定 userId 的大画面或小画面

此功能需要该 userId 通过 enableSmallVideoStream 提前开启双路编码模式。如果该 userId 没有开启双路编码模式，则此操作无效。

参数：

Name	Type	Description
userId	String	用户 ID
streamType	TRTCVideoStreamType	视频流类型，即选择看大画面还是小画面，默认为 TRTCVideoStreamTypeBig TRTCVideoStreamTypeBig : 大画面视频流 TRTCVideoStreamTypeSmall: 小画面视频流

### 12.4.22. snapshotVideo(userId, streamType)

**概述：**视频画面截图

调用截图接口后，您会收到来自 TRTCCallback 中的 onSnapshotComplete 回调：截取本地、远程主路和远端辅流的视频画面，并通过 HBITMAP 对象返回给您。

Name	Type	Description
userId	String	用户 ID，空字符串表示截取本地视频画面
streamType	TRTCVideoStream Type	视频流类型，支持摄像头画面 (TRTCVideoStreamTypeBig) 和屏幕分享画面 (TRTCVideoStreamTypeSub)

### 12.4.23. setPriorRemoteVideoStreamType(type)

**废弃接口：**设定观看方优先选择的视频质量

低端设备推荐优先选择低清晰度的小画面。如果对方没有开启双路视频模式，则此操作无效。

**Deprecated:** 从 TRTCSDK 8.0 后该接口已被废弃，请使用 startRemoteView 接口设置视频流类型参数进行替代。

**参数：**

Name	Type	Description
type	TRTCVideoStream Type	默认观看大画面还是小画面，默认为 TRTCVideoStreamTypeBig TRTCVideoStreamTypeBig：大画面视频流 TRTCVideoStreamTypeSmall：小画面视频流

### 12.4.24. startLocalRecording(options)

**开启本地媒体录制**

开启后把直播过程中的音视和视频内容录制到本地的一个文件中。

**参数：**

Name	Type	Description
options	Object	录制参数
		Properties
Name	Type	Description
filePath	string	<p>录制文件路径。</p> <p>录制的文件地址（必填），请确保路径有读写权限且合法，否则录制文件无法生成。</p> <p>【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录制出的文件格式，目前支持的格式暂时只有 MP4。</p> <p>例如：假如您指定路径为 "mypath/record/test.mp4"，代表您希望 SDK 生成一个 MP4 格式的本地视频文件。</p>

			请您指定一个有读写权限的合法路径，否则录制文件无法生成。
	recordType	TRTCRecordType	媒体录制类型，默认值：TRTCRecordTypeBoth，即同时录制音频和视频。
	interval	number	录制进行中事件 onLocalRecording 触发频率，单位毫秒，有效范围：1000-10000 和 -1。默认值为-1，表示不触发录制进行中事件。

### 12.4.25. stopLocalRecording()

停止本地媒体录制

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

## 12.5. 音频

### 12.5.1. startLocalAudio(quality)

开启本地音频的采集和上行

该函数会启动麦克风采集，并将音频数据传输给房间里的其他用户。SDK 并不会默认开启本地的音频上行，也就说，如果您不调用这个函数，房间里的其他用户就听不到您的声音。

注意：TRTC SDK 并不会默认打开本地的麦克风采集。

参数：

Name	Type	Description
quality	TRTCAudioQuality	音频质量 TRTCAudioQualitySpeech: 语音模式：采样率：16k TRTCAudioQualityDefault: 标准模式（或者默认模式）：采样率：48k TRTCAudioQualityMusic: 音乐模式：采样率：48k

### 12.5.2. stopLocalAudio()

关闭本地音频的采集和上行

当关闭本地音频的采集和上行，房间里的其它成员会收到 onUserAudioAvailable(false) 回调通知。

### 12.5.3. muteLocalAudio(mute)

静音本地的音频

当静音本地音频后，房间里的其它成员会收到 onUserAudioAvailable(false) 回调通知。与 stopLocalAudio 不同之处在于，muteLocalAudio 并不会停止发送音视频数据，而是会继续发送码率极低的静音包。在对录制质量要求很高的场景中，选择 muteLocalAudio 是更好的选择，能录制出兼容性更好的 MP4 文件。这是由于 MP4 等视频文件格式，对于音频的连续性是要求很高的，简单粗暴地 stopLocalAudio 会导致录制出的 MP4 不易播放。

参数：

Name	Type	Description
mute	Boolean	true：屏蔽；false：开启，默认值：false

#### 12.5.4. muteRemoteAudio(userId, mute)

静音掉某一个用户的声音，同时不再拉取该远端用户的音频数据流

参数：

Name	Type	Description
userId	String	用户 ID
mute	Boolean	true：静音；false：非静音

#### 12.5.5. muteAllRemoteAudio(mute)

静音/取消静音所有远端用户的音频流

当您静音所有用户的远端音频时，SDK 会停止播放所有来自远端的音频流，同时也会停止拉取所有用户的音频数据。

注意：该接口支持您在进入房间（enterRoom）前调用，暂停状态会在退出房间（exitRoom）在之后会被重置。

参数：

Name	Type	Description
mute	Boolean	true：静音；false：非静音

#### 12.5.6. setRemoteAudioVolume(userId, volume)

设置某个远程用户的播放音量

参数：

Name	Type	Description
userId	String	远程用户 ID
volume	Number	音量大小，100 为原始音量，范围是：[0 ~ 100]，默认值为 100

