

实时音视频

常见问题

产品文档



腾讯云

【版权声明】

©2013-2025 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

常见问题

迁移指南

Twilio Video to TRTC

iOS

Android

Web

计费相关

功能相关

UserSig 相关

应对防火墙限制相关

缩减安装包体积相关

Android 与 iOS 相关

Web 端相关

Flutter 相关

Electron 相关

TRTCCalling Web 相关

音视频质量相关

其他问题

常见问题

迁移指南

Twilio Video to TRTC

iOS

最近更新时间：2023-12-22 15:56:44

概述

欢迎阅读适用于 iOS 平台的 Twilio Video 到 TRTC 的迁移指南。本迁移指南将为应用程序从 Twilio 切换到 TRTC 提供无缝过渡。为了确保顺利的迁移过程，本指南在实现接入基础 RTC 视频功能的每个步骤中，都提供了 Twilio 和 TRTC 之间的 API 比较。

如果您只是计划开发一个新应用程序，而不是进行迁移，您可以直接访问 [TRTC 首页](#) 和 [文档](#) 您将找到开始新项目所需的所有关于 TRTC 的资料。

创建应用

[创建](#)一个新的 TRTC 应用程序，然后您可以获得用于 TRTC 服务认证的以下必要信息。

SDKAppID

SDKAppID 是 TRTC 应用程序的唯一标识。在创建应用程序时会自动生成。

SDKSecretKey

SDKSecretKey 是用于生成安全签名的关键参数之一。生成的签名确保在调用 SDK 的初始化和登录 API 时访问 TRTC 服务。

安装并设置 SDK

环境要求

Xcode 9.0 或更高版本

配备 iOS 9.0 或更高版本的 iPhone 或 iPad

项目的有效开发者签名

安装 SDK

您可以选择使用 CocoaPods 或手动下载并将 SDK 导入到您的项目中。如果使用 CocoaPods，请确保已安装 CocoaPods，并通过以下命令创建 Podfile：

```
sudo gem install cocoapods
pod init
```

编辑创建的 Podfile，并根据您的项目需求填写一个 TRTC 版本：

```
pod 'TXLiteAVSDK_TRTC', 'version'
```

然后通过以下命令更新本地仓库并安装 SDK：

```
pod install
pod update
```

项目配置

要使用 SDK 的音频/视频功能，您需要为应用程序授予麦克风和相机权限。将两个项目（隐私 - 麦克风使用说明和隐私 - 相机使用说明）添加到您的 Xcode 应用程序的 Info.plist 中。

导入 SDK

完成上述步骤后，您可以根据需要通过以下方式将 TRTC SDK 提供的 API 导入到您的项目中。

1. 使用 Objective-C 或 Swift API

```
// Import the SDK module
#import TXLiteAVSDK_TRTC;

// Import the header file
#import "TXLiteAVSDK_TRTC/TRTCCLoud.h"
```

2. 使用 C++ API

```
#include "TXLiteAVSDK_TRTC/cpp_interface/ITRTCCLoud.h"
```

进入房间

Twilio

```
@IBAction func createARoom(sender: AnyObject) {
    let connectOptions = ConnectOptions(token: accessToken) { (builder) in
        builder.roomName = "my-room"
    }
}
```

```
    }
    room = TwilioVideoSDK.connect(options: connectOptions, delegate: self)
  }

  // MARK: RoomDelegate

  func roomDidConnect(room: Room) {
    print("Did connect to Room")

    if let localParticipant = room.localParticipant {
      print("Local identity \\(localParticipant.identity)")

      // Set the delegate of the local participant to receive callbacks
      localParticipant.delegate = self
    }
  }
}
```

TRTC

```
// Create trtc instance (singleton) and set up event listeners
self.trtcCloud = [TRTCCloud sharedInstance];
self.trtcCloud.delegate = self;

// Package the room entry parameter
// Please replace each field in TRTCParams with your own parameters
TRTCParams *params = [[TRTCParams alloc] init];
params.sdkAppId = 1400000123; // Please replace with your own SDKAppID
params.roomId = 123321; // Please replace with your own room number
params.userId = @"denny"; // Please replace with your own userid
params.userSig = @"xxx"; // Please replace with your own userSig
params.role = TRTCRoleAnchor;

// If your application scenario is a video call between several people, please use
[self.trtcCloud enterRoom:params appScene:TRTCAppSceneLIVE];

// Listen to the onEnterRoom event of the SDK and get notification whether the room
- (void)onEnterRoom:(NSInteger)result {
  if (result > 0) {
    [self toastTip:@"Enter room succeed!"];
  } else {
    [self toastTip:@"Enter room failed!"];
  }
}
}
```

发布本地视频/音频

Twilio

```
T// start camera
var cameraSource = CameraSource(delegate: self),
let captureDevice = CameraSource.captureDevice(position: .front)

cameraSource.startCapture(device: captureDevice, completion: nil)

// Use CameraSource to produce video from the device's front camera.

if let camera = TVICameraCapturer(source: .frontCamera),
    let videoTrack = TVILocalVideoTrack(capturer: camera) {

    // TVIVideoView is a TVIVideoRenderer and can be added to any TVIVideoTrack.
    let renderer = TVIVideoView(frame: view.bounds)

    // Add renderer to the video track
    videoTrack.addRenderer(renderer)

    self.localVideoTrack = videoTrack
    self.camera = camera
    self.view.addSubview(renderer)
}
```

TRTC

```
self.trtcCloud = [TRTCCloud sharedInstance];
// Set the preview mode of the local video image: Enable horizontal mirroring and s
TRTCRenderParams *param = [[TRTCRenderParams alloc] init];
param.fillMode = TRTCVideoFillMode_Fill;
param.mirrorType = TRTCVideoMirrorTypeAuto;
[self.trtcCloud setLocalRenderParams:param];

// Enable local camera preview(`localCameraVideoView` is used to render the local v
[self.trtcCloud startLocalPreview:YES view:localCameraVideoView];

// Enable mic and set `quality` to `SPEECH` for the voice mode
[self.trtcCloud startLocalAudio:TRTCAudioQualitySpeech];

[self.trtcCloud startLocalAudio:TRTCAudioQualitySpeech];
```

订阅远程视频/音频

Twilio

```
// MARK: RemoteParticipantDelegate

/*
 * In the Participant Delegate, we can respond when the Participant adds a Video
 * Track by rendering it on screen.
 */
func didSubscribeToVideoTrack(videoTrack: RemoteVideoTrack,
                               publication: RemoteVideoTrackPublication,
                               participant: RemoteParticipant) {

    if let remoteView = VideoView.init(frame: self.view.bounds,
                                         delegate:self) {

        videoTrack.addRenderer(remoteView)
        self.view.addSubview(remoteView)
        self.remoteView = remoteView
    }
}

// MARK: VideoViewDelegate

// Lastly, we can subscribe to important events on the VideoView
func videoViewDimensionsDidChange(view: VideoView, dimensions: CMVideoDimensions) {
    self.view.setNeedsLayout()
}
```

TRTC

```
self.trtcCloud = [TRTCCloud sharedInstance];
// Play back the camera (primary stream) image of `denny`
[self.trtcCloud startRemoteView:@"denny" streamType:TRTCVideoStreamTypeBig view:cam
```

离开房间

Twilio

```
//To disconnect from a room, we call:
room?.disconnect()

//This results in a callback to RoomDelegate#roomDidDisconnect(room:Room, error:Error?)

//MARK: RoomDelegate

func roomDidDisconnect(room:Room, error:Error?) {
```

```
print("Disconnectedfromroom\\(room.name)")
}
```

TRTC

```
self.trtcCloud = [TRTCCloud sharedInstance];
// Exit the current room
[self.trtcCloud exitRoom];

// Listen the `onExitRoom` callback to get notification
- (void)onExitRoom:(NSInteger)reason {
    if (reason == 0) {
        NSLog(@"Exit current room by calling the 'exitRoom' api of sdk ...");
    } else if (reason == 1) {
        NSLog(@"Kicked out of the current room by server through the restful api...");
    } else if (reason == 2) {
        NSLog(@"Current room is dissolved by server through the restful api...");
    }
}
```

总结

通过与 Twilio Video API 的接入对照，本迁移指南概述了如何使用 Tencent RTC (TRTC) 为 iOS 平台构建基本的视频 RTC 体验。本指南中每个步骤的 API 级别“映射”将帮助开发者快速、直接地从 Twilio Video 切换到 TRTC。

TRTC 有更多与音视频的功能及服务，可以帮助开发者实现经济高效、低延迟和高质量的互动音视频服务。有关 TRTC 功能和实现规范的详细信息，请参阅[Tencent RTC 官方网站](#)。如果您需要开发者支持或关于 TRTC 集成的任何进一步帮助，请随时 [联系我们](#)，或者您可以加入我们的 [Discord](#) 和 [Telegram](#)。我们将确保顺利集成并解答您可能遇到的任何问题。

Android

最近更新时间：2023-12-22 15:55:27

概述

欢迎阅读适用于 Android 平台的 Twilio Video 到 TRTC 的迁移指南。本迁移指南将为应用程序从 Twilio 切换到 TRTC 提供无缝过渡。为了确保顺利的迁移过程，本指南在实现接入基础 RTC 视频功能的每个步骤中，都提供了 Twilio 和 TRTC 之间的 API 比较。

如果您只是计划开发一个新应用程序，而不是进行迁移，您可以直接访问 [TRTC 首页](#) 和 [文档](#) 您将找到开始新项目所需的所有关于 TRTC 的资料。

创建应用

[创建](#) 一个新的 TRTC 应用程序，然后您可以获得用于 TRTC 服务认证的以下必要信息。

SDKAppID

SDKAppID 是 TRTC 应用程序的唯一标识。在创建应用程序时会自动生成。

SDKSecretKey

SDKSecretKey 是用于生成安全签名的关键参数之一。生成的签名确保在调用 SDK 的初始化和登录 API 时访问 TRTC 服务。

安装并设置 SDK

环境要求

Android Studio 3.5 or later.

Android 4.1 (SDK API level 16) or later

安装 SDK

TRTC SDK 已发布到 mavenCentral 仓库，您可以按照以下步骤配置 Gradle 来下载 SDK，

1. 将 TRTC SDK 依赖项添加到依赖项中。

```
dependencies {  
    implementation 'com.tencent.liteav:LiteAVSDK_TRTC:latest.release'  
}
```

2. 在 `defaultConfig` 中，指定您的应用程序要使用的 CPU 架构。

```
defaultConfig {
    ndk {
        abiFilters "armeabi-v7a", "arm64-v8a"
    }
}
```

3. 点击 `sync` 以下载 SDK 并将它们集成到您的项目中。

项目配置

在 `AndroidManifest.xml` 中配置应用程序权限。TRTC SDK 需要以下权限。

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera.autofocus" />
```

设置混淆规则

在 `proguard-rules.pro` 文件中，将与 TRTC SDK 相关的类添加到“do not obfuscate”列表中。

```
-keep class com.tencent.** { *;}
```

加入房间

Twilio

```
// Join room
val connectOptions = ConnectOptions.Builder(accessToken)
    .roomName(roomName)
    .audioTracks(localAudioTracks)
    .videoTracks(localVideoTracks)
    .dataTracks(localDataTracks)
    .build()
val room = Video.connect(context, connectOptions, roomListener)
```

TRTC

```
//Initialize TRTC
// Create trtc instance singleton and set up event listeners
val mCloud = TRTCCloud.sharedInstance(getApplicationContext());
mCloud.setListener(this);

// Join room
// Please replace each field in TRTCParams with your own parameters
val params = TRTCParams()
params.sdkAppId = 1400000123 // Please replace with your own SDKAppID
params.userId = "denny" // Please replace with your own userid
params.roomId = 123321 // Please replace with your own room number
params.userSig = "xxx" // Please replace with your own userSig
params.role = TRTCCloudDef.TRTCRoleAnchor

// If your application scenario is a video call between several people, please use
mCloud.enterRoom(params, TRTCCloudDef.TRTC_APP_SCENE_LIVE)

// Listen for the `onEnterRoom` event of the SDK to get the room entry result
// Listen for the `onEnterRoom` event of the SDK and learn whether the room is succ
@Override
public void onEnterRoom(long result) {
    if (result > 0) {
        Log.d(TAG, "Enter room succeed");
    } else {
        Log.d(TAG, "Enter room failed");
    }
}
}
```

发布本地视频/音频

Twilio

```
// Start video
val cameraEnumerator = Camera2Enumerator(context)
cameraEnumerator.deviceNames.firstOrNull { cameraEnumerator.isFrontFacing(it) }?.let {
    val cameraCapturer = Camera2Capturer(context, it)
    val localVideoTrack = LocalVideoTrack.create(context, true, cameraCapturer)
    cameraCapturer.startCapture(width, height, framerate)
}

// Stop video
cameraCapturer.stopCapture()
```

TRTC


```
// Set the preview mode of the local video image: Enable horizontal mirroring and s
TRTCCloudDef.TRTCRenderParams param = new TRTCCloudDef.TRTCRenderParams();
param.fillMode = TRTCCloudDef.TRTC_VIDEO_RENDER_MODE_FILL;
param.mirrorType = TRTCCloudDef.TRTC_VIDEO_MIRROR_TYPE_AUTO;
mCloud.setLocalRenderParams(param);

// Enable local camera preview (`localCameraVideo` is the control used to render th
TXCloudVideoView cameraVideo = findViewById(R.id.txcvv_main_local);
mCloud.startLocalPreview(true, cameraVideo);

// Use `TXDeviceManager` to enable autofocus and turn on the flash
TXDeviceManager manager = mCloud.getDeviceManager();
if (manager.isAutoFocusEnabled()) {
    manager.enableCameraAutoFocus(true);
}
manager.enableCameraTorch(true);

// Enable mic and set `quality` to `SPEECH` for the voice mode
mCloud.startLocalAudio(TRTCCloudDef.TRTC_AUDIO_QUALITY_SPEECH );
```

订阅远程视频/音频

Twilio

```
private lateinit var remoteVideoView: VideoView
private fun setRoomListener() {
    remoteVideoView = findViewById(R.id.remote_video_view)
    val roomListener = object : Room.Listener {
        override fun onConnected(room: Room) {
            room.remoteParticipants.forEach { remoteParticipant ->
                addRemoteParticipant(remoteParticipant)
            }
        }

        override fun onParticipantConnected(room: Room, remoteParticipant: Remo
            addRemoteParticipant(remoteParticipant)
        }
    }
}

private fun addRemoteParticipant(remoteParticipant: RemoteParticipant) {
    remoteParticipant.setListener(object : RemoteParticipant.Listener {
        override fun onVideoTrackSubscribed(
```

```
        remoteParticipant: RemoteParticipant,  
        remoteVideoTrackPublication: RemoteVideoTrackPublication,  
        remoteVideoTrack: RemoteVideoTrack  
    ) {  
        remoteVideoTrack.addSink(remoteVideoView)  
    }  
  
    override fun onVideoTrackUnsubscribed(  
        remoteParticipant: RemoteParticipant,  
        remoteVideoTrackPublication: RemoteVideoTrackPublication,  
        remoteVideoTrack: RemoteVideoTrack  
    ) {  
        remoteVideoTrack.removeSink(remoteVideoView)  
    }  
  