#### 12.5.7. setAudioCaptureVolume(volume)

设置 SDK 采集音量

参数：

Name	Type	Description
volume	Number	音量大小，取值 0 - 100，默认值为 100

#### 12.5.8. getAudioCaptureVolume() → {Number}

获取 SDK 采集音量

Returns:

SDK 采集音量

Type **Number**

### 12.5.9. setAudioPlayoutVolume(volume)

设置 SDK 播放音量

注意：在混合远程用户、Bgm和音效的音频流后，送入系统播放前生效。 会影响本地录制的音量大小。 不会影响耳返的音量。

参数：

Name	Type	Description
volume	Number	音量大小，取值 0 - 100，默认值为 100

### 12.5.10. getAudioPlayoutVolume() → {Number}

获取 SDK 播放音量

Returns:

SDK 播放音量

Type **Number**

### 12.5.11. enableAudioVolumeEvaluation(interval)

启用或关闭音量大小提示

开启此功能后，SDK 会在 onUserVoiceVolume() 中反馈对每一路声音音量大小值的评估。 我们在 Demo 中有一个音量大小的提示条，就是基于这个接口实现的。 如希望打开此功能，请在 startLocalAudio() 之前调用。

参数：

Name	Type	Description
interval	Number	设置 onUserVoiceVolume 回调的触发间隔，单位为 ms，最小间隔为 100ms，如果小于等于 0 则会关闭回调，建议设置为 300ms

### 12.5.12. startAudioRecording(params) → {Number}

开始录音

该方法调用后，SDK 会将通话过程中的所有音频(包括本地音频，远端音频，BGM等)录制到一个文件里。 无论是否进房，调用该接口都生效。 如果调用 exitRoom 时还在录音，录音会自动停止。

注意：录音文件路径需精确到文件名及格式后缀，格式后缀决定录制文件的格式。 例如：指定路径为 path/to/audio.aac，则会生成一个 AAC 格式的文件。目前支持的格式有 PCM, WAV, AAC

参数：

Name	Type	Description
params	TRTCAudioRecordingParams	录音参数。
		Properties
Name	Type	Description

		filePath	String	文件路径（必填），录音文件的保存路径。该路径需要用户自行指定，请确保路径存在且可写
		recordingContent	TRTCAudioRecordingContent	音频录制内容类型

Returns:

0: 成功; -1: 录音已开始; -2: 文件或目录创建失败; -3: 后缀指定的音频格式不支持

注意： - 在 9.3 之前的版本，params 为 string 代表 录音文件路径（必填），9.3 以及以后同时支持 string 类型和 TRTCAudioRecordingParams。

Type **Number**

### 12.5.13. stopAudioRecording()

停止录音

如果调用 exitRoom 时还在录音，录音会自动停止。

### 12.5.14. setAudioQuality(quality)

废弃接口：设置音频质量

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 startLocalAudio 接口设置音频质量参数进行替代。设置音频质量，主播端的音质越高，观众端的听感越好，但传输所依赖的带宽也就越高，在带宽有限的场景下也更容易出现卡顿。注意：该方法需要在 startLocalAudio 之前进行设置，否则不会生效。

参数：

Name	Type	Description
quality	TRTCAudioQuality	音频质量 TRTCAudioQualitySpeech: 语音模式：采样率：16k TRTCAudioQualityDefault: 标准模式（或者默认模式）：采样率：48k TRTCAudioQualityMusic: 音乐模式：采样率：48k

### 12.5.15. setMicVolumeOnMixing(volume)

废弃接口：设置麦克风的音量大小

Deprecated: 从 TRTCSDK6.9 后该接口已被废弃，请使用 setAudioCaptureVolume 接口替代。

参数：

Name	Type	Description
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 200。

## 12.6. 摄像头

### 12.6.1. getCameraDevicesList() → {Array.<TRTCDeviceInfo>}

获取摄像头设备列表

示例：

```
1. var cameralist = rtcCloud.getCameraDevicesList();
2. for (i=0;i<cameralist.length;i++) {
3.     var camera = cameralist[i];
4.     console.info("camera deviceName: " + camera.deviceName + " deviceId:" + camera.deviceId);
5. }
```

Returns:

摄像头管理器列表

Type **Array.<TRTCDeviceInfo>**

### 12.6.2. setCurrentCameraDevice(deviceId)

设置要使用的摄像头

参数：

Name	Type	Description
deviceId	String	从 getCameraDevicesList 中得到的设备 ID

### 12.6.3. getCurrentCameraDevice() → {TRTCDeviceInfo}

获取当前使用的摄像头

Returns:

设备信息，能获取设备 ID 和设备名称

Type **TRTCDeviceInfo**

## 12.7. 音频设备

### 12.7.1. getMicDevicesList() → {Array.<TRTCDeviceInfo>}

获取麦克风设备列表

示例：

```
1. var miclist = rtcCloud.getMicDevicesList();
2. for (i=0;i<miclist.length;i++) {
3.     var mic = miclist[i];
4.     console.info("mic deviceName: " + mic.deviceName + " deviceId:" + mic.deviceId);
5. }
```

Returns:

麦克风管理器列表

Type **Array.<TRTCDeviceInfo>**

### 12.7.2. getCurrentMicDevice() → {TRTCDeviceInfo}

获取当前选择的麦克风

Returns:

设备信息，能获取设备 ID 和设备名称

Type **TRTCDeviceInfo**

### 12.7.3. setCurrentMicDevice(micId)

设置要使用的麦克风

选择指定的麦克风作为录音设备，不调用该接口时，默认选择索引为0的麦克风

参数：

Name	Type	Description
micId	String	从 getMicDevicesList 中得到的设备 ID

### 12.7.4. getCurrentMicDeviceVolume() → {Number}

获取系统当前麦克风设备音量

注意：查询的是系统硬件音量大小。

Returns:

音量值，范围是0 - 100

Type **Number**

### 12.7.5. setCurrentMicDeviceVolume(volume)

设置系统当前麦克风设备的音量

注意：该接口的功能是调节系统采集音量，如果用户直接调节系统设置的采集音量时，该接口的设置结果会被用户的操作所覆盖。

参数：

Name	Type	Description
volume	Number	麦克风音量值，范围 0 - 100

### 12.7.6. setCurrentMicDeviceMute(mute)

设置系统当前麦克风设备的静音状态

参数：

Name	Type	Description
mute	Boolean	设置为 true 时，麦克风设备静音；设置为 false 时，麦克风设备取消静音

### 12.7.7. getCurrentMicDeviceMute() → {Boolean}

获取系统当前麦克风设备是否静音

Returns:

静音状态

Type **Boolean**

### 12.7.8. getSpeakerDevicesList() → {Array.<TRTCDeviceInfo>}

获取扬声器设备列表

示例：

```
1. var speakerlist = rtcCloud.getSpeakerDevicesList();
2. for (i=0;i<speakerlist.length;i++) {
3.     var speaker = speakerlist[i];
4.     console.info("mic deviceName: " + speaker.deviceName + " deviceId:" + speaker.deviceId);
5. }
```

Returns:

扬声器管理器列表

Type **Array.<TRTCDeviceInfo>**

### 12.7.9. getCurrentSpeakerDevice() → {TRTCDeviceInfo}

获取当前的扬声器设备

Returns:

设备信息，能获取设备 ID 和设备名称

Type **TRTCDeviceInfo**

### 12.7.10. setCurrentSpeakerDevice(speakerId)

设置要使用的扬声器

参数：

Name	Type	Description
speakerId	String	从 getSpeakerDevicesList 中得到的设备 ID

### 12.7.11. getCurrentSpeakerVolume() → {Number}

获取系统当前扬声器设备音量

Returns:

扬声器音量，范围0 - 100

Type **Number**

### 12.7.12. setCurrentSpeakerVolume(volume)

设置系统当前扬声器设备音量

注意：该接口的功能是调节系统播放音量，如果用户直接调节系统设置的播放音量时，该接口的设置结果会被用户的操作所覆盖。

参数：

Name	Type	Description
volume	Number	设置的扬声器音量，范围 0 - 100

### 12.7.13. setCurrentSpeakerDeviceMute(mute)

设置系统当前扬声器设备的静音状态

参数：

Name	Type	Description
mute	Boolean	设置为 true 时，扬声器设备静音；设置为 false 时，扬声器设备取消静音

### 12.7.14. getCurrentSpeakerDeviceMute() → {Boolean}

获取系统当前扬声器设备是否静音

Returns:

静音状态

Type Boolean

## 12.8. 音头像前处理

### 12.8.1. setBeautyStyle(style, beauty, white, ruddiness)

设置美颜、美白、红润效果级别

SDK 内部集成了两套风格不同的磨皮算法，一套我们取名叫“光滑”，适用于美女秀场，效果比较明显。另一套我们取名“自然”，磨皮算法更多地保留了面部细节，主观感受上会更加自然。

参数：

Name	Type	Description
style	TRTCBeautyStyle	美颜风格，光滑或者自然，光滑风格磨皮更加明显，适合娱乐场景。 TRTCBeautyStyleSmooth: 光滑，适用于美女秀场，效果比较明显。 TRTCBeautyStyleNature: 自然，磨皮算法更多地保留了面部细节，主观感受上会更加自然。
beauty	Number	美颜级别，取值范围 0 - 9，0 表示关闭，1 - 9 值越大，效果越明显
white	Number	美白级别，取值范围 0 - 9，0 表示关闭，1 - 9 值越大，效果越明显
ruddiness	Number	红润级别，取值范围 0 - 9，0 表示关闭，1 - 9 值越大，效果越明显，该参数 windows 平台暂未生效

### 12.8.2. setWaterMark(streamType, srcData, srcType, nWidth, nHeight, xOffset, yOffset, fWidthRatio)

概述：设置水印

水印的位置是通过 xOffset, yOffset, fWidthRatio 来指定的。

- xOffset: 水印的坐标，取值范围为0 - 1的浮点数。
- yOffset: 水印的坐标，取值范围为0 - 1的浮点数。
- fWidthRatio: 水印的大小比例，取值范围为0 - 1的浮点数。

参数：

Name	Type	Description
streamType	TRTCVideoStreamType	要设置水印的流类型 (TRTCVideoStreamTypeBig、TRTCVideoStreamTypeSub)
srcData	ArrayBuffer   String	水印图片源数据 (传 null 表示去掉水印)
srcType	TRTCWaterMarkSrcType	水印图片源数据类型 TRTCWaterMarkSrcTypeFile：图片文件路径，支持 BMP、GIF、JPEG、PNG、TIFF、Exif、WMF 和 EMF 文件格式 TRTCWaterMarkSrcTypeBGRA32: BGRA32 格式内存块 TRTCWaterMarkSrcTypeRGBA32: RGBA32 格式内存块
nWidth	Number	水印图片像素宽度 (源数据为文件路径时忽略该参数)
nHeight	Number	水印图片像素高度 (源数据为文件路径时忽略该参数)
xOffset	Number	水印显示的左上角 x 轴偏移
yOffset	Number	水印显示的左上角 y 轴偏移
fWidthRatio	Number	水印显示的宽度占画面宽度比例 (水印按该参数等比例缩放显示)