    })  
}
```

TRTC

```
// Play back the camera (primary stream) image of `denny`  
mCloud.startRemoteView("denny", TRTCCloudDef.TRTC_VIDEO_STREAM_TYPE_BIG, cameraView
```

离开房间

Twilio

```
room.disconnect()
```

TRTC

```
// Exit the current room  
mCloud.exitRoom();  
  
// Listen for the `onExitRoom` callback to get the reason for room exit  
@Override  
public void onExitRoom(int reason) {  
    if (reason == 0) {  
        Log.d(TAG, "Exit current room by calling the 'exitRoom' api of sdk ...");  
    } else if (reason == 1) {  
        Log.d(TAG, "Kicked out of the current room by server through the restful ap  
    } else if (reason == 2) {  
        Log.d(TAG, "Current room is dissolved by server through the restful api..."  
    }  
}
```

总结

通过与 Twilio Video API 的接入对照，本迁移指南概述了如何使用 Tencent RTC (TRTC) 为 Android 平台构建基本的视频 RTC 体验。本指南中每个步骤的 API 级别“映射”将帮助开发者快速、直接地从 Twilio Video 切换到 TRTC。TRTC 有更多与音视频的功能及服务，可以帮助开发者实现经济高效、低延迟和高质量的互动音视频服务。有关 TRTC 功能和实现规范的详细信息，请参阅[Tencent RTC 官方网站](#)。如果您需要开发者支持或关于 TRTC 集成的任何进一步帮助，请随时 [联系我们](#)，或者您可以加入我们的 [Discord](#) 和 [Telegram](#)。我们将确保顺利集成并解答您可能遇到的任何问题。

Web

最近更新时间：2023-12-22 15:54:57

概述

欢迎阅读适用于 Web 平台的 Twilio Video 到 TRTC 的迁移指南。本迁移指南将为应用程序从 Twilio 切换到 TRTC 提供无缝过渡。为了确保顺利的迁移过程，本指南在实现接入基础 RTC 视频功能的每个步骤中，都提供了 Twilio 和 TRTC 之间的 API 比较。

如果您只是计划开发一个新应用程序，而不是进行迁移，您可以直接访问 [TRTC 首页](#) 和 [文档](#) 您将找到开始新项目所需的所有关于 TRTC 的资料。

创建应用

[创建](#)一个新的 TRTC 应用程序，然后您可以获得用于 TRTC 服务认证的以下必要信息。

SDKAppID

SDKAppID 是 TRTC 应用程序的唯一标识。在创建应用程序时会自动生成。

SDKSecretKey

SDKSecretKey 是用于生成安全签名的关键参数之一。生成的签名确保在调用 SDK 的初始化和登录 API 时访问 TRTC 服务。

安装并设置 SDK

By npm

1. 在您的项目中使用 npm 安装 trtc-sdk-v5 包。在项目脚本中导入模块。

```
npm install trtc-sdk-v5 --save
```

2. 在项目脚本中导入模块

```
import TRTC from 'trtc-sdk-v5';
```

By script

首先，从 [Github](#) 下载 SDK 文件 trtc.js。然后，在您的网页中添加以下代码：

```
<script src="trtc.js"></script>
```

加入房间

Twilio

```
// Replace Twilio Video import
import * as TwilioVideo from 'twilio-video'

var twilioVideo = TwilioVideo
var twilioRoom

twilioRoom = await twilioVideo.connect(TOKEN, { name: 'yourName', audio: false, vid
```

TRTC

```
const trtc = TRTC.create();

try {
  await trtc.enterRoom({ roomId: 8888, scene:'rtc', sdkAppId, userId, userSig });
  console.log('Entered the room successfully');
} catch (error) {
  console.error('Failed to enter the room ' + error);
}
```

发布本地视频/音频

Twilio

```
// video
let localVideoTrack = await twilioVideo.createLocalVideoTrack({
  height: { ideal: 720, min: 480, max: 1080 },
  width: { ideal: 1280, min: 640, max: 1920 },
  aspectRatio: 16/9,
})

twilioRoom.localParticipant.publishTrack(localVideoTrack)
const localMediaContainer = document.getElementById('video-container-div')
localMediaContainer!.appendChild(localVideoTrack.attach())

// audio
```

```
let localAudioTrack = await twilioVideo.createLocalAudioTrack()

twilioRoom.localParticipant.publishTrack(localAudioTrack);
const audioElement = localAudioTrack.attach();
document.body.appendChild(audioElement);
```

TRTC

```
const view = document.getElementById('localVideo');
const cameraList = await TRTC.getCameraList();
if (cameraList[0]) {
  await trtc.startLocalVideo({
    view,
    option: {
      cameraId: cameraList[0].deviceId,
    }
  });
}

// To preview the camera image, you need to place an HTMLElement in the DOM, which
const view = 'local-video';
await trtc.startLocalVideo({ view });

await trtc.startLocalAudio();
```

订阅远程视频/音频

Twilio

```
twilioRoom.on('participantConnected', (participant) => {
  participant.on('trackSubscribed', (track) => {
    // a user turned on their video, render it
    document.getElementById('remote-video-container-div').appendChild(track.atta
  });

  participant.on('trackUnsubscribed', (track) => {
    // a user turned off their video, stop rendering it
    var selfTwilioVideo = document.getElementById('remote-video-container-div')
    selfTwilioVideo.querySelector('video').remove()
  })
})
```

TRTC

```
// Listen for the TRTC.EVENT.REMOTE_VIDEO_AVAILABLE event before entering the room
trtc.on(TRTC.EVENT.REMOTE_VIDEO_AVAILABLE, ({ userId, streamType }) => {
  // To play the video image, you need to place an HTML element in the DOM, which can
  const view = `${userId}_${streamType}`;
  trtc.startRemoteVideo({ userId, streamType, view });
});
```

离开房间

Twilio