## 12.9. 辅流 (屏幕分享)

### 12.9.1. startRemoteSubStreamView(userId, view)

废弃接口：开始显示远端用户的辅路画面 (TRTCVideoStreamTypeSub，一般用于屏幕分享)

startRemoteView() 用于显示主路画面 (TRTCVideoStreamTypeBig，一般用于摄像头)。

startRemoteSubStreamView() 用于显示辅路画面 (TRTCVideoStreamTypeSub，一般用于屏幕分享)。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 startRemoteView 接口替代。注意：请在 onUserSubStreamAvailable 回调后再调用这个接口。

参数：

Name	Type	Description
userId	String	对方的用户标识
view	HTMLElement	承载预览画面的 DOM

### 12.9.2. stopRemoteSubStreamView(userId)

废弃接口：停止显示远端用户的辅路画面 (TRTCVideoStreamTypeSub，一般用于屏幕分享)。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 stopRemoteView 接口替代。

参数：

Name	Type	Description
userId	String	对方的用户标识

### 12.9.3. setRemoteSubStreamViewFillMode(userId, mode)

废弃接口：设置辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的显示模式。

setRemoteViewFillMode() 用于设置远端主路画面（TRTCVideoStreamTypeBig，一般用于摄像头）的显示模式。

setRemoteSubStreamViewFillMode() 用于设置远端辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的显示模式。

**Deprecated:** 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setRemoteRenderParams 接口替代。

参数：

Name	Type	Description
userId	String	用户的 ID
mode	TRTCVideoFillMode	<p>填充（画面可能会被拉伸裁剪）或适应（画面可能不会有黑边），默认值： TRTCVideoFillMode_Fit</p> <p>TRTCVideoFillMode_Fill：图像铺满屏幕，超出显示视窗的视频部分将被截掉，所以画面显示可能不完整。</p> <p>TRTCVideoFillMode_Fit：图像长边填满屏幕，短边区域会被填充黑色，但画面的内容肯定是完整的。</p>

### 12.9.4. setRemoteSubStreamViewRotation(userId, rotation)

废弃接口：设置辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的顺时针旋转角度

setRemoteViewRotation() 用于设置远端主路画面（TRTCVideoStreamTypeBig，一般用于摄像头）的旋转角度。

setRemoteSubStreamViewRotation() 用于设置远端辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的旋转角度。

**Deprecated:** 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setRemoteRenderParams 接口替代。

参数：

Name	Type	Description
userId	String	用户 ID
rotation	TRTCVideoRotation	支持 90、180、270 旋转角度

### 12.9.5. getScreenCaptureSources(thumbWidth, thumbHeight, iconWidth, iconHeight) → {Array.<TRTCScreenCaptureSourceInfo>}

枚举可共享的窗口列表

如果您要给您的 App 增加屏幕分享功能，一般需要先显示一个窗口选择界面，这样用户可以选择希望分享的窗口。通过此函数，您可以获得可分享窗口的 ID、类型、窗口名称以及缩略图。拿到这些信息后，您就可以实现一个窗口选择界面，当然，您也可以使用我们在 Demo 源码中已经实现好的一个界面。

注意：返回的列表中包括屏幕和应用窗口，屏幕会在列表的前面几个元素中。

参数：

Name	Type	Description
thumbWidth	Number	缩略图宽度，指定要获取的窗口缩略图大小，缩略图可用于绘制在窗口选择界面上
thumbHeight	Number	缩略图高度，指定要获取的窗口缩略图大小，缩略图可用于绘制在窗口选择界面上
iconWidth	Number	图标宽度，指定要获取的窗口图标大小
iconHeight	Number	图标高度，指定要获取的窗口图标大小

Returns:

窗口列表包括屏幕

Type `Array.<TRTCScreenCaptureSourceInfo>`

## 12.9.6. selectScreenCaptureTarget(source, captureRect, property)

设置屏幕共享参数，该方法在屏幕共享过程中也可以调用

如果您期望在屏幕分享的过程中，切换想要分享的窗口，可以再次调用这个函数而不需要重新开启屏幕分享。

支持如下四种情况：

- 共享整个屏幕：sourceInfoList 中 type 为 Screen 的 source，captureRect 设为 { 0, 0, 0, 0 }
- 共享指定区域：sourceInfoList 中 type 为 Screen 的 source，captureRect 设为非 NULL，例如 { 100, 100, 300, 300 }
- 共享整个窗口：sourceInfoList 中 type 为 Window 的 source，captureRect 设为 { 0, 0, 0, 0 }
- 共享窗口区域：sourceInfoList 中 type 为 Window 的 source，captureRect 设为非 NULL，例如 { 100, 100, 300, 300 }

示例：

```

1. // 示例 1: 选择要分享的窗口或屏幕
2. import TRTCCloud, {
3.   Rect,
4.   TRTCScreenCaptureProperty
5. } from 'trtc-electron-sdk';
6. const rtcCloud = new TRTCCloud();
7. const screenAndWindows = rtcCloud.getScreenCaptureSources(320, 180, 32, 32);
8. const selectedScreenOrWindow = screenAndWindows[0];
9. const selectRect = new Rect(0, 0, 0, 0);
10. const captureProperty = new TRTCScreenCaptureProperty(
11.   true, // enable capture mouse
12.   true, // enable highlight
13.   true, // enable high performance
14.   0, // default highlight color
15.   0, // default highlight width
16.   false // disable capture child window
17. );
18. rtcCloud.selectScreenCaptureTarget(
19.   selectedScreenOrWindow,

```

```

20. selectRect,
21. captureProperty
22. );

```

```

1. // 示例 2: 选择要分享的窗口或屏幕, 废弃的接口调用方式
2. import TRTCCloud, { Rect } from 'trtc-electron-sdk';
3. const rtcCloud = new TRTCCloud();
4. const screenAndWindows = rtcCloud.getScreenCaptureSources(320, 180, 32, 32);
5. const selectedScreenOrWindow = screenAndWindows[0];
6. const selectRect = new Rect(0, 0, 0, 0);
7. rtcCloud.selectScreenCaptureTarget(
8.   selectedScreenOrWindow.type,
9.   selectedScreenOrWindow.sourceId,
10.  selectedScreenOrWindow.sourceName,
11.  selectRect,
12.  true, // enable capture mouse
13.  true // enable highlight
14. );

```

参数：

Name	Type	Description
source	TRTCScreenCaptureSourceInfo	指定分享源, 详情参考 TRTCScreenCaptureSourceInfo 定义
captureRect	Rect	指定捕获的区域
property	TRTCScreenCaptureProperty	指定屏幕分享目标的属性, 包括捕获鼠标, 高亮捕获窗口等, 详情参考 TRTCScreenCaptureProperty 定义

## 12.9.7. startScreenCapture(view, type, params)

启动屏幕分享, 支持选择使用主路或辅路进行屏幕分享。

注意: 一个用户同时最多只能上传一条主路 (TRTCVideoStreamTypeBig) 画面和一条辅路 (TRTCVideoStreamTypeSub) 画面, 默认情况下, 屏幕分享使用辅路画面, 如果使用主路画面, 建议您提前停止摄像头采集 (stopLocalPreview) 避免相互冲突。

示例：

```

1. // 分享屏幕或窗口
2. import TRTCCloud, {
3.   TRTCVideoStreamType,
4.   TRTCVideoEncParam,
5.   TRTCVideoResolution,
6.   TRTCVideoResolutionMode
7. } from 'trtc-electron-sdk';
8. const rtcCloud = new TRTCCloud();
9. const screenAndWindows = rtcCloud.getScreenCaptureSources(320, 180, 32, 32);
10. const selectedScreenOrWindow = screenAndWindows[0];
11. const selectRect = new Rect(0, 0, 0, 0);

```

```

12. const captureProperty = new TRTCScreenCaptureProperty(
13.   true, // enable capture mouse
14.   true, // enable highlight
15.   true, // enable high performance
16.   0, // default highlight color
17.   0, // default highlight width
18.   false // disable capture child window
19. );
20. rtcCloud.selectScreenCaptureTarget(
21.   selectedScreenOrWindow,
22.   selectRect,
23.   captureProperty
24. );
25. const screenShareEncParam = new TRTCVideoEncParam(
26.   TRTCVideoResolution.TRTCVideoResolution_1280_720,
27.   TRTCVideoResolutionMode.TRTCVideoResolutionModeLandscape,
28.   15,
29.   1600,
30.   0,
31.   true,
32. );
33. rtcCloud.startScreenCapture(
34.   view, // HTML Element
35.   TRTCVideoStreamType.TRTCVideoStreamTypeSub,
36.   screenShareEncParam,
37. );

```

参数：

Name	Type	Description
view	HTMLElement	null
type	TRTCVideoStream Type	
params	TRTCVideoEncPar am	null

### 12.9.8. pauseScreenCapture()

概述：暂停屏幕分享

### 12.9.9. resumeScreenCapture()

概述：恢复屏幕分享

### 12.9.10. stopScreenCapture()

概述：停止屏幕分享

### 12.9.11. setSubStreamEncoderParam(params)

概述：设置屏幕分享（即辅路）的视频编码参数

该接口可以设定远端用户所看到的屏幕分享（即辅路）的画面质量，同时也能决定云端录制出的视频文件中屏幕分享的画面质量。请注意如下两个接口的差异：

- `setVideoEncoderParam()` 用于设置主路画面（`TRTCVideoStreamTypeBig`, 一般用于摄像头）的视频编码参数。
- `setSubStreamEncoderParam()` 用于设置辅路画面（`TRTCVideoStreamTypeSub`, 一般用于屏幕分享）的视频编码参数。

注意：即使您使用主路传输屏幕分享（在调用 `startScreenCapture` 时设置 `type=TRTCVideoStreamTypeBig`），依然要使用 `setSubStreamEncoderParam` 设定屏幕分享的编码参数，而不要使用 `setVideoEncoderParam`。

参数：

Name	Type	Description		
params	TRTCVideoEncParam	辅流（屏幕分享）编码参数		
		Properties		
Name	Type	Description		
videoResolution	TRTCVideoResolution	视频分辨率		
resMode	TRTCVideoResolutionMode	分辨率模式（横屏分辨率 - 竖屏分辨率） TRTCVideoResolutionModeLandscape: 横屏分辨率 TRTCVideoResolutionModePortrait : 竖屏分辨率		
videoFps	Number	视频采集帧率		
videoBitrate	Number	视频上行码率		
minVideoBitrate	Number	最小码率		