```
twilioVideo.disconnect()
```

TRTC

```
await trtc.exitRoom();
// After the exit is successful, if you do not need to use the trtc instance later,
trtc.destroy();
```

总结

通过与 Twilio Video API 的接入对照，本迁移指南概述了如何使用 Tencent RTC (TRTC) 为 Web 平台构建基本的视频 RTC 体验。本指南中每个步骤的 API 级别“映射”将帮助开发者快速、直接地从 Twilio Video 切换到 TRTC。

TRTC 有更多与音视频的功能及服务，可以帮助开发者实现经济高效、低延迟和高质量的互动音视频服务。有关 TRTC 功能和实现规范的详细信息，请参阅[Tencent RTC 官方网站](#)。如果您需要开发者支持或关于 TRTC 集成的任何进一步帮助，请随时 [联系我们](#)，或者您可以加入我们的 [Discord](#) 和 [Telegram](#)。我们将确保顺利集成并解答您可能遇到的任何问题。

计费相关

最近更新时间：2024-08-21 11:16:24

我的服务用量很大，能打折吗？

如果您的 TRTC 月消耗金额超过 3,000 美金，可以 [联系我们](#)，通过签订合同的方式约定长期折扣。

实时音视频如何收费？

实时音视频收费项根据服务类型划分为基础服务和增值服务两大类，详细计费说明请参见 [计费概述](#)，用户可以每月获得10000分钟 [免费时长](#)。

价格计算器

请查看 [TRTC价格计算器](#)。

如何查看账单及扣费明细？

您可以在 [费用中心 > 账单详情](#) 中查看详细账单及扣费明细。

如何查看/获取计费用量明细？

实时用量：您可以在 TRTC 控制台 > [用量统计](#) 页面直接查看用量曲线图和详细流水数据。单日查看显示每5分钟明细，多日查看显示每日汇总明细。单位精确到分钟。

账单用量：您也可以在腾讯云费用中心 [下载](#) 已出账单对应的用量明细，下载结果为 Excel 文件，内含每5分钟明细和每日明细。单位精确到秒。

注意：

实时用量数据实时变化，与最终结算用量可能会略有差异，**请以账单用量为准**。

如何查看套餐包剩余分钟数？

套餐包采取实时扣除的方式，每5分钟更新一次剩余分钟数，您可以在 [套餐包管理](#) 页面查看套餐包剩余分钟数。

计费精确到秒，余数不足1分钟按1分钟计，使用套餐包会不会多扣我的分钟数？

不会。套餐包扣除分钟数的时候按照当日累计时长计算，不会重复扣除。

套餐包扣除分钟数操作示例如下表所示：

统计区间	区间服务用量	累计服务用量	累计计费时长	区间扣除时长	累计扣除时长
00:00:00 - 00:04:59	30秒	30秒	1分钟	扣除1分钟	扣除1分钟
00:05:00 - 00:09:59	20秒	50秒	1分钟	扣除0分钟	扣除1分钟
00:10:00 - 00:14:59	40秒	90秒	2分钟	扣除1分钟	扣除2分钟

如何预估自己业务的基础服务用量和费用？

若您不知道如何预估自己业务将产生的用量和费用，可以通过 [TRTC 价格计算器](#) 来辅助计算。

视频通话或视频互动直播为什么会产生语音时长？

通常情况下，用户订阅一路音视频流时，既包含音频数据，也包含视频数据。当出现发送端关闭摄像头、接收端关闭视频画面、接收端网络异常、只有1个人在房间等情况时，将导致用户实际接收不到视频画面。为节省您的费用，当用户没有接收到视频画面时，TRTC 将按照语音时长来统计用量。

同一个应用（SDKAppID）订阅多个套餐，免费音视频时长和免费 MAU 额度如何计算？

同一个应用（SDKAppID）内我们的音视频时长和免费 MAU 额度的计算是叠加的。如果您购买了多个产品，如 Chat、Live、Call 或 Conference，音视频时长和免费 MAU 额度都会相加。比如，您同一个应用（SDKAppID）购买了 Live Standard 套餐，其中包含1万的免费 MAU 额度，300,000分钟的免费音视频时长，同时购买了 Call 1v1 套餐，其中包含5000的免费 MAU 额度，100,000分钟的免费音视频时长，则您当月的免费 MAU 额度为15000，免费音视频时长为400,000分钟。

月活跃用户数（MAU）如何计算？

MAU 的计算方式为，调用 IM SDK Login 操作与 IM 后台建立长链接后，MAU 将会加1，即单个用户当月登录 Chat 计为1个 MAU，同一用户重复登录时，MAU 不累加。请根据业务场景合理使用 IM SDK Login 操作，避免出现 MAU 过高的情况。此外，如果在同一个应用（SDKAppID）内，同一个用户同时使用 Chat、Call 或任何其他产品，他们也只会被计算为一个用户数。

屏幕分享如何计费？

屏幕分享，又称屏幕共享，是单独的一路视频流。用户订阅屏幕分享的视频流并且接收到了视频画面时，按照视频时长计费。

CDN 直播观看如何计费？

TRTC 通过旁路推流使用云直播的能力为您提供 CDN 直播观看的功能。[云直播](#)将根据实际使用情况向您收取 CDN 直播观看 > 相关费用。

只有1个人在房间也会计费吗？

只有1个人在房间的时候，即使不推流（不产生上行数据），也会占用 TRTC 的云服务资源。1个人在房间时无法订阅他人的音视频流，因此不会接收到视频画面，按照[语音时长](#)统计服务用量。

为什么我的服务状态显示“已停用”？

腾讯云账户欠费后导致停用：[欠费冲正](#)后将自动恢复服务。

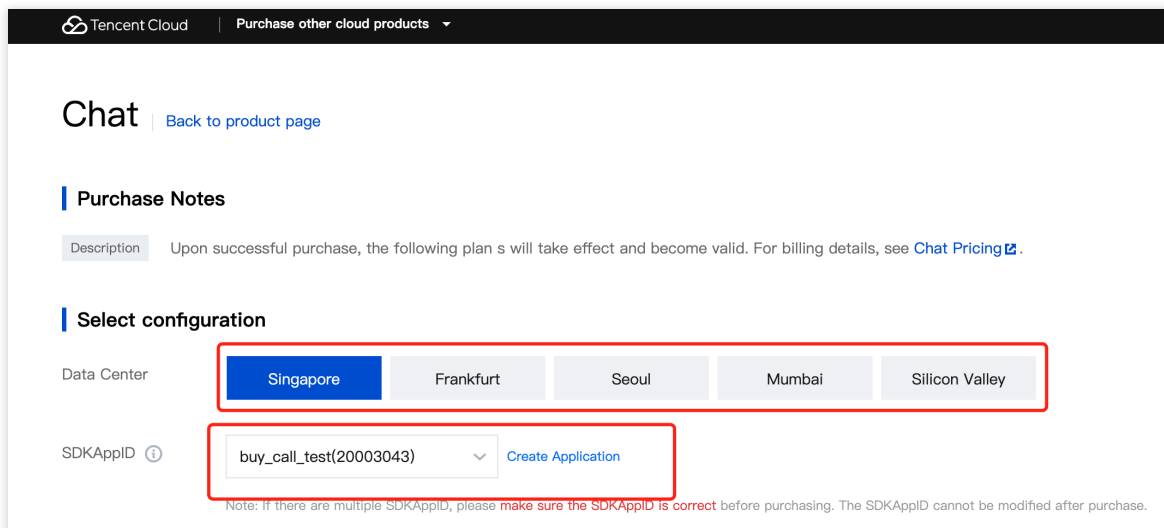
您手动操作将应用停用使其处于停用状态：单击[启用应用](#)即可重新恢复服务。

购买 Call 时，为什么没有找到我想要选择的应用（SDKAppID）？

目前 TRTC 购买页仅支持购买 IM 新加坡数据中心的预付费应用，其他类型应用购买方式如下：

若您的应用是 IM 后付费应用，可参考 [购买指南](#) 进行购买。

若您的应用是 IM 预付费应用，但不是 IM 新加坡数据中心的应用，可前往 [IM 购买页](#) 购买，在页面内选择对应数据中心及应用，随后勾选“音视频通话能力”，选择 Call 版本进行购买。



Plans		Item	Premium Recommended	Standard
Exclusive for Premium edition	Pushing to all users		✓	/
	Concurrent logins on mul...		✓	/
	Search for local messag...		✓	/
	Viewing of historical mes...		✓	/
	Create a community group		✓	/
	Read receipts for group ...		✓	/
Expand ▾				

Audio/video call	<input checked="" type="checkbox"/> Also purchase the Call service (UI components included and Weixin-like call features)
------------------	---

Capabilities		Item	Group call version Support group call and multi-platform call	1-to-1 call version Suitable for 1-to-1 and video call scenarios
	Free call duration		600,000 minutes/month	200,000 minutes/month
	Audio/Video calls		✓	✓
	Complete chat app UI		✓	✓

功能相关

最近更新时间：2024-03-11 17:04:23

实时音视频是什么？

腾讯实时音视频（Tencent Real-Time Communication, TRTC），将腾讯多年来在网络与音视频技术上的深度积累，以多人音视频通话和低延时互动直播两大场景化方案，通过腾讯云服务向开发者开放，致力于帮助开发者快速搭建低成本、低延时、高品质的音视频互动解决方案，详细说明请参见 [产品概述](#)。

实时音视频 Demo 怎么体验？

具体请参见 [Demo 体验](#)。

实时音视频如何快速入门？

实时音视频为您提供了各个平台的 Demo 源码，您只需花费极少的时间即可快速搭建属于您自己的小应用，具体请参见 [新手入门](#)。

实时音视频 RoomID 是什么？取值区间值是多少？

RoomID 即房间号，用于唯一标识一个房间。房间号取值区间为1 - 4294967295，由开发者自行维护和分配。

TRTC 进房 UserID 是什么？取值范围是多少？

UserID 即用户ID，用于在一个实时音视频应用中唯一标识一个用户。取值范围长度建议不超过32字节。请使用英文字母、数字或下划线，区分大小写。

TRTC 房间的生命周期是多久？

第一个加入房间的用户为当前房间的所有者，但该用户无法主动解散房间。

通话模式下：所有用户都主动退房时，后台立即解散房间。

直播模式下：最后一个退房的用户是主播角色时，后台立即解散房间；最后一个退房的用户是观众角色时，后台等待10分钟后解散房间。

如果房间内的单个用户异常掉线，90秒后服务端会将该用户清理出当前房间。如果房间内的所有用户都异常掉线，90秒后服务端会自动解散当前房间。**用户异常掉线等待时长会被纳入计费用时统计。**

当用户要加入的房间不存在时，后台会自动创建一个房间。

TRTC 是否支持不订阅音视频流？

为了实现“秒开”效果，默认是进房自动订阅流，可以通过 `setDefaultStreamRecvMode` 接口切换手动订阅模式。

TRTC 是否支持自定义旁路推流的流 ID？

支持，您可以通过 `enterRoom` 的参数 `TRTCParams` 指定 `streamId`，也可以调用 `startPublishing` 接口传参数 `streamId`。

TRTC 直播支持什么角色？有什么区别？

直播场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）支持 TRTCRoleAnchor（主播）和 TRTCRoleAudience（观众）两种角色，区别是主播角色可以同时上行、下行音视频数据，观众角色只支持下行播放其他人的数据。您可以通过调用 `switchRole()` 进行角色切换。

怎么理解 TRTC 的角色 Role？

只有在直播场景下，可以设置主播和观众角色。主播角色 TRTCRoleAnchor 具有上行和下行音视频的权限，最高并发支持50人，观众 TRTCRoleAudience 只具有下行音视频的权限，最高并发支持10万人。

TRTC 房间支持哪些应用场景？

支持以下场景：

TRTCAppSceneVideoCall：视频通话场景，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。

TRTCAppSceneLIVE：视频互动直播，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。

TRTCAppSceneAudioCall：语音通话场景，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。

TRTCAppSceneVoiceChatRoom：语音互动直播，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。

TRTC 支持哪些平台？

支持的平台包括 iOS、Android、Windows(C++)、Unity、Mac、Web、Electron，更多详情请参见 [平台支持](#)。

TRTC 精简版和全功能版的区别？

详情请参见 [各版本差异对照表](#)。

实时音视频支持直播连麦吗？

支持，具体操作指南请参考：

[跑通直播模式\(iOS&Mac\)](#)

[跑通直播模式\(Android\)](#)

[跑通直播模式\(Windows\)](#)

[跑通直播模式\(Electron\)](#)

[跑通直播模式\(Web\)](#)

实时音视频最多可以同时创建多少个房间？

支持同时并发存在4294967294个房间，累计房间数量无限制。

如何创建房间？

房间是由腾讯云后台在客户端进房时自动创建的，您无需手动创建房间，只需调用客户端的相关接口“进入房间”即可：

[iOS & Mac > enterRoom](#)

[Android > enterRoom](#)

[Windows \(C++\) > enterRoom](#)

[Windows \(C#\) > enterRoom](#)

[Electron > enterRoom](#)

[Web > join](#)

实时音视频的视频服务端最大支持带宽是多少？

没有限制。

实时音视频是否支持私有化部署？

实时音视频私有化部署未完全开放。若您需咨询或使用私有化服务，请联系：colleenyu@tencent.com。

实时音视频开通旁路直播，域名是否需要进行备案使用？

如需开通旁路直播，依据国家相关部门要求，播放域名需要备案才能使用。

实时音视频延时大约多少？

全球端到端平均延时小于300ms。

实时音视频是否支持主动呼叫功能？

需要结合信令通道解决，例如使用 [即时通信 IM](#) 服务的自定义消息实现呼叫，可以参考 [SDK](#) 源码中的场景化 Demo 示例。

实时音视频双人视频通话是否支持蓝牙耳机？

支持。

实时音视频接入 PC 端是否支持屏幕分享功能？

支持，您可以参考如下文档：

[屏幕分享 \(Windows\)](#)

[屏幕分享 \(Mac\)](#)

[屏幕分享 \(Web\)](#)

屏幕分享接口详情请参见 [Windows \(C++\) API](#)。另外，您也可以使用 [Electron 接口](#)。

是否支持将本地视频文件分享到实时音视频中？

支持，可以通过 [自定义采集](#) 功能来实现。

实时音视频能否把直播视频录制后存储在手机本地？

不支持直接存储在手机本地，录制后视频文件默认存储在云点播平台，您可以自行下载并保存到手机中。

实时音视频是否支持单纯的实时音频？

支持纯音频。

一个房间里可以同时有多路屏幕分享吗？

目前一个房间只能有一路辅流的屏幕分享。

指定窗口分享（SourceTypeWindow），当窗口大小变化时，视频流的分辨率会不会也跟着变化？

默认情况下，SDK 内部会自动根据分享的窗口大小进行编码参数的调整。

如需固定分辨率，需调用 `setSubStreamEncoderParam` 接口设置屏幕分享的编码参数，或在调用 `startScreenCapture` 时指定对应的编码参数。

TRTC 是否支持1080P？

支持，您可以通过 SDK 的视频编码参数 `setVideoEncoderParam` 对分辨率进行设置。

TRTC 是否可以自定义采集数据？

部分平台支持，详细信息请参见 [自定义采集和渲染](#)。

TRTC 可以跟互动直播 SDK 通信吗？

不可以。

TRTC 是否可以与移动直播通信？

TRTC 与移动直播后台方案架构不同，所以不支持直接相互通信，仅可从 TRTC 后台旁路推流到 CDN。

TRTC 进房模式 AppScene 有什么区别？

TRTC 支持四种不同的进房模式，其中视频通话（VideoCall）和语音通话（VoiceCall）统称为通话模式，视频互动直播（Live）和语音互动直播（VoiceChatRoom）统称为直播模式。

通话模式下的 TRTC，支持单个房间最多300人同时在线，支持最多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、远程面试、视频客服、在线狼人杀等应用场景。

直播模式下的 TRTC，支持单个房间最多10万人同时在线，具备小于300ms的连麦延迟和小于1000ms的观看延迟，以及平滑上下麦切换技术。适用低延时互动直播、十万人互动课堂、视频相亲、在线教育、远程培训、超大型会议等应用场景。

TRTC 是否支持音视频通话免提模式？

支持，免提模式通过设置音频路由实现，Native SDK 通过 `setAudioRoute` 接口切换。

TRTC 是否支持音量大小提示？

支持，通过 `enableAudioVolumeEvaluation` 接口启用。

TRTC 是否支持设置镜像画面？

支持，通过 `setLocalViewMirror` 接口设置本地摄像头预览画面的镜像模式，或通过 `setVideoEncoderMirror` 接口设置编码器输出的画面镜像模式。

TRTC 是否支持录制通话过程中的音频到本地文件？

支持，通过 `startAudioRecording` 接口可以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，目前支持的音频格式有 PCM, WAV, AAC。

TRTC 是否支持音视频互通过程中的视频录制成文件？

支持自有服务端录制（即录音/录像），如需使用请 [提工单](#) 联系我们获取 SDK 及相关指引。您也可以使用 [实现云端录制与回放](#) 录制视频。

TRTC 是否支持类似微信视频通话的悬浮窗、大小画面切换等功能？

此类功能属于 UI 布局逻辑，SDK 并不限制 UI 上的展示处理。在官方 Demo 中提供了画面前后堆叠和九宫格布局模式的示例代码，并且支持悬浮窗、大小画面切换和画面拖动，更多详情请参考 [官方 Demo](#)。

TRTC 怎么实现纯音频通话？

TRTC 没有音频和视频通道的区分，只调用 `startLocalAudio` 不调用 `startLocalPreview` 时即为纯音频通话模式。

TRTC 纯音频通话怎么实现旁路推流和录制？

6.9以前版本：进房时需要构造 `json{"Str_uc_params":{"pure_audio_push_mod":1}}` 传入 `TRTCParams.businessInfo` 中，1表示旁路推流 2表示旁路推流+录制。

TRTC SDK 6.9及以后版本：在进房时选择场景参数为 `TRTCAppSceneAudioCall` 或 `TRTCAppSceneVoiceChatRoom` 即可。

TRTC 房间支不支持踢人、禁止发言、静音？

支持。

如果是简单的信令操作，可以使用 TRTC 的自定义信令接口 `sendCustomCmdMsg`，开发者自己定义相应的控制信令，收到控制信令的通话方执行对应操作即可。例如，踢人就是定义一个踢人的信令，收到此信令的用户就自行退出房间。

如果是需要实现更完善的操作逻辑，建议开发者通过 [即时通信 IM](#) 来实现相关逻辑，将 TRTC 的房间与 IM 群组进行映射，在 IM 群组中收发自定义消息来实现相应的操作。

TRTC 支持拉流播放 RTMP/FLV 流吗？

支持，目前 TRTC SDK 中已打包 `TXLivePlayer`。如果有更多播放器功能需求，可以直接使用 `LiteAVSDK_Professional` 版本，包含了全部功能。

实时音视频最多可以支持多少个人同时通话？

通话模式下，单个房间最多支持300人同时在线，最多支持50人同时开启摄像头或麦克风。

直播模式下，单个房间支持10万人以观众身份在线观看，最多支持50人以主播身份开启摄像头或麦克风。

TRTC 怎么实现直播场景类应用？

TRTC 专门针对在线直播场景推出了10万人低延时互动直播解决方案，能保证主播与连麦主播的最低延时到200ms，普通观众的延时在1s以内，并且超强的抗弱网能力适应移动端复杂的网络环境。

具体操作指引请参考 [跑通直播模式](#)。

能用 TRTC 发送自定义消息接口实现聊天室、弹幕等功能吗？

不能，TRTC 发送自定义消息适用于简单低频的信令传输场景，具体限制参见 [使用限制](#)。

TRTC SDK 播放背景音是否支持循环播放？是否支持调整背景音的播放进度？

支持，循环播放可以在完成回调里面重新调用播放，播放进度可以通过 TXAudioEffectManager seekMusicToPosInMS 设置。

说明：

setBGMPosition() 在 v7.3 版本废弃，通过 TXAudioEffectManager seekMusicToPosInMS 替代。

TRTC 有没有房间成员进出房间的监听回调？onUserEnter/onUserExit 是否可以用？

有，TRTC 使用 onRemoteUserEnterRoom/onRemoteUserLeaveRoom 监听房间成员进出房间（有上行音视频权限的用户才会触发）。

说明：

onUserEnter/onUserExit 在 6.8 版本废弃，通过 onRemoteUserEnterRoom/onRemoteUserLeaveRoom 替代。

TRTC 怎么监测断网和重连？

通过以下监听回调监听：

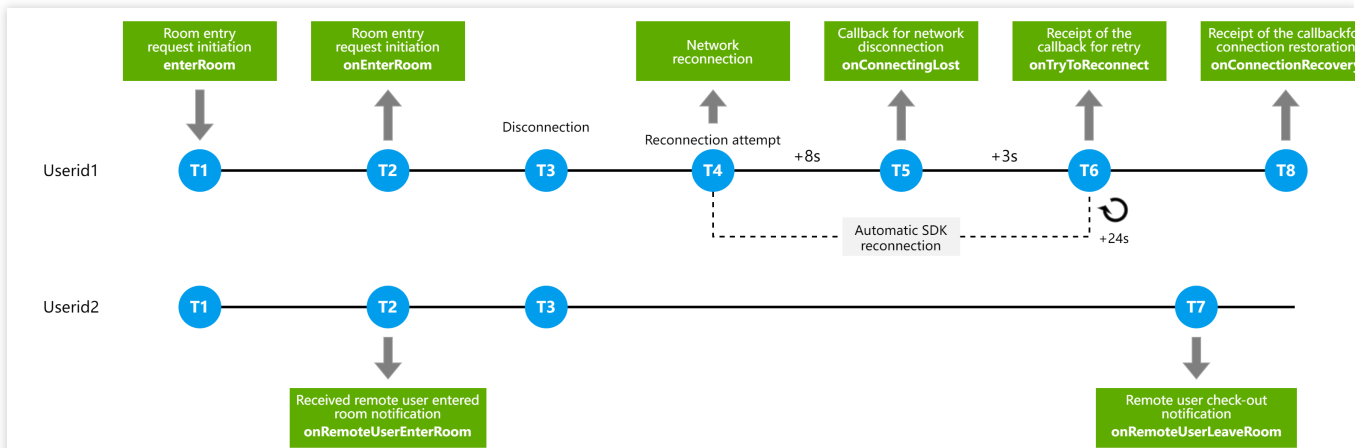
onConnectionLost：SDK 跟服务器的连接断开。

onTryToReconnect：SDK 尝试重新连接到服务器。

onConnectionRecovery：SDK 跟服务器的连接恢复。

TRTC SDK 是否支持断线重连？

SDK 支持用户断线情况下自动重连（若持续30分钟都未重连成功，则自动退房并返回-3301错误码），连接过程中具体的连接状态和处理逻辑如下说明。下图展示了从用户 Userid1 加入频道，到连接中断，再到重新加入房间过程中，收到的监听回调事件：



具体说明：

- T1：用户侧发起调用 `enterRoom` 接口发起进房请求。
- T2：用户 Userid1 收到 `onEnterRoom` 回调，Userid2 感知 Userid1 存在延迟，大约300ms后，Userid2 收到 `onRemoteUserEnterRoom` 回调。
- T3：Userid1 客户端因网络问题断网，SDK 会尝试重新加入房间。
- T4：Userid1 如果连续8秒没有连接上服务端，Userid1 收到 `onConnectionLost` 断连回调。
- T5：Userid1 接着隔3秒没有连接上服务端，Userid1 收到 `onTryToReconnect` 重试回调。
- T6：Userid1 接着每隔24秒，收到 `onTryToReconnect` 重试回调。
- T7：Userid2 会在收到 Userid1 掉线通知90s后，SDK 判断远端用户 Userid1 掉线，Userid2 收到 `onRemoteUserLeaveRoom` 回调。
- T8：如果 Userid1 断连期间任意时刻重连成功，Userid1 收到 `onConnectionRecovery` 恢复回调。

TRTC 有没有首帧渲染回调？能否监听画面开始渲染，声音开始播放？

支持，通过 `onFirstVideoFrame/onFirstAudioFrame` 可以监听。

TRTC 是否支持视频画面截图功能？

目前在 iOS/Android 端调用 `snapshotVideo()` 支持本地及远端视频画面截图。

TRTC 接入蓝牙耳机等外设异常？

目前 TRTC 有对主流的蓝牙耳机和外设做兼容，但是还会遇到某些设备上有兼容问题。建议使用官方 Demo 以及QQ 音视频通话测试对比下是否都正常。

TRTC 音视频过程中的上下行码率、分辨率、丢包率、音频采样率等信息怎么获取到？

可以通过 SDK 接口 `onStatistics()` 获取到这些统计信息。

TRTC 播放背景音乐接口 `playBGM()` 是否支持在线音乐？

目前只支持本地音乐，可以先下载至本地再调用 `playBGM()` 播放。

TRTC 是否支持设置本地采集音量？是否支持设置每个远端用户的播放音量？

支持，通过 `setAudioCaptureVolume()` 接口可以设置 SDK 的采集音量，通过 `setRemoteAudioVolume()` 接口可以设置某个远程用户的播放音量。

stopLocalPreview 和 muteLocalVideo 有什么区别？

`stopLocalPreview` 是停止本地视频采集，调用该接口后自己本地和远端画面都将是黑屏。

`muteLocalVideo` 是设置是否向后台发送自己的视频画面，调用该接口后其他用户观看的画面将变成黑屏，自己本地的预览还是能看到画面。

stopLocalAudio 和 muteLocalAudio 的区别是什么？

`stopLocalAudio` 是关闭本地音频的采集和上行。

`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。

TRTC SDK 都支持哪些分辨率？

建议参考 [设定画面质量](#) 配置分辨率达到更合适的画面质量。

TRTC SDK 怎么设置上行视频码率、分辨率、帧率？

可以通过 TRTCCloud 的 `setVideoEncoderParam()` 接口设置 `TRTCVideoEncParam` 参数中 `videoResolution`（分辨率）、`videoFps`（帧率）、`videoBitrate`（码率）。

SDK 控制画面角度与方向是如何实现的？

详情请参见 [视频画面旋转和缩放](#)。

怎样实现横屏视频通话？

详情请参见 [视频画面旋转和缩放](#)。

TRTC 本地和远端画面方向不一致怎么调整？

详情请参阅 [视频画面旋转和缩放](#)。

TRTC 有没有推荐的画面质量（码率、分辨率、帧率）相关参数配置？

详情请参见 [推荐的配置](#)。

TRTC 是否支持对网络测速？如何操作？

详情请参见 [通话前网络测速](#)。

TRTC 是否支持对房间进行权限校验，例如会员才能进入的场景？

支持，详情请参见 [进房权限保护](#)。

TRTC 音视频流是否支持通过 CDN 拉流观看？

支持。

TRTC 自定义渲染支持哪些格式？

iOS 端支持 i420、NV12 和 BGRA。

Android 端支持 I420 和 texture2d。

实时音视频是什么？

腾讯实时音视频（Tencent Real-Time Communication, TRTC），将腾讯多年来在网络与音视频技术上的深度积累，以多人音视频通话和低延时互动直播两大场景化方案，通过腾讯云服务向开发者开放，致力于帮助开发者快速搭建低成本、低延时、高品质的音视频互动解决方案，详细说明请参见 [产品概述](#)。

实时音视频 Demo 怎么体验？

具体请参见 [Demo 体验](#)。

实时音视频如何快速入门？

实时音视频为您提供了各个平台的 Demo 源码，您只需花费极少的时间即可快速搭建属于您自己的小应用，具体请参见 [新手入门](#)。

实时音视频如何实现云端录制与回放？

具体请参见 [实现云端录制与回放](#)。

TRTC 支不支持美颜？

支持美颜功能。TRTC 提供基于人脸识别技术的 AI 美颜、美妆、微整形、绿幕等各类型多种特效。

Web 端请参见 [开启美颜](#)。

客户端 AI 美颜属于增值服务，由 腾讯特效 SDK 收取相关费用。

说明：

只有全功能版支持美颜特效组件，现阶段 iOS 和 Android 平台有这个功能。

是否支持在非大陆地区使用？

TRTC 支持在国外和香港使用。

说明：

提供覆盖全球的高连通、高可靠、强安全的网络连接通道，自研多重最优寻址算法，具有全网调度能力。丰富的高带宽资源储备，全球节点布局，保证国际链路端到端平均时延 < 300ms。

由于国际链路问题，具体使用时也会受地区与实际使用场景等因素影响。

是否支持实时不良画面检测？

关于涉黄、涉政、敏感信息等，会自动进行拦截关闭，不会允许出现在直播间中。

如何查询当前房间所有用户信息？

您可以通过控制台-监控仪表盘-[实时监控](#) 查看当前在线的房间和用户。

TRTC 可以接收其它的 RTSP 推流吗？

不支持，目前新增支持 RTMP 推流，详情请参见 [RTMP 协议推流接入 TRTC](#)文档。

TRTC 支持双声道编码吗？

支持双声道。

TRTC 推流过程是先封装还是先编码？

TRTC 采集后先进行编码，再进行封装。

TRTC SDK 是 swift 版本吗？

TRTC 目前 Model 层是 OC 版本，UI 层是 Swift 版本。

个人账号可以用 TRTC 吗？

TRTC 支持个人账户使用。

TRTC SDK 支持 IPv6 吗？

支持，建议升级到 10.6 及之上版本，进一步优化了 IPv6 进房速度。

TRTC 包月套餐中的兴趣区域视频编码功能如何使用？

兴趣区域视频编码功能需调用实验性接口（[callExperimentalAPI](#)，以 Android 为例）设置感兴趣区域，示例：