### 12.9.12. setSubStreamMixVolume(volume)

设置辅流（屏幕分享）的混音音量大小

这个数值越高，辅路音量的占比就约高，麦克风音量占比就越小，所以不推荐设置得太大，否则麦克风的声音就被压制了。

参数：

Name	Type	Description	
volume	Number	设置的混音音量大小，范围 0 - 100	

### 12.9.13. addExcludedShareWindow(win)

将指定窗口加入屏幕分享的排除列表中，加入排除列表中的窗口不会被分享出去  
支持启动屏幕分享前设置过滤窗口，也支持屏幕分享过程中动态添加过滤窗口。

参数：

Name	Type	Description
win	String	不希望分享出去的窗口 该方法只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeScreen 时生效，即分享屏幕时生效 Windows 下该方法添加的窗口列表会在退房后清空；Mac 下在退出房间时不会自动清空，需要调用一次 removeAllExcludedShareWindow() 方法清空列表。

### 12.9.14. removeExcludedShareWindow(win)

将指定窗口从屏幕分享的排除列表中移除

参数：

Name	Type	Description
win	String	不希望分享出去的窗口 该方法只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeScreen 时生效，即分享屏幕时生效

## 12.10. 自定义消息发送

### 12.10.1. removeAllExcludedShareWindow()

将所有窗口从屏幕分享的排除列表中移除

### 12.10.2. sendCustomCmdMsg(cmdId, msg, reliable, ordered) → {Boolean}

发送自定义消息给房间内所有用户

该接口可以借助音视频数据通道向当前房间里的其他用户广播您自定义的数据，但因为复用了音视频数据通道，请务必严格控制自定义消息的发送频率和消息体的大小，否则会影响音视频数据的质量控制逻辑，造成不确定性的后果。

注意：本接口有以下限制：

- 发送消息到房间内所有用户，每秒最多能发送30条消息。
- 每个包最大为1KB，超过则很有可能会被中间路由器或者服务器丢弃。
- 每个客户端每秒最多能发送总计8KB数据。
- 将 reliable 和 ordered 同时设置为 true 或 false，暂不支持交叉设置。
- 强烈建议不同类型的消息使用不同的 cmdID，这样可以在要求有序的情况下减小消息时延。

参数：

Name	Type	Description
cmdId	Number	消息 ID，取值范围为 1 - 10
msg	String	待发送的消息，最大支持 1KB (1000 字节) 的数据大小
reliable	Boolean	是否可靠发送，可靠发送的代价是会引入一定的延时，因为接收端要暂存一段时间的数据来等待重传
ordered	Boolean	是否要求有序，即是否要求接收端接收的数据顺序和发送端发送的顺序一致，这会带来一定的接收延时，因为在接收端需要暂存并排序这些消息

Returns:

true：消息已经发出；false：消息发送失败

Type **Boolean**

## 12.11. 背景混音

### 12.11.1. playBGM(path)

废弃接口：启动播放背景音乐

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 startPlayMusic 接口替代。

参数：

Name	Type	Description
path	String	音乐文件路径

### 12.11.2. stopBGM()

废弃接口：停止播放背景音乐

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 stopPlayMusic 接口替代。

### 12.11.3. pauseBGM()

废弃接口：暂停播放背景音乐

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 pausePlayMusic 接口替代。

### 12.11.4. resumeBGM()

废弃接口：继续播放背景音乐

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 resumePlayMusic 接口替代。

### 12.11.5. getBGMDuration(path) → {Number}

废弃接口：获取背景音乐文件总时长，单位毫秒

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 getMusicDurationInMS 接口替代。

参数：

Name	Type	Description
path	String	音乐文件路径

Returns:

成功返回时长，失败返回-1

Type Number

### 12.11.6. setBGMPosition(pos)

废弃接口：设置背景音乐播放进度

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 seekMusicToPosInTime 接口替代。

参数：

Name	Type	Description
pos	Number	单位毫秒

### 12.11.7. setBGMVolume(volume)

废弃接口：设置背景音乐的音量大小，播放背景音乐混音时使用，用来控制背景音音量大小

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setAllMusicVolume 接口替代。

参数：

Name	Type	Description
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 200。

### 12.11.8. setBGMPlayoutVolume(volume)

废弃接口：设置背景音乐本地播放音量的大小

播放背景音乐混音时使用，用来控制背景音乐在本地播放时的音量大小。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setMusicPlayoutVolume 接口替代。

参数：

Name	Type	Description
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 100；默认值：100

### 12.11.9. setBGMPublishVolume(volume)

废弃接口：设置背景音乐远端播放音量的大小

播放背景音乐混音时使用，用来控制背景音乐在远端播放时的音量大小。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setMusicPublishVolume 接口替代。

参数：

Name	Type	Description
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 100；默认值：100

### 12.11.10. startSystemAudioLoopback(path)

打开系统声音采集

开启后可以采集整个操作系统的播放声音（path 为空）或某一个播放器（path 不为空）的声音，并将其混入到当前麦克风采集的声音中一起发送到云端。

参数：

Name	Type	Description
path	String	不传 path 或为 null，代表采集整个操作系统的聲音；path 填写 exe 程序（如 QQ 音乐）所在的路径，将会启动此程序并只采集此程序的聲音。