```
//设置 ROI 参数
{
  "api": "UpdateRoiConfig",
  "params": {
    "roiConfigArray": [
      {
        "stream": 1,
        "x": 0,
        "y": 0,
        "width": 100,
        "height": 100,
        "level": 2
      },
      {
        "stream": 1,
        "x": 20,
        "y": 200,
        "width": 100,
        "height": 100,

```

```

        "level":2
    }
]
}
}
    
```

字段名	说明
stream	流类型：0大流，1小流，2辅流。
x,y,w,h	roi 区域坐标点，以编码输出的分辨率为参考。
level	[0, 12]，就是 roi 的强度，值越大，roi 区域的效果更明显，但非 roi 区域可能会更模糊。

UserSig 相关

最近更新时间：2024-10-25 17:49:51

概览

本文档简要介绍了腾讯实时通信服务的两种认证方法，重点介绍 UserSig——腾讯云设计的保护签名，防止恶意攻击者窃取云服务使用权。使用基础云服务时，在 SDK 初始化或登录过程中需提供 SDKAppID、UserID 和 UserSig 三项关键信息。

SDKAppID 用于标识您的应用。

UserID 用于标识您的用户。

UserSig 则是基于前两者计算出的安全签名，它由 **HMAC SHA256** 加密算法计算得出。只要攻击者不能伪造 UserSig，就无法盗用您的云服务流量。

在调试阶段如何计算 UserSig？

您可以通过 [客户端示例代码](#) 和 [控制台](#) 两种方法计算获取 UserSig，具体请参考以下介绍。

不安全：

注意，如下两种 UserSig 获取计算方案仅适用于调试，如果产品要正式上线，**不推荐**采用这种方案，因为客户端代码（尤其是 Web 端）中的 SECRETKEY 很容易被反编译逆向破解。一旦您的密钥泄露，攻击者就可以盗用您的腾讯云流量。

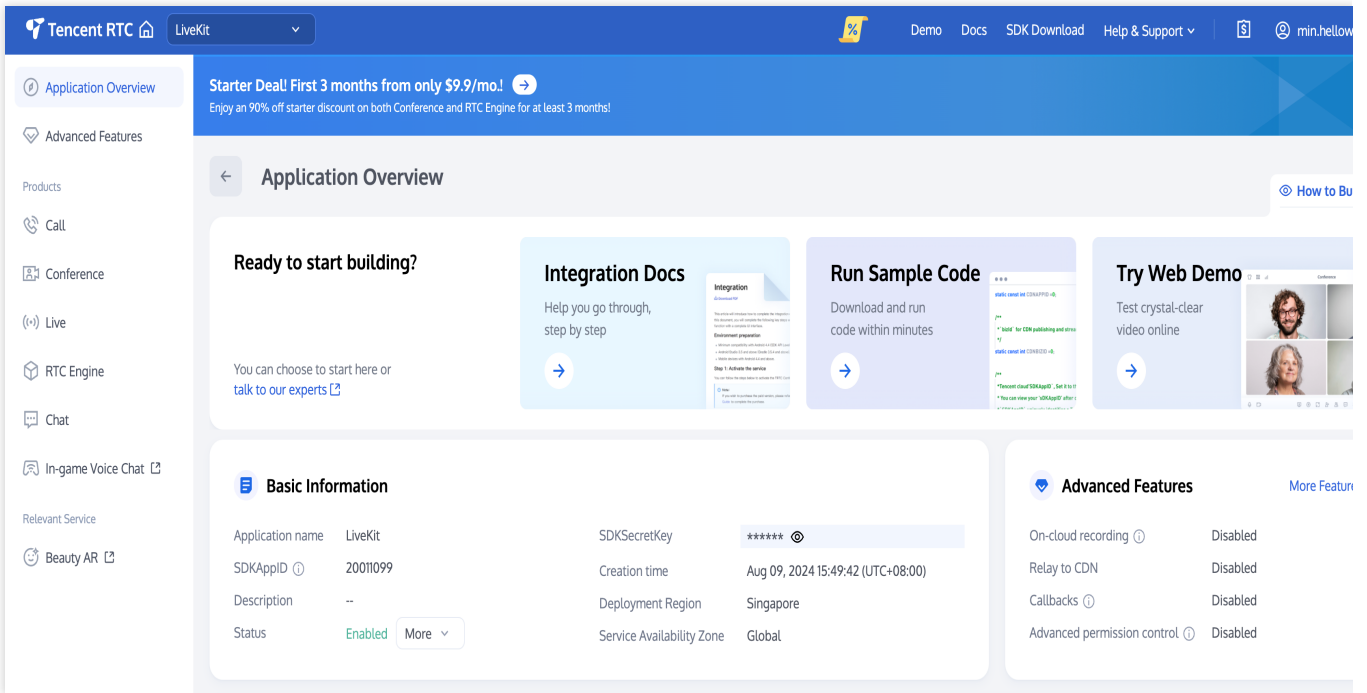
客户端计算 UserSig

1. 获取 SDKAPPID 和密钥：

登录 [Tencent-RTC 控制台](#) > [应用管理](#)。

找到您需查看的 SDKAppID 对应的应用后，点击 **应用名称** 进入详情；

点击 **SDKSecretKey** 显示并复制。



2. 计算 UserSig :

为了方便客户端使用，我们提供各平台计算 UserSig 的源码文件，您可直接下载使用：

Android	iOS	Web	Windows (C++)	Windows (C#)	Flutter	Mac
Github	Github	Github	Github	Github	Github	Github

示例代码如下（当然您也可以参考我们各产品的 Demo 工程，详见各产品的开发文档）：

Android

iOS

Web

Window (C++)

Window (C#)

Flutter

Mac

```
// Step1: 导入源码文件
import com.xxx.xxx.GenerateTestUserSig;

// Step2: 填写上一步骤中获取到的 SDKAppID, SDK 密钥
GenerateTestUserSig.SDKAPPID = xxxxxxx;
GenerateTestUserSig.SECRETKEY = "xxxxxxx";
```



```
// Step3: 根据 userID, 生成 userSig
String userSig = GenerateTestUserSig.genTestUserSig("userID");

// Step1: 导入头文件
#import "GenerateTestUserSig.h"

// Step2: 填写上一步骤中获取到的 SDKAppID, SDK 密钥
[GenerateTestUserSig setSDKAPPID:xxxxxx];
[GenerateTestUserSig setSECRETKEY:@"xxxxxx"];

// Step3: 根据 userID, 生成 userSig
NSString *userSig = [GenerateTestUserSig genTestUserSig:@"userID"];

// Step1: 导入模块
<script src='js/libs/lib-generate-test-usersig.min.js'></script>
<script src='js/libs/generateTestUserSig.js'></script>

// Step2: 填写上一步骤中获取到的 SDKAppID, SDK 密钥、输入自定义的userID, 生成 userSig
const {sdkAppId, userSig } = genTestUserSig({
  sdkAppId: xxxxxx,
  userId: 'xxxxxx',
  sdkSecretKey: 'xxxxxx',
})

// Step1: 导入头文件
#include "GenerateTestUserSig.h"

// Step2: 填写上一步骤中获取到的 SDKAppID, SDK 密钥
const int SDKAPPID = xxxxxx;
const char* SECRETKEY = "xxxxxx";

// Step3: 根据 userID, 生成 userSig
const char* userSig = GenerateTestUserSig::genTestUserSig("userID", SDKAPPID, SECRE

// Step1: 导入头文件
using GenerateTestUserSig;

// Step2: 填写上一步骤中获取到的 SDKAppID 和 SDK 密钥
GenerateTestUserSig.SDKAPPID = xxxxxx;
GenerateTestUserSig.SECRETKEY = "xxxxxx";

// Step3: 根据 userID, 生成 userSig
string userSig = GenerateTestUserSig.GetInstance().GenTestUserSig("userID");
```

```
// Step1: 导入源码文件
import 'package:xxx/GenerateTestUserSig.dart';

// Step2: 填写上一步骤中获取到的 SDKAppID, SDK 密钥
GenerateTestUserSig.SDKAPPID = xxxxxxx;
GenerateTestUserSig.SECRETKEY = "xxxxxxx";

// Step3: 根据 userID, 生成 userSig
String userSig = GenerateTestUserSig.genTestUserSig("userID");

// Step1: 导入头文件
#import "GenerateTestUserSig.h"

// Step2: 填写上一步骤中获取到的 SDKAppID, SDK 密钥
[GenerateTestUserSig setSDKAPPID:xxxxxxx];
[GenerateTestUserSig setSECRETKEY:@"xxxxxxx"];