### 12.11.11. stopSystemAudioLoopback()

关闭系统声音采集

### 12.11.12. setSystemAudioLoopbackVolume(volume)

设置系统声音采集的音量

参数：

Name	Type	Description
volume	Number	音量大小，取值范围为 0 - 100。

## 12.12. 音效

### 12.12.1. startPlayMusic(musicParam, callbackMap)

概述：启动播放背景音乐

示例：

```

1. // 音乐播放
2. import TRTCCloud, { AudioMusicParam } from 'trtc-electron-sdk';
3. const rtcCloud = new TRTCCloud();
4. const params = new AudioMusicParam();
5. params.id = 1;
6. params.path = 'path';
7. params.publish = true;
8. rtcCloud.startPlayMusic(params, {
9.     onStart: (id: number, errCode: number) => {
10.         console.log(`onStart, id: ${id}, errorCode: ${errCode}`);
11.     },
12.     onPlayProgress: (id: number, curPtsMS: number, durationMS: number) => {
13.         console.log(`onPlayProgress, id: ${id}, curPtsMS: ${curPtsMS}, durationMS: ${durationMS}`);
14.     },
15.     onComplete: (id: number, errCode: number) => {
16.         console.log(`onComplete, id: ${id}, errorCode: ${errCode}`);
17.     }
18. })

```

参数：

Name	Type	Description												
musicParam	AudioMusicParam	背景音乐参数												
callback Map	Object	播放音乐事件回调(可选)												
		Properties <table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>onStart</td><td>function</td><td>开始播放背景音乐回调</td></tr> <tr> <td>onPlayProgress</td><td>function</td><td>播放背景音乐的进度</td></tr> <tr> <td>onComplete</td><td>function</td><td>播放背景音乐结束</td></tr> </tbody> </table>	Name	Type	Description	onStart	function	开始播放背景音乐回调	onPlayProgress	function	播放背景音乐的进度	onComplete	function	播放背景音乐结束
Name	Type	Description												
onStart	function	开始播放背景音乐回调												
onPlayProgress	function	播放背景音乐的进度												
onComplete	function	播放背景音乐结束												

### 12.12.1. stopPlayMusic(id)

概述：停止播放背景音乐

参数：

Name	Type	Description
id	Number	音乐 ID

### 12.12.2. pausePlayMusic(id)

暂停播放背景音乐

参数：

Name	Type	Description
id	Number	音乐 ID

### 12.12.3. resumePlayMusic(id)

恢复播放背景音乐

参数：

Name	Type	Description
id	Number	音乐 ID

### 12.12.4. getMusicDurationInMS(path) → {Number}

获取背景音乐文件总时长，单位毫秒

参数：

Name	Type	Description
path	String	音乐文件路径

Returns:

成功返回时长，失败返回-1

Type Number

### 12.12.5. seekMusicToPosInTime(id, pts)

设置背景音乐播放进度

参数：

Name	Type	Description
id	Number	音乐 ID
pts	Number	单位：毫秒

### 12.12.6. setAllMusicVolume(volume)

设置所有背景音乐的本地音量和远端音量的大小

该接口可以设置所有背景音乐的本地音量和远端音量。

- 本地音量：即主播本地可以听到的背景音乐的音量大小。
- 远端音量：即观众端可以听到的背景音乐的音量大小。

注意：如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的

volume 会有爆音的风险，请谨慎操作。

参数：

Name	Type	Description
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 200。

### 12.12.7. setMusicPlayoutVolume(id, volume)

设置背景音乐本地播放音量的大小

播放背景音乐混音时使用，用来控制背景音乐在本地播放时的音量大小。

参数：

Name	Type	Description
id	Number	音乐 ID
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 100；默认值：100

### 12.12.8. setMusicPublishVolume(id, volume)

设置背景音乐远端播放音量的大小

播放背景音乐混音时使用，用来控制背景音乐在远端播放时的音量大小。

参数：

Name	Type	Description
id	Number	音乐 ID
volume	Number	音量大小，100 为正常音量，取值范围为 0 - 100；默认值：100

### 12.12.9. playAudioEffect(effect)

废弃接口：播放音效

每个音效都需要您指定具体的 ID，您可以通过该 ID 对音效的开始、停止、音量等进行设置。若您想同时播放多个音效，请分配不同的 ID 进行播放。因为使用同一个 ID 播放不同音效，SDK 将会停止上一个 ID 对应的音效播放，再启动新的音效播放。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 startPlayMusic 接口替代。

参数：

Name	Type	Description
effect	TRTCAudioEffectParam	音效

Throws:

**String**

### 12.12.10. setAudioEffectVolume(effectId, volume)

废弃接口：设置音效音量

注意：会覆盖通过 setAllAudioEffectsVolume 指定的整体音效音量。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setMusicPublishVolume, setMusicPlayoutVolume 接口替代。

参数：

Name	Type	Description
effectId	Number	音效 ID
volume	Number	音量大小，取值范围为 0 - 100；默认值：100

### 12.12.11. stopAudioEffect(effectId)

废弃接口：停止音效

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 stopPlayMusic 接口替代。

参数：

Name	Type	Description
effectId	Number	音效 ID

### 12.12.12. stopAllAudioEffects()

废弃接口：停止所有音效

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃

### 12.12.13. setAllAudioEffectsVolume(volume)

废弃接口： 设置所有音效的音量

注意：该操作会覆盖通过 setAudioEffectVolume 指定的单独音效音量。

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 setAllMusicVolume 接口替代。

参数：

Name	Type	Description
volume	Number	音量大小，取值范围为 0 - 100；默认值：100

### 12.12.14. pauseAudioEffect(effectId)

废弃接口：暂停音效

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 pausePlayMusic 接口替代。

参数：

Name	Type	Description
effectId	Number	音效 ID

### 12.12.15. resumeAudioEffect(effectId)

废弃接口：恢复音效

Deprecated: 从 TRTCSDK 8.0 后该接口已被废弃，请使用 resumePlayMusic 接口替代。

参数：

Name	Type	Description
effectId	Number	音效 ID

## 12.13. 设备和网络测试

### 12.13.1. startSpeedTest(params)

开始进行网络测速（视频通话期间请勿测试，以免影响通话质量）

测速结果将会用于优化 SDK 接下来的服务器选择策略，因此推荐您在用户首次通话前先进行一次测速，这将有助于我们选择最佳的服务器。同时，如果测试结果非常不理想，您可以通过醒目的 UI 提示用户选择更好的网络。