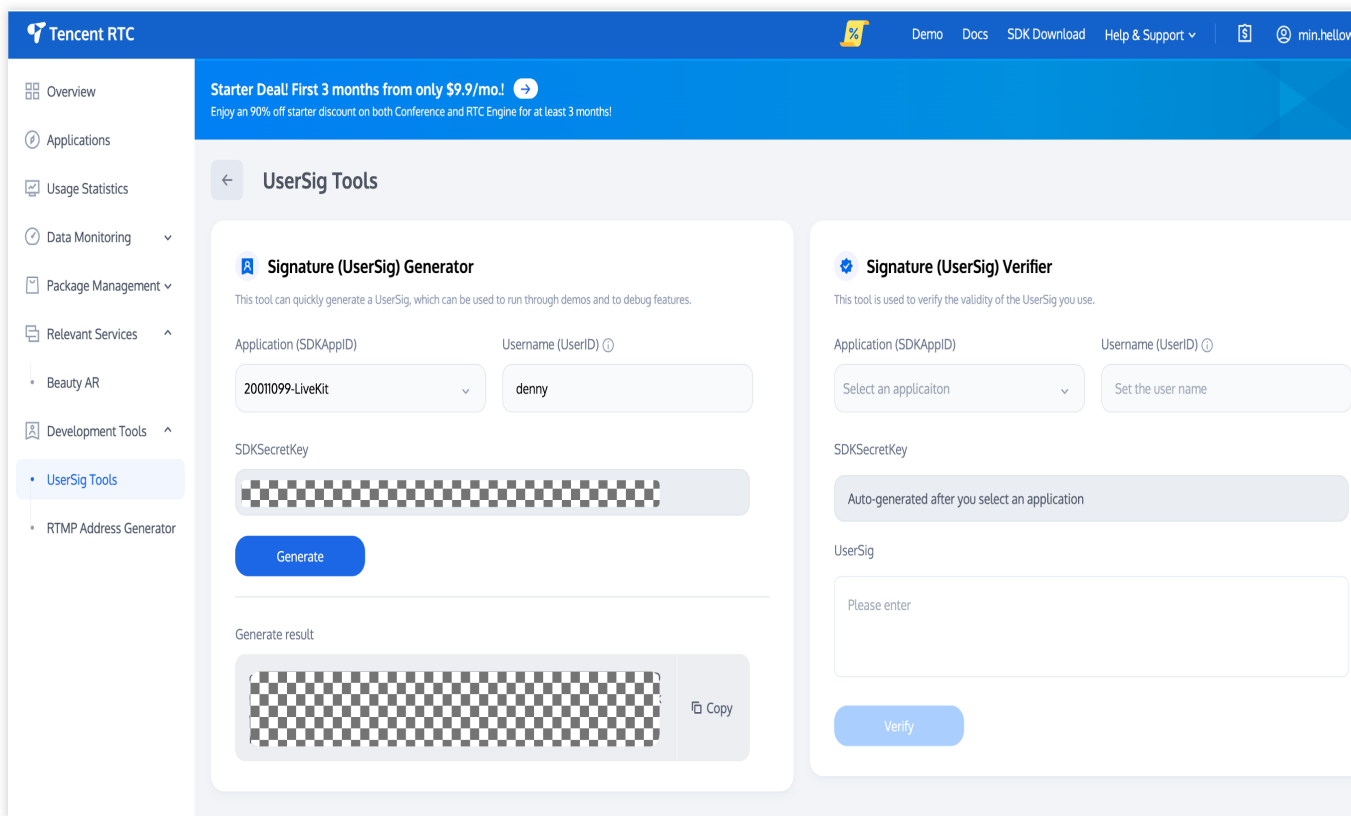
// Step3: 根据 userID, 生成 userSig
NSString *userSig = [GenerateTestUserSig genTestUserSig:@"userID"];
```

控制台获取 UserSig

登录 **Tencent-RTC 控制台**, 进入 Development Tools > [UserSig Tools](#)。

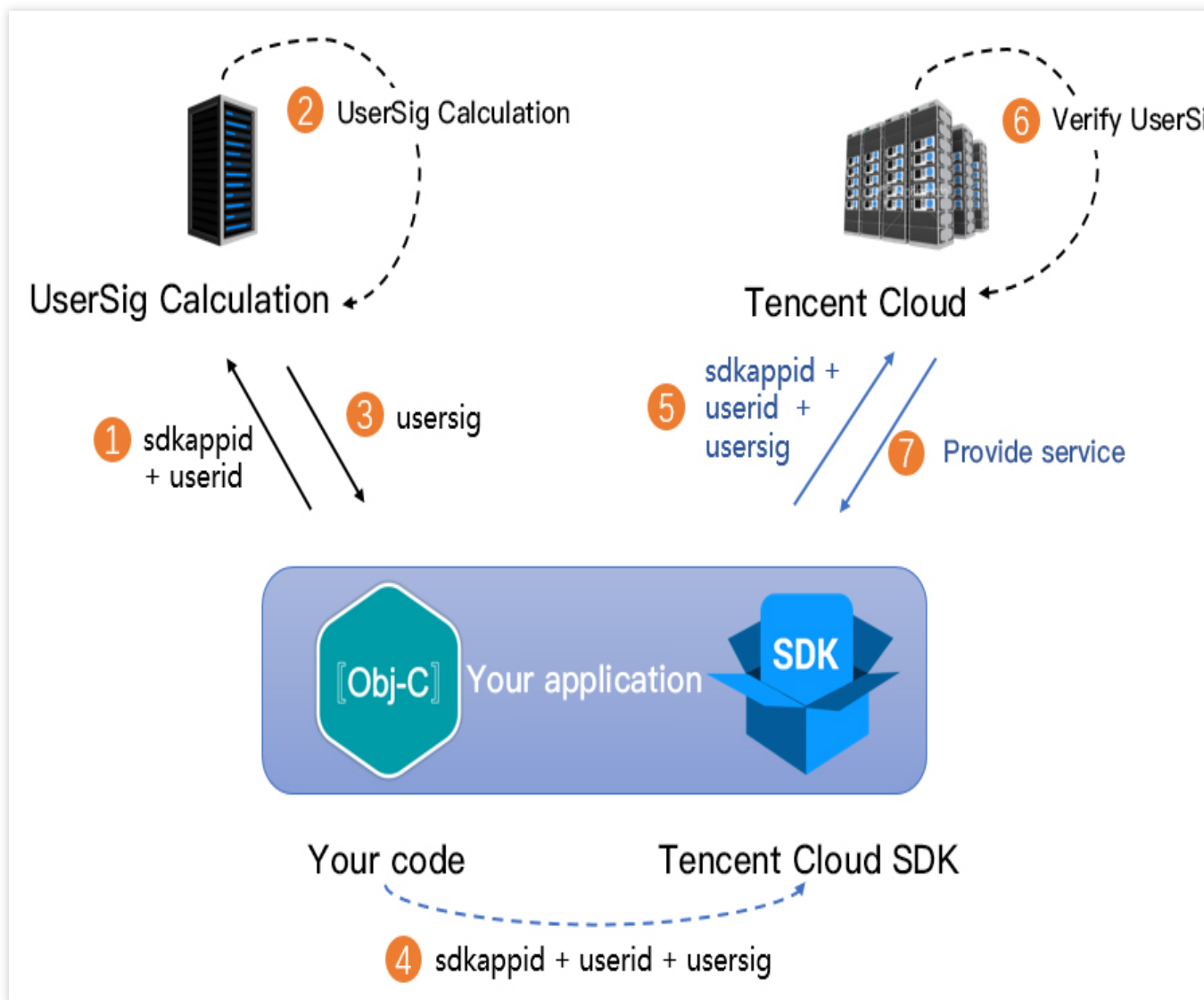
在签名 (UserSig) 生成工具下, 选择对应的 SDKAppID 和 UserID。

单击 **Generate**, 即可计算得到对应的 UserSig。



在正式运行阶段如何计算 UserSig ？

业务正式运行阶段，Tencent-RTC 提供安全等级更高的服务端计算 UserSig 的方案，可以最大限度地保障计算 UserSig 用的密钥不被泄露，因为攻破一台服务器的难度要高于逆向一款 App。具体的实现流程如下：



1. 您的 App 在调用 SDK 的初始化函数之前，首先要向您的服务器请求 UserSig。
2. 您的服务器根据 SDKAppID 和 UserID 计算 UserSig，计算源码见文档前半部分。
3. 服务器将计算好的 UserSig 返回给您的 App。
4. 您的 App 将获得的 UserSig 通过特定 API 传递给 SDK。
5. SDK 将 `SDKAppID + UserID + UserSig` 提交给腾讯云服务器进行校验。
6. 腾讯云校验 UserSig，确认合法性。
7. 校验通过后，会向该 SDK 提供服务。

为了简化您的实现过程，我们提供了多个语言版本的 UserSig 计算源代码及其示例：

语言版本	签名算法	源代码	使用示例
Java	HMAC-SHA256	genSig	Github
GO	HMAC-SHA256	GenSig	Github

PHP	HMAC-SHA256	genSig	Github
Node.js	HMAC-SHA256	genSig	Github
Python	HMAC-SHA256	genSig	Github
C#	HMAC-SHA256	GenSig	Github

应对防火墙限制相关

最近更新时间：2024-02-29 11:03:10

客户端 Native SDK 需要配置哪些端口或域名为白名单？

防火墙端口如下表所示：

TRTC SDK (Native)	白名单项目
TCP 端口	443、20166、10443、10444、10445、10446、10447、10448、10449、10450、10451、13275、23275、33000、37528
UDP 端口	8000、8080、8001、8002、8003、8004、8005、8006、8007、8008、8009、16285、9000

域名白名单：

```
cloud.tim.qq.com
gz.file.myqcloud.com
avc.qcloud.com
yun.tim.qq.com
dldir1.qq.com
mlvbdc.live.qcloud.com
query.tencent-cloud.com
*.trtc.tencent-cloud.com
events.my-imcloud.com
apisgp.my-imcloud.com
mlvbdc.live.tlivesource.com
sdkdc.live.tlivesource.com
*.intl.tencentcos.com
*.tencentcos.cn
```

注意：

因为腾讯云服务端 IP 地址/域名是动态的，并不是固定的一批 IP 地址/域名，所以我们无法提供固定的列表给您。

WebRTC 需要配置哪些端口或域名为白名单？

防火墙端口如下表所示：

WebRTC (H5)	白名单项目
TCP 端口	443
UDP 端口	8000 ; 8080 ; 8800 ; 843 ; 443 ; 16285

域名白名单：

```
signaling.rtc.tencentcloud.com
schedule.rtc.tencentcloud.com
*.rtc.tencentcloud.com
*.rtc-web.com
*.rtc-web.io
```

TRTC web 端内网环境怎么设置代理？

可采用 Nginx+coturn 代理方案，详情请参见 [应对防火墙受限](#)。

方案名	适用场景	网络要求
方案一	允许客户端访问特定的外网代理服务器	允许客户端访问外网的 proxy server
方案二	允许客户端通过内网代理服务器访问外网	允许 proxy server 访问外网

微信小程序需要配置哪些域名为白名单？

<trtc-room> 域名白名单：

```
https://official.opensso.tencent-cloud.com
https://yun.tim.qq.com
https://cloud.tencent.com
https://webim.tim.qq.com
https://query.tencent-cloud.com
https://events.my-imcloud.com
```

缩减安装包体积相关

最近更新时间：2024-06-07 11:20:49

集成 TRTC SDK 后，文件增量有多大？

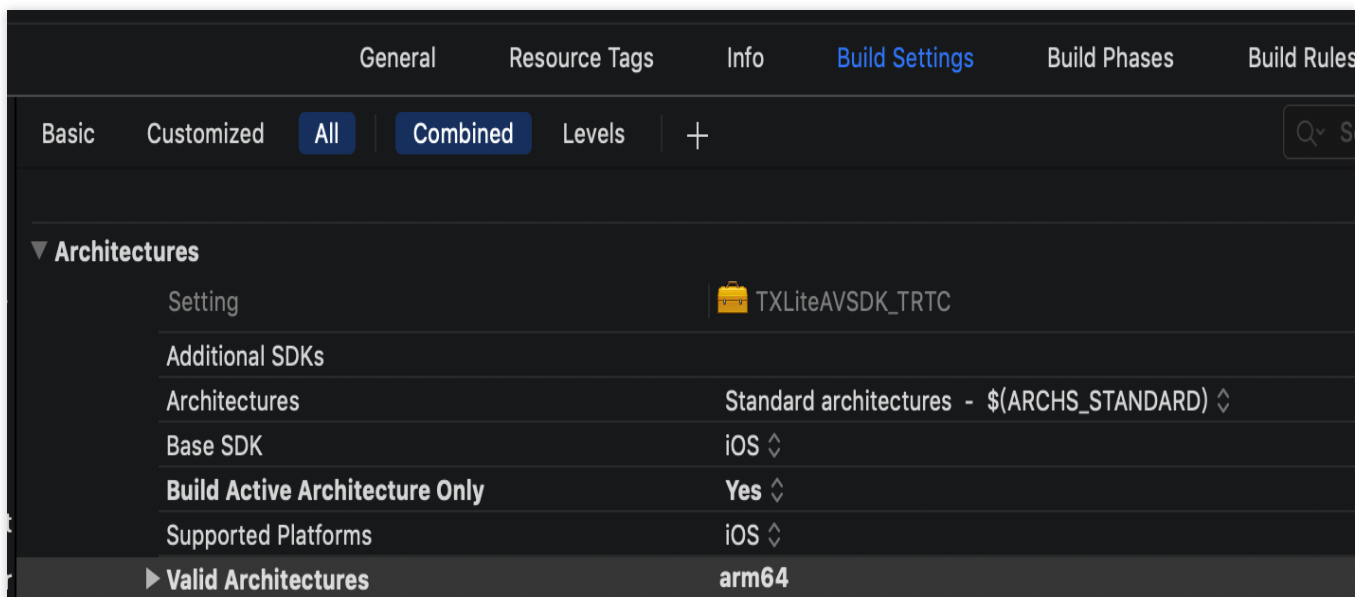
TRTC 各个版本 SDK 的体积增量不同，详情请参见 [SDK 下载](#)。

iOS 平台如何缩减安装包体积？

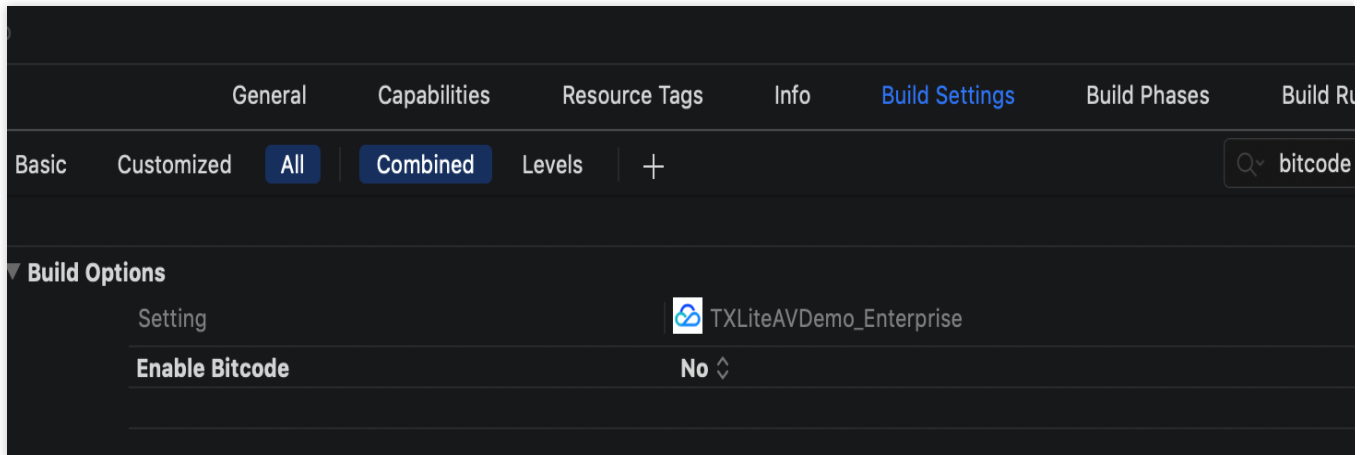
方式一：只打包 arm64 架构（推荐）

方式二：开启 BitCode

苹果 iPhone 5s 以上版本的手机都可以支持只打包 x64 架构，在 XCode 中的 Build Setting 里将 Build Active Architecture Only 设置为 YES，同时将 Valid Architectures 里只写 arm64 即可，TRTC SDK 的单架构 ipa 增量只有 1.9M。



苹果 iPhone 5s 及以前版本的手机，**如果项目中的所有第三方库都支持 BitCode**，您可以通过开启 BitCode 来缩减安装包体积。在 Build Settings > Build Options 中将 Enable Bitcode 选项打开，即可开启 BitCode。



2016年开始，苹果在其 XCode 开发环境中开始支持 BitCode 编译选项，开启 BitCode 之后，编译器会对 App 生成中间代码而非实际的汇编机器码，用户从 App Store 下载安装的是针对具体手机 CPU 架构生成的机器码，所以该方式能够大幅减少安装包体积。

Android 平台如何缩减安装包体积？

方式一：只打包部分 so 文件

方式二：只打包 jar 文件（即安装后下载 so 文件）

如果您的 App 只在中国大陆地区使用，可以只打包 `armeabi-v7a` 架构的 so 文件，从而将安装包增量压缩到5M以内。如果您的 App 希望上架 Google Play，可以打包 `armeabi-v7a` 和 `arm64-v8a` 两个架构的 so 文件。

具体操作方法：在当前项目的 `build.gradle` 里添加 `abiFilters "armeabi-v7a"` 指定打包单架构的 so 文件，或添加 `abiFilters "armeabi-v7a", "arm64-v8a"` 指定打包双架构的 so 文件。

如果只打包 `armeabi-v7a` 架构的 so 文件（即无需上架 Google Play）：

```

android {
    compileSdkVersion rootProject.ext.compileSdkVersion
    buildToolsVersion rootProject.ext.buildToolsVersion
    defaultConfig {
        applicationId "com.tencent.liteav.demo"
        minSdkVersion rootProject.ext.minSdkVersion
        targetSdkVersion rootProject.ext.targetSdkVersion
        versionCode 1
        versionName "2.0"
        multiDexEnabled true

        ndk {
            abiFilters "armeabi-v7a"
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles.add(file('proguard-rules.pro'))
        }
    }
}
    
```

如果打包 `armeabi-v7a` 和 `arm64-v8a` 两个架构的 so 文件（即上架到 Google Play）：

```

android {
    compileSdkVersion rootProject.ext.compileSdkVersion
    buildToolsVersion rootProject.ext.buildToolsVersion
    defaultConfig {
        applicationId "com.tencent.liteav.demo"
        minSdkVersion rootProject.ext.minSdkVersion
        targetSdkVersion rootProject.ext.targetSdkVersion
        versionCode 1
        versionName "2.0"
        multiDexEnabled true

        ndk {
            abiFilters "armeabi-v7a","arm64-v8a"
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles.add(file('proguard-rules.pro'))
        }
    }
}
    
```

注意：

如果您的 App 希望上架 Google Play，请不要使用该方案，有可能会无法上架。

Android 版 SDK 的体积主要来自于 so 文件，如果您希望将安装包增量压缩到1M以内，可以考虑采用安装后再下载 so 的方式：

1. 在 [Github](#) 文件夹下，单击下载类似 `LiteAVSDK_TRTC_x.x.xxx.zip` 命名的压缩包，将其解压并找到指定架构的 so 文件。
2. 上传 [步骤1](#) 中下载到的 so 文件，上传到您的服务器（或腾讯云 [COS](#) 对象存储服务）上，并记录下载地址，例如 `http://xxx.com/so_files.zip`。
3. 在用户启动 SDK 相关功能前，例如开始播放视频之前，先用 loading 动画提示用户“正在加载相关的功能模块”。在用户等待过程中，App 可以到 `http://xxx.com/so_files.zip` 下载 so 文件，并存入应用目录下（例如应用根目录下的 `files` 文件夹），为了确保这个过程不受运营商 DNS 劫持的影响，请在文件下载完成后校验 so 文件的完整性，防止 zip 压缩包被运营商篡改。
4. 等待所有 so 文件就位后，调用 `TXLiveBase` 类（LiteAVSDK 最早的一个基础模块）中的 `setLibraryPath()` 接口，将下载 so 的目标路径设置到 SDK。SDK 会到这些路径下加载需要的 so 文件并启动相关功能。

Andriod 与 iOS 相关

最近更新时间：2024-03-11 17:04:23

移动端（Andriod/iOS）支持哪几种系统音量模式？

支持2种系统音量类型，即通话音量类型和媒体音量类型：

通话音量，手机专门为通话场景设计的音量类型，使用手机自带的回声抵消功能，音质相比媒体音量类型较差，无法通过音量按键将音量调成零，但是支持蓝牙耳机上的麦克风。

媒体音量，手机专门为音乐场景设计的音量类型，音质相比于通话音量类型要好，通过通过音量按键可以将音量调成零。使用媒体音量类型时，如果要开启回声抵消（AEC）功能，SDK 会开启内置的声学处理算法对声音进行二次处理。在媒体音量模式下，蓝牙耳机无法使用自带的麦克风采集声音，只能使用手机上的麦克风进行声音采集。

移动端 SDK 推流怎么设置1080p分辨率？

1080P在 TX_Enum_Type_VideoResolution 定义是114，直接设置分辨率枚举值即可。

在小程序端创建了一个房间，移动端能否进入该房间？

可以，实时音视频支持全平台互通。

TRTC 移动端怎么实现录屏（屏幕分享）？

Android 端：Version 7.2 及以上版本支持手机录屏，具体实践方法请参见 [实时屏幕分享（Android）](#)。

iOS 端：Version 7.2 及以上版本支持 App 内录屏；Version 7.6 及以上版本支持手机录屏和 App 内录屏。具体实践方法请参见 [实时屏幕分享（iOS）](#)。

TRTC Android 端能不能支持64位的 arm64-v8a 架构？

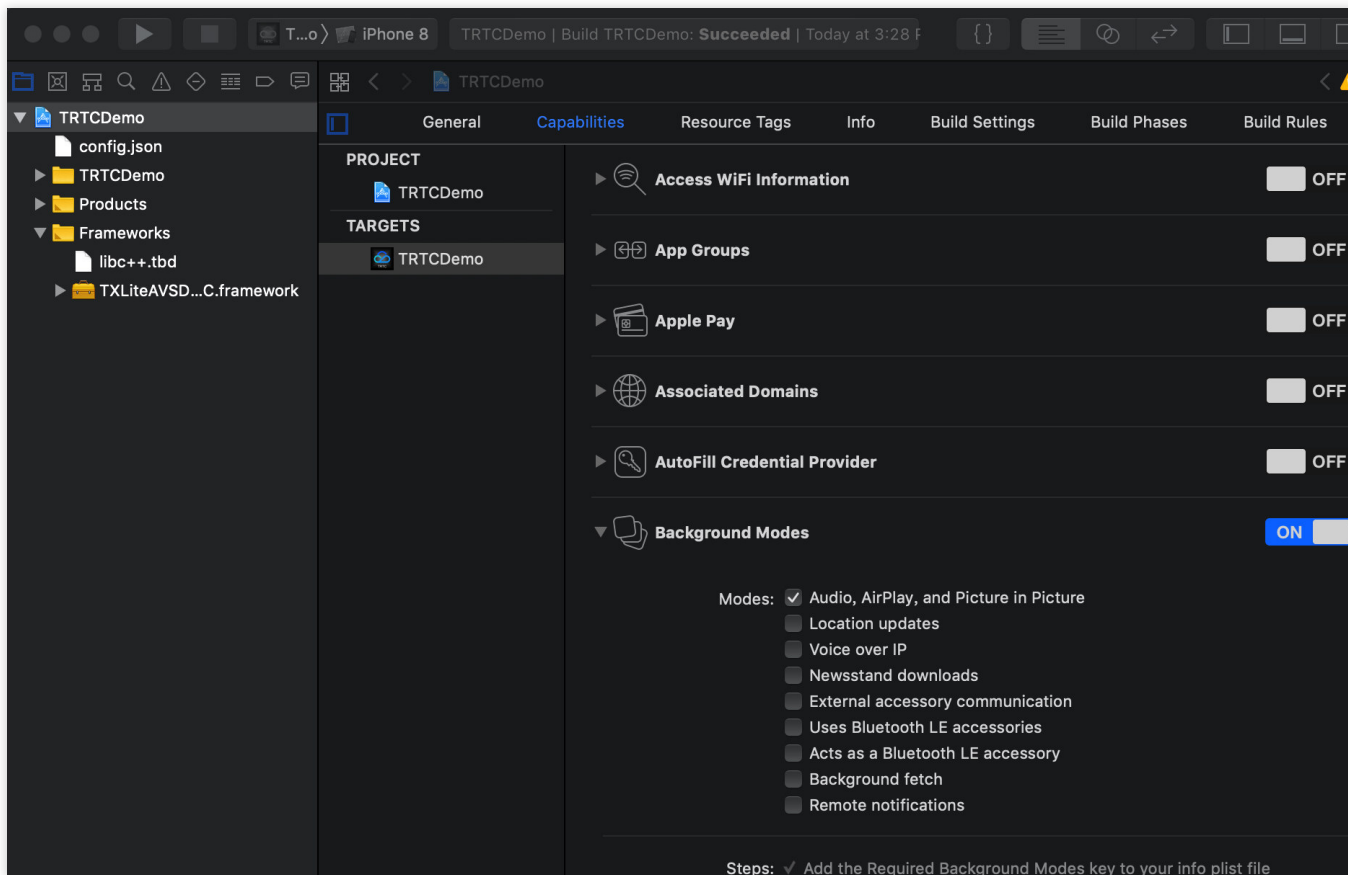
TRTC 6.3 版本开始已提供 arm64-v8a 架构 ABI 支持。

在 iOS 端是否支持 Swift 集成？

支持，直接按照支持集成三方库的流程集成 SDK 即可，还可以参考 [跑通Demo\(iOS&Mac\)](#)。

TRTC SDK 是否支持 iOS 后台运行？

支持，您只需选中当前工程项目，在 **Capabilities** 下的设置 **Background Modes** 为 **ON**，并勾选 **Audio, AirPlay and Picture in Picture**即可实现后台运行，详情如下图所示：



iOS 端是否可以监听远端离开房间？

可以使用 [onRemoteUserLeaveRoom](#) 来监听用户离开房间事件，且该接口仅在 VideoCall 的所有用户和 LIVE 模式下的主播离开房间时会触发回调，观众离开房间不会有回调。

手机锁屏状态、App 在后台或 App 被关闭，音视频如何拨通？

实现离线接听等功能，详情请参见 [实现离线接听](#)。

是否支持 Android 和 Web 端互通？

支持。使用相同的 [SDKAppID](#)，并进入同一个房间进行通话。详情请参见下列文档链接配置 Demo：

[跑通 Demo \(Android\)](#)

[跑通 Demo \(Web\)](#)

主播和粉丝在直播过程中连麦，是否双方都可以主动发起连麦？

双方都可以主动发起，观众和主播发起逻辑一致，具体操作请参见 [跑通直播模式\(Android\)](#)。

多人音视频房间中，移动端和 Web 端是否可以进入同一房间？

可以。需保证 [SDKAppID](#) 和房间号一致，且用户 ID 不一致。

同一个页面中，是否可以创建 N 个 TRTC 对象，通过 N 个 UserID，分别登录到 N 个房间？

可以。[Version 7.6 版本](#) 开始支持一个用户进入多个房间了。

如何查询 SDK 最新版本号？

若您使用自动加载的方法，`latest.release` 为匹配最新版并进行自动加载，不需要对版本号进行修改。具体集成方法请参见 [一分钟集成 SDK](#)。

当前 SDK 最新版本号可通过发布日志查看，具体请参见：

iOS & Android 端，请参见 [发布日志（App）](#)。

Web 端，请参见 [发布日志（Web）](#)。

Electron 端，请参见 [发布日志（Electron）](#)。

Web 端相关

最近更新时间：2024-11-14 11:48:05

一、基础环境问题

Web 端 SDK 支持哪些浏览器？

TRTC Web SDK 对浏览器的详细支持度，请参见 [TRTC Web SDK 对浏览器支持情况](#)。

对于上述没有列出的环境，您可以在当前浏览器打开 [TRTC 能力测试](#) 测试是否完整的支持 WebRTC 的功能。

通话前音视频设备测试？

您可以查看 [通话前环境与设备检测](#)。

如何实时检测当前网络的情况？

具体请参见 [通话前的网络质量检测](#)。

为什么本地开发测试能正常使用 TRTC Web SDK，但是部署到线上无法使用？

出于对用户安全、隐私等问题的考虑，浏览器限制网页只有在安全的环境下（例如 `https`、`localhost`、`file://` 等协议），才能采集麦克风、摄像头。HTTP 协议是不安全的，浏览器会禁止在 HTTP 协议下采集媒体设备。

若您在本本地开发测试一切正常，但是页面部署后，却无法正常使用采集摄像头、麦克风。则请检查您的网页是否部署到了 HTTP 协议上，若是，请使用 HTTPS 部署您的网页，并确保具备合格的 HTTPS 安全证书。

更多详情请参见 [URL 域名及协议限制说明](#)。

是否支持混流、旁路推流、大小流、美颜、水印？

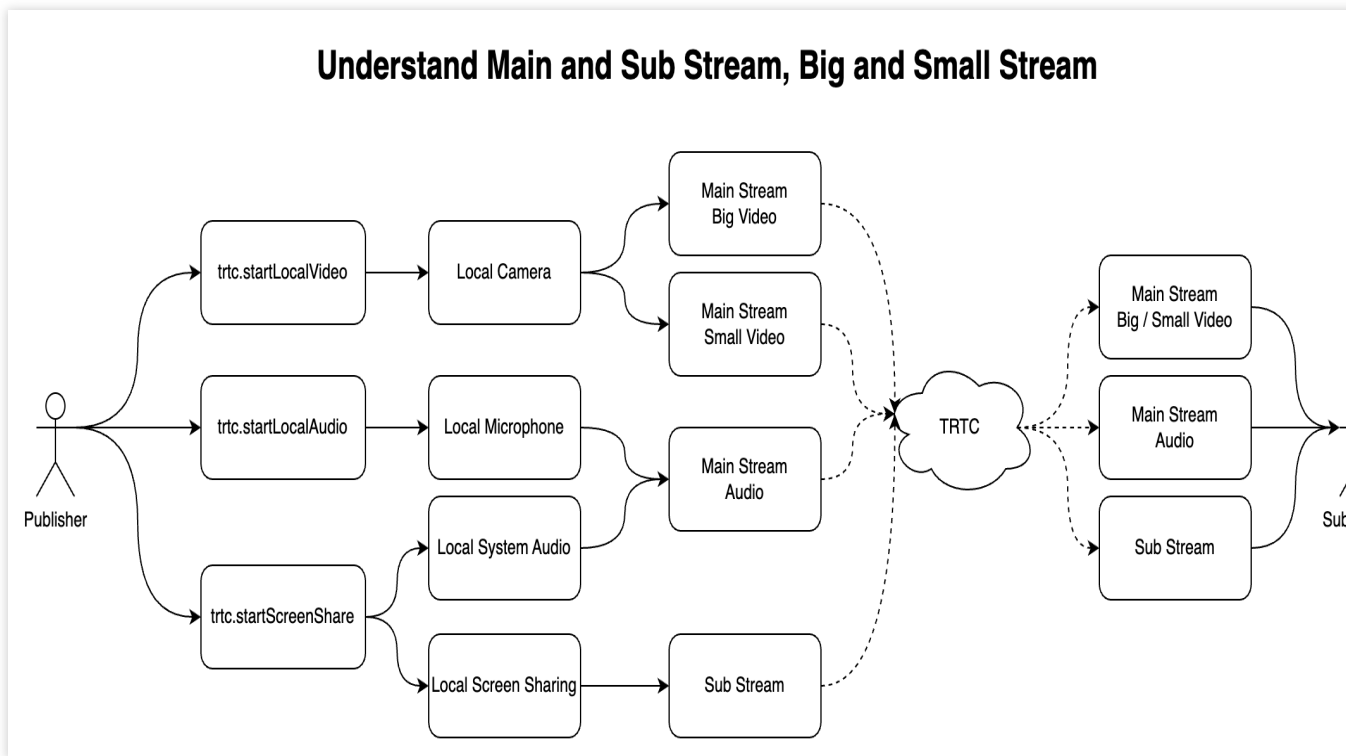
您可请参见 [云端混流、旁路推流、大小流、美颜、水印](#) 文档实现高级功能。

WebRTC 有哪些已知问题？

具体请参见 [WebRTC 已知问题及规避方案](#)。

什么是主流、辅流、大流、小流？

1. TRTC 推流时，有三路流：主流音频、主流视频、辅流视频。
2. 麦克风通过主流音频推流、摄像头通过主流视频推流、屏幕分享通过辅流推流、系统音频通过与麦克风混音后，从主流音频推流。
3. 主流视频包括大流和小流。大小流主要用于在多人视频通话场景下，节省下行带宽。参考：[多人视频通话最佳实践](#)。



二、推拉流问题

Web 端 SDK 日志中报错 **NotFoundError**、**NotAllowedError**、**NotReadableError**、**OverConstrainedError** 以及 **AbortError** 分别是什么意思？

错误名	描述	处理建议
NotFoundError	找不到满足请求参数的媒体类型（包括音频、视频、屏幕分享）。例如：PC 没有摄像头，但是请求浏览器获取视频流，则会报此错误。	建议在通话开始前引导用户检查通话所需的摄像头或麦克风等设备。
NotAllowedError	用户拒绝了当前的浏览器实例的访问音频、视频、屏幕分享请求。	提示用户不授权摄像头/麦克风访问将无法进行音视频通话。
NotReadableError	用户已授权使用相应的设备，但由于操作系统上某个硬件、浏览器或者网页层面发生的错误导致设备无法被访问。	根据浏览器的报错信息处理，并提示用户“暂时无法访问摄像头/麦克风，请确保当前没有其他应用请求访问摄像头/麦克风，并重试”。
OverConstrainedError	cameraId/microphoneId 参数的值无效。	请确保 cameraId/microphoneId 传值正确且有效。

AbortError	由于某些未知原因导致设备无法被使用。	-
------------	--------------------	---

更多详情请参见 [device_error](#)。

部分手机上的浏览器无法正常运行 TRTC 进行推拉流？

TRTC Web SDK 对浏览器的详细支持度，请参见 [TRTC Web SDK 对浏览器支持情况](#)。

对于上述没有列出的环境，您可以在当前浏览器打开 [TRTC 能力测试](#) 测试是否完整的支持 WebRTC 的功能。

Web 端用宽高设置推流的分辨率是所有浏览器都适用吗？

由于设备和浏览器的限制，视频分辨率不一定能够完全匹配，在这种情况下，浏览器会自动调整分辨率使其接近 Profile 对应的分辨率。详情请参见 [VideoProfile](#)。

Web 端支持哪些视频画面分辨率设置？

目前Web端支持设置720p、1080p、2k和4k的视频画面分辨率，更多设置详情请见[Web端画面质量设置](#)。

为什么摄像头支持采集 1080p，而采集出来的分辨率不到 1080p？

这种情况一般是由于摄像头提前被低分辨率采集占用，导致 SDK 无法采集到目标分辨率。

解决方案：检查业务侧逻辑，如果提前采集了摄像头，请在无需使用时，及时释放摄像头采集。

Web 端屏幕分享的样式支持修改吗？

屏幕分享的样式由浏览器控制，目前不能修改。

Web 端支持混流吗？

Web 端支持发起混流，具体请参见[云端混流](#)。

Web 端 SDK 在使用的过程中拔掉摄像头，怎么清除摄像头列表里面的数据？

可以尝试调用 [TRTC.getCameraList](#) 方法是否能获取新的设备列表，如果仍然有拔掉的摄像头信息，说明浏览器底层也没有刷新这个列表，Web 端 SDK 也获取不到新的设备列表信息。

iOS 的微信内嵌浏览器不能正常推流？

请参见 [浏览器支持情况](#)，查看 iOS 上的微信内嵌浏览器对推拉流的支持情况。

三、播放问题

音视频互通过程中出现有画面没有声音问题？

因浏览器自动播放策略限制，音频播放会出现 PLAY_NOT_ALLOWED 异常，此时业务层需要引导用户手动操作 `Stream.resume()` 来恢复音频播放，具体请参见 [自动播放受限处理建议](#)。
未知异常导致，请通过监控仪表盘查询收发两端的 `audioLevel` & `audioEnergy`。

Web 通话画面显示不了？

检查一下 Web 页面上是否有获取到数据，在确认数据收发正常时，可以检查 `<video>` 元素的 `srcObject` 属性是否赋值了正确的 `mediaStream` 对象，如果赋值错误，肯定显示不了。

Web 通话过程中出现回声、杂音、噪声、声音小？

通话双方的设备相距太近的时候，属于正常现象，测试时请相互距离远一点。当其他端听到 Web 端的声音存在回声、噪声、杂音等情况时，说明 Web 端的 3A 处理没有生效。

若您使用了浏览器原生 `getUserMedia` API 进行自定义采集，则需要手动设置 3A 参数：

`echoCancellation`：回声消除开关

`noiseSuppression`：噪声抑制开关

`autoGainControl`：自动增益开关，详细设置请参见 [媒体追踪约束](#)。

Android 端采集麦克风后，扬声器声音变成从听筒播放了？

Android 端一般会有多个麦克风，`label` 列表为：`['default', 'Speakerphone', 'Headset earpiece']`，如果在 `trtc.startLocalAudio` 时，不指定麦克风，浏览器默认麦克风可能会采集到 `Headset earpiece`，此时声音会从听筒放出。若需要通过扬声器外放，则需要指定采集 `label` 为 `Speakerphone` 的麦克风。

四、其他

Web 端 SDK 可以获取当前音量大小吗？

可以通过 `enableAudioVolumeEvaluation` 获取当前音量大小，具体请参见 [音量大小检测](#)。

什么情况会触发 `trtc.on(TRTC.EVENT.KICKED_OUT)`？

由于某种原因被踢出房间，包括 `kick`：相同用户进房，`banned`：被管理员踢出，`room_disband`：房间被解散。更多具体详情，请参见 [KICKED_OUT 事件](#)。

注意：

同名用户同时登录是不允许的行为，可能会导致双方通话异常，业务层应避免出现同名用户同时登录。

Web 端如何在屏幕分享的时候采集系统声音？

具体操作请参见 [屏幕分享采集系统声音](#)。

目前采集系统声音只支持 Chrome M74+，在 Windows 和 Chrome OS 上，可以捕获整个系统的音频，在 Linux 和 Mac 上，只能捕获选项卡的音频。其它 Chrome 版本、其它系统、其它浏览器均不支持。

Web 端如何切换摄像头和麦克风？

您可以先获取到系统的摄像头和麦克风设备后，调用 [updateLocalVideo](#) 来进行切换，具体操作请参见 [切换摄像头和麦克风](#)。

在 iframe 使用 TRTC Web SDK 报错 Permission denied？

在 iframe 中使用 WebRTC 需要给 iframe 标签增加属性来开启相关权限，具体参考如下。

麦克风、摄像头、屏幕分享权限：