注意：

- 测速过程将产生少量的基础服务费用，详见 计费概述 > 基础服务 文档说明。
- 请在进入房间前进行网速测试，在房间中网速测试会影响正常的音视频传输效果，而且由于干扰过多，网速测试结果也不准确。
- 同一时间只允许一项网速测试任务运行。

参数：

Name	Type	Description
params	TRTCSpeedTestParams	测速参数

Properties		
Name	Type	Description
sdkAppId	Number	应用标识
userId	String	用户标识
userSig	String	用户签名
expectedUpBandwidth	Number	预期的上行带宽 (kbps, 取值范围: 10 ~ 5000, 为 0 时不测试)。
expectedDownBandwidth	Number	预期的下行带宽 (kbps, 取值范围: 10 ~ 5000, 为 0 时不测试)。

Returns:

接口调用结果，<0：失败

### 12.13.1. stopSpeedTest()

停止网络测速

### 12.13.2. startCameraDeviceTest(view)

开始进行摄像头测试

会触发 onFirstVideoFrame 回调接口

注意：在测试过程中可以使用 setCurrentCameraDevice 接口切换摄像头。

参数：

Name	Type	Description
view	HTMLElement	承载预览画面的 DOM

### 12.13.3. stopCameraDeviceTest()

停止摄像头测试

### 12.13.4. startMicDeviceTest(interval)

开始进行麦克风测试

回调接口 onTestMicVolume 获取测试数据

该方法测试麦克风是否能正常工作，volume 的取值范围为0 - 100。

参数：

Name	Type	Description
interval	Number	反馈音量提示的时间间隔 (ms)，建议设置到大于 200 毫秒

### 12.13.5. stopMicDeviceTest()

停止麦克风测试

### 12.13.6. startSpeakerDeviceTest(testAudioFilePath)

开始进行扬声器测试

回调接口 `onTestSpeakerVolume` 获取测试数据

该方法播放指定的音频文件测试播放设备是否能正常工作。如果能听到声音，说明播放设备能正常工作。

参数：

Name	Type	Description
testAudioFile ePath	String	音频文件的绝对路径，路径字符串使用 UTF-8 编码格式，支持文件格式： WAV、MP3

### 12.13.7. stopSpeakerDeviceTest()

停止扬声器测试

## 12.14. LOG

### 12.14.1. getSDKVersion() → {String}

获取 SDK 版本信息

Returns:

UTF-8 编码的版本号。

Type String

### 12.14.1. setLogLevel(level)

设置 Log 输出级别

参数：

Name	Type	Description
level	TRTCLogLevel	Log 输出等级，默认值：TRTCLogLevelNone TRTCLogLevelNone : 不输出任何 SDK Log TRTCLogLevelVerbose: 输出所有级别的 Log TRTCLogLevelDebug : 输出 DEBUG, INFO, WARNING, ERROR 和 FATAL 级别的 Log TRTCLogLevelInfo : 输出 INFO, WARNING, ERROR 和 FATAL 级别的 Log TRTCLogLevelWarn : 只输出 WARNING, ERROR 和 FATAL 级别的 Log

		TRTCLogLevelError : 只输出 ERROR 和 FATAL 级别的 Log TRTCLogLevelFatal : 只输出 FATAL 级别的 Log
--	--	--

#### 12.14.2. setConsoleEnabled(enabled)

启用或禁用控制台日志打印

参数：

Name	Type	Description
enabled	Boolean	指定是否启用，默认为禁止状态

#### 12.14.3. setLogCompressEnabled(enabled)

启用或禁用 Log 的本地压缩

开启压缩后，Log 存储体积明显减小，但需要腾讯云提供的 Python 脚本解压后才能阅读。禁用压缩后，Log 采用明文存储，可以直接用记事本打开阅读，但占用空间较大。

参数：

Name	Type	Description
enabled	Boolean	指定是否启用，默认为禁止状态

#### 12.14.4. setLogDirPath(path)

设置日志保存路径

注意：

- windows 日志文件默认保存在 C:/Users/[系统用户名]/AppData/Roaming/Tencent/liteav/log，即 %appdata%/Tencent/liteav/log 下，如需修改，必须在所有方法前调用。
- mac 日志文件默认保存在 sandbox Documents/log 下，如需修改，必须在所有方法前调用。

参数：

Name	Type	Description
path	String	存储日志的文件夹，例如 "D:\Log"，UTF-8 编码

#### 12.14.5. setLogCallback(callback)

设置日志回调

参数：

Name	Type	Description
callback	function	日志回调

#### 12.14.6. callExperimentalAPI(jsonStr)

调用实验性 API 接口

注意：该接口用于调用一些实验性功能

参数：

Name	Type	Description
jsonStr	String	接口及参数描述的 JSON 字符串

#### 12.14.7. setRenderMode(mode)

废弃接口：选择绘制的模式 (webgl/yuvcanvs/)

Deprecated: 从 10.3.403 版本该接口被废弃。SDK 内部会根据视频流数据，自动选择合适的渲染方式，可能的渲染方式有 WebGL、Canvas 2D、video 标签。调用该接口不再修改视频的渲染方式。

参数：

Name	Type	Default	Description
mode	Number	1	1 webgl 2 yuvcanvs