```
<iframe allow="microphone; camera; display-capture;">
```

Vue 3 使用注意事项

在 Vue3 框架使用时，需要注意使用 [markRaw](#) 接口标记 trtc 实例，避免 Vue 将 trtc 实例转成 Proxy 对象，导致不可预知的问题。

```
import { markRaw } from 'vue';  
const trtc = markRaw(TRTC.create());
```

多人音视频通话，设备发烫怎么办？

参见：[多人音视频通话最佳实践](#)。

Flutter 相关

最近更新时间：2025-03-11 17:29:40

两台手机同时运行 Demo，为什么看不到彼此的画面？

请确保两台手机在运行 Demo 时使用的是不同的 UserID，TRTC 不支持同一个 UserID（除非 SDKAppID 不同）在两个终端同时使用。

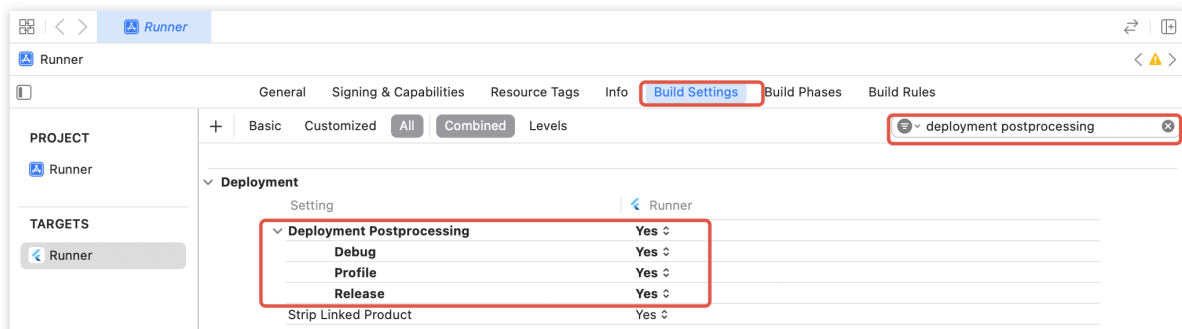
防火墙有什么限制？

由于 SDK 使用 UDP 协议进行音视频传输，所以在对 UDP 有拦截的办公网络下无法使用。如遇到类似问题，请参见[应对公司防火墙限制](#) 排查并解决。

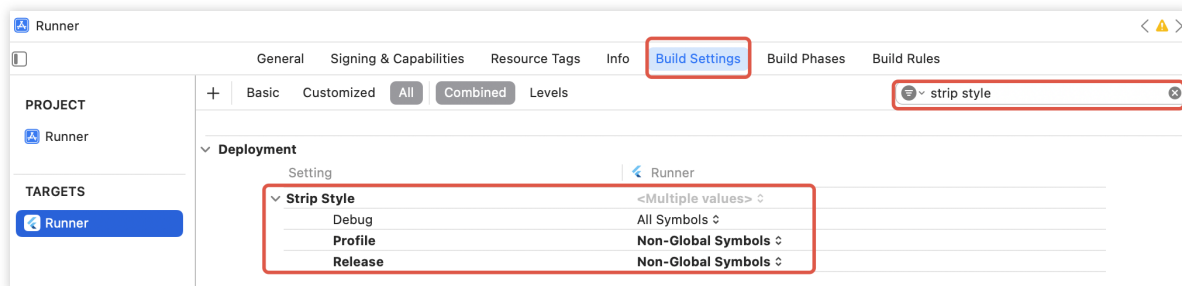
iOS release 包运行时 [symbol not found]？

由于 tencent_rtc_sdk 通过 Flutter FFI 调用接口，iOS Release 构建时 Xcode 的符号裁剪优化可能误移除 TRTC 的 C 符号，引发 `symbol not found` 错误。解决方案如下：

1. 在项目的 Build Settings 中找到 `deployment postprocessing`，将其设置为 **Yes**。



2. 在项目的 Build Settings 中找到 `strip style`，将其中 Release 的值设置为 **Non-Global Symbols**。



iOS 打包运行 Crash？

请排查是否 iOS14 以上的 debug 模式问题，具体请参见 [官方说明](#)。

iOS 无法显示视频（Android 正常）？

请确认在您项目的 `info.plist` 中 `io.flutter.embedded_views_preview` 值为 **YES**。

更新 SDK 版本后, iOS CocoaPods 运行报错?

1. 删除 iOS 目录下 `Podfile.lock` 文件。
2. 执行 `pod repo update` 。
3. 执行 `pod install` 。
4. 重新运行。

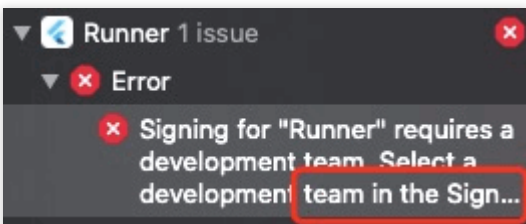
Android Manifest merge failed 编译失败?

1. 请打开 `/example/android/app/src/main/AndroidManifest.xml` 文件。
2. 将 `xmlns:tools="http://schemas.android.com/tools"` 加入到 manifest 中。
3. 将 `tools:replace="android:label"` 加入到 application 中。

```
android > app > src > main > AndroidManifest.xml
 1  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
 2      xmlns:tools="http://schemas.android.com/tools"
 3      package="com.example.mlp">
 4      <!-- io.flutter.app.FlutterApplication is an android.app.Application that
 5           calls FlutterMain.startInitialization(this); in its onCreate method.
 6           In most cases you can leave this as-is, but you if you want to provide
 7           additional functionality it is fine to subclass or reimplement
 8           FlutterApplication and put your custom class here. -->
 9      <application
10          tools:replace="android:label"
11          android:name="io.flutter.app.FlutterApplication"
12          android:label="mlp"
13          android:icon="@mipmap/ic_launcher">
```

因为没有签名, 真机调试报错?

若报错信息如下图所示:



1. 您需购买苹果证书, 并进行配置、签名操作后, 即可在真机上调试。

2. 证书购买成功后，在 `target > signing & capabilities` 中进行配置。

对插件内的 swift 文件做了增删后，build 时查找不到对应文件？

在主工程目录的 `/ios` 文件路径下 `pod install` 即可。

Run 报错“Info.plist, error: No value at that key path or invalid key path: NSBonjourServices”？

执行 `flutter clean` 后，重新运行即可。

Pod install 报错？

若报错信息如下图所示：

```
linzhi@MacBook-Pro ios % pod install

[!] Invalid `Podfile` file: /Users/linzhi/Desktop/source/trtc-flutter-plugin-de
/trtc-flutter-plugin/example/ios/Flutter/Generated.xconfig must exist. If you
running pod install manually, make sure flutter pub get is executed first.
```

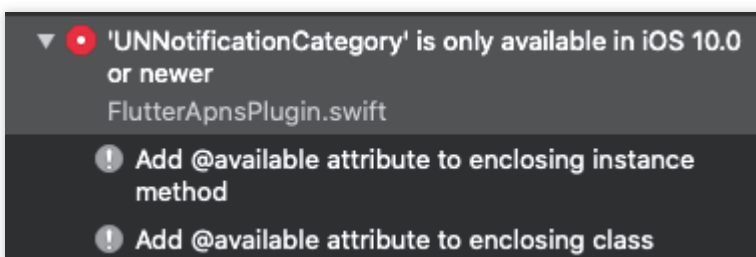
报错信息里面提示 `pod install` 的时候没有 `generated.xconfig` 文件，因此运行报错，您根据提示需要执行 `flutter pub get` 解决。

说明：

该问题是 flutter 编译后的问题，新项目或者执行了 `flutter clean` 后，都不存在这个问题。

Run 的时候 iOS 版本依赖报错？

若报错信息如下图所示：



可能是 pods 的 target 版本无法满足所依赖的插件，因此造成报错。因此您需修改报错 pods 中的 target 到对应的版本。

Flutter 支持自定义采集和渲染吗？

目前不支持。自定义采集和渲染平台支持详情，请参见 [支持的平台](#)。

Electron 相关

最近更新时间：2024-06-07 11:20:49

安装相关

trtc-electron-sdk 是否兼容官方 Electron v12.0.1 版本?

兼容的，trtc-electron-sdk 没有特别依赖 electron 自身的 SDK，所以没有相关的版本依赖。

下载 Electron 时出现 404 错误？

```
(node:23166) UnhandledPromiseRejectionWarning: HTTPError: Response code 404 (Not Found) for https://npm.taobao.org/mirrors/electron/v8.1.1/electron-v8.1.1-darwin-x64.zip
    at EventEmitter.emitter.on.response (/Users/.../.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/as-stream.js:35:24)
    at EventEmitter.emit (events.js:198:13)
    at module.exports (/Users/.../.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/get-response.js:22:10)
    at ClientRequest.handleResponse (/Users/.../.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/request-as-event-emitter.js:155:5)
    at Object.onceWrapper (events.js:286:20)
    at ClientRequest.emit (events.js:203:15)
    at ClientRequest.origin.emit.args [as emit] (/Users/.../.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/eszmrczak/http-timer/source/index.js:37:11)
    at HTTPParser.parserOnIncomingClient [as onIncoming] (_http_client.js:556:21)
    at HTTPParser.parserOnHeadersComplete (_http_common.js:109:17)
    at TLSSocket.socketOnData (_http_client.js:442:20)
(node:23166) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise
was not handled with .catch(). (rejection id: 1)
(node:23166) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero
code.
```

在终端中输入如下指令：

```
$ npm config set electron_custom_dir 8.1.1 # 根据版本号来决定
```

运行相关

Windows 32 系统运行报错 `Error: resource\\trtc_electron_sdk.node is not a valid Win32 application`，提示需要 32 位的 `trtc_electron_sdk.node`?


```

✖ ▶ Error: Cannot open resources\trtc_electron_sdk.node: Error:
resources\trtc_electron_sdk.node is not a valid Win32 application.
resources\trtc_electron_sdk.node
  at Object.<anonymous> (chunk-5974bb06.287d70ae.js:12)
  at Object.6a75 (chunk-5974bb06.287d70ae.js:12)
  at c (app.2e27c2f9.js:1)
  at Object.<anonymous> (chunk-5974bb06.287d70ae.js:1)
  at Object.0542 (chunk-5974bb06.287d70ae.js:1)
  at c (app.2e27c2f9.js:1)
  at Module.97ec (chunk-5974bb06.287d70ae.js:24)
  at c (app.2e27c2f9.js:1)
chunk-vendors.be39b84f.js:57

✖ ▶ Uncaught (in promise) Error: Cannot open resources\trtc_electron_sdk.node:
Error: resources\trtc_electron_sdk.node is not a valid Win32 application.
resources\trtc_electron_sdk.node
  at Object.<anonymous> (chunk-5974bb06.287d70ae.js:12)
  at Object.6a75 (chunk-5974bb06.287d70ae.js:12)
  at c (app.2e27c2f9.js:1)
  at Object.<anonymous> (chunk-5974bb06.287d70ae.js:1)
  at Object.0542 (chunk-5974bb06.287d70ae.js:1)
  at c (app.2e27c2f9.js:1)
  at Module.97ec (chunk-5974bb06.287d70ae.js:24)
  at c (app.2e27c2f9.js:1)
    
```

解决方法：

1. 进入到工程目录下的 trtc-electron-sdk 库目录下 (xxx/node_modules/trtc-electron-sdk)。执行：

```
npm run install -- arch=ia32
```

2. 下载完 32 位的 `trtc_electron_sdk.node` 后，重新对项目进行打包。

vscode terminal 启动 Electron Demo，进入房间后白屏？

vscode 需有摄像头权限，可采用如下方式进行权限添加。

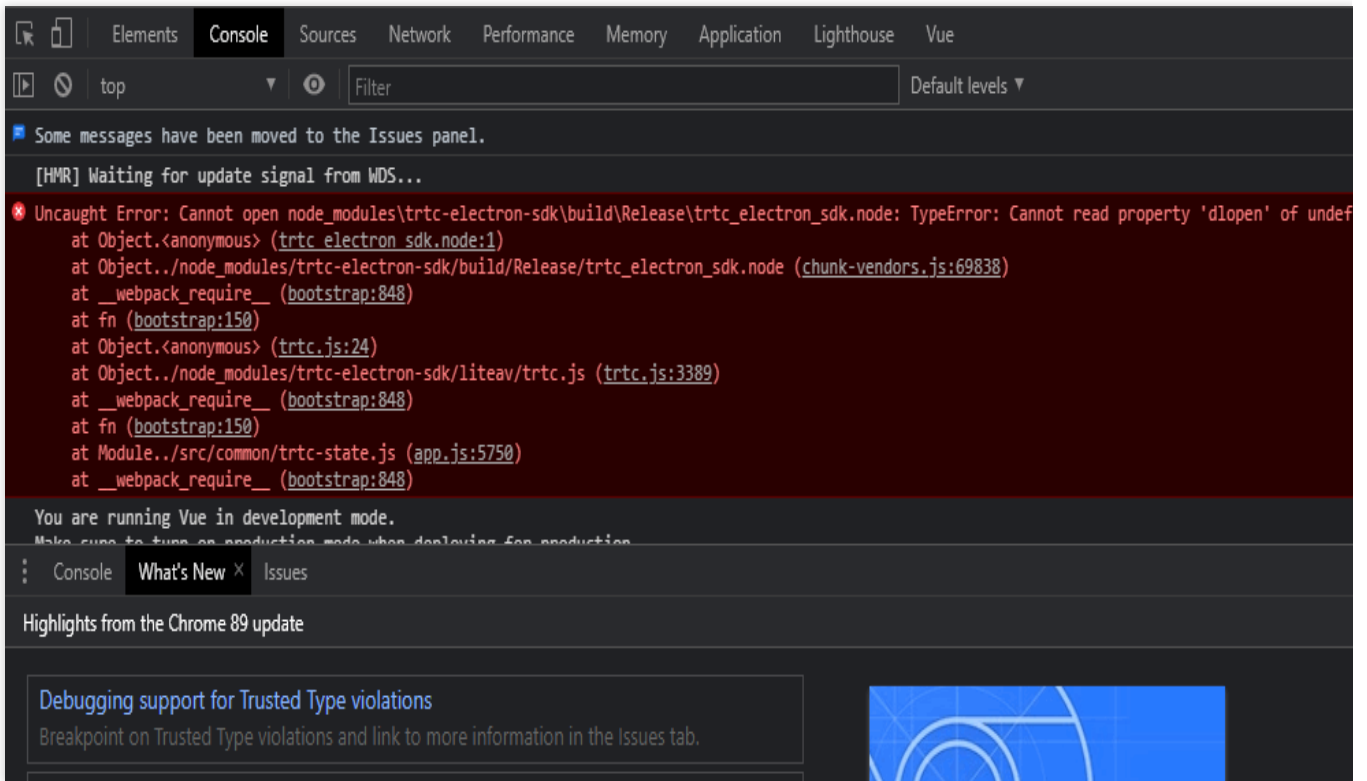
```

cd ~/Library/Application\ Support/com.apple.TCC/
cp TCC.db TCC.db.bak
sqlite3 TCC.db # sqlite> prompt appears.

# for Mojave, Catalina
INSERT into access VALUES('kTCCServiceCamera','com.microsoft.VSCode',0,1,1,NULL

# for BigSur
INSERT into access VALUES('kTCCServiceCamera','com.microsoft.VSCode',0,1,1,NU
    
```

跑 Demo 抛出空指针未定义的错误：“cannot read property 'dlopen' of undefined”？



解决方法：

Electron 12 版本上下文隔离默认启用，可设置 `contextIsolation` 为 `false`。

```
let win = new BrowserWindow({
  width: 1366,
  height: 1024,
  minWidth: 800,
  minHeight: 600,
  webPreferences: {
    nodeIntegration: true,
    contextIsolation: false
  },
});
```

Electron 多次出现重新进房问题？

需要具体 case 进行分析，大致原因如下：

客户端网络状态不好（断网会触发重进房）。

连着发两次进房信令也会重进房的。

有可能是设备负载过高，导致解码失败的重进房。

同一个 UID 多端登录互踢导致的重进房。

终端出现提示“Electron failed to install correctly”？

当看似安装完成，运行项目时，终端上出现以下错误：

```
Error: Electron failed to install correctly, please delete
node_modules/electron and try installing again
```

按照如下三个步骤进行**手动下载**：

1. 执行 `npm config get cache` 查看缓存目录。
2. 手动下载 Electron，并放到缓存目录中。
3. 重新执行 `npm install`。

调用摄像头或麦克风时直接崩溃？

使用 vscode 终端启动项目，当 trtc-electron-sdk 启动摄像头和麦克风时，程序直接崩溃：

解决方案 A：使用有授权的终端运行项目。

解决方案 B：给 vscode 授权：在 **系统偏好设置 > 安全与隐私** 中允许 vscode 的授权。

解决方案 C：按以下步骤关闭保护机制：

- 1.1 重启系统，**按住** `command + r` 键，**直到**系统进入保护模式。
- 1.2 打开 terminal 输入 `csrutil disable` 禁用保护机制。
- 1.3 重启，正常进入系统，此时就可以使用 **vscode** 的终端启动项目了。
- 1.4 如需重新启动保护机制，只需要在第二步中执行 `csrutil enable`。

Electron 在控制台中报错“xx is not defined”？

当运行项目时，Electron 在控制台中提示 `xx is not defined`，其中 `xx` 指代 node 模块。例如：

```
Uncaught ReferenceError: require is not defined
```

在 Electron 的 `main.js` 文件中将 `nodeIntegration` 配置项改成 `true`：

```
let win = new BrowserWindow({
  width: 1366,
  height: 1024,
  webPreferences: {
    nodeIntegration: true, // 请将此项设置为 true
  },
});
```

打包相关

.node 模块的加载问题？

报错信息

打包编译出的程序在运行时，在控制台中看到看到类似的报错信息：

NodeRTCCLoud is not a constructor

```

✖ ▶ TypeError: 1.NodeRTCCLoud is not a constructor
  ▼ TypeError: 1.NodeRTCCLoud is not a constructor
    at new m (file:///E:/www/trtc-electron-test-demo/bin/win-ur
    at e.value (file:///E:/www/trtc-electron-test-demo/bin/win-
    at Module.419 (file:///E:/www/trtc-electron-test-demo/bin/y
  
```

Cannot open xxx/trtc_electron_sdk.node 或者 The specified module could not be found

```

✖ ▶ Error: Cannot open /trtc_electron_sdk.node: Error: The specified module could not b
found.
/trtc_electron_sdk.node
  at Object.<anonymous> (trtc electron sdk.node:1)
  at Object.405 (4.25013499.chunk.js:2)
  at i (index.html:1)
  at Object.<anonymous> (trtc.js:3)
  at Object.395 (4.25013499.chunk.js:2)
  at i (index.html:1)
  at Module.420 (trtc-factory.ts:5)
  at i (index.html:1)
  
```

dlopen(xxx/trtc_electron_sdk.node, 1): image not found

```

Error: Can not open nodeFile: Error: dlopen(/build/trtc_electron_sdk.node, 1): image not fou
  at Object.<anonymous> (4.d24a6187.chunk.js:2)
  at Object.405 (4.d24a6187.chunk.js:2)
  at a (index.html:1)
  at Object.<anonymous> (4.d24a6187.chunk.js:2)
  at Object.395 (4.d24a6187.chunk.js:2)
  at a (index.html:1)
  at Module.420 (6.h83695ae.chunk.js:1)
  
```

解决方法

出现类似上述的信息，说明 trtc_electron_sdk.node 模块没有被正确的打包到程序中，可按照以下步骤进行处理。

1. 安装 native-ext-loader 。

```
$ npm i native-ext-loader -D
```

2. 修改 webpack 配置。

2.1 使 `webpack.config.js` 在构建时可以接收名为 `--target_platform` 的命令行参数，以使代码构建过程按不同的目标平台特点正确打包，在 `module.exports` 之前添加以下代码：

```
const os = require('os');
// 如果不传 target_platform 参数，程序会默认按当前平台类型进行打包
const targetPlatform = (function(){
  let target = os.platform();
  for (let i=0; i<process.argv.length; i++) {
    if (process.argv[i].includes('--target_platform=')) {
      target = process.argv[i].replace('--target_platform=', '');
      break;
    }
  }
  // win32 统一表示 Windows 平台，包含 32 位和 64 位。darwin 表示 Mac 平台
  if (!['win32', 'darwin'].includes) target = os.platform();
  return target;
})();
```

2.2 添加以下 rules 配置：

```
module: {
  rules: [
    {
      test: /\.node$/,
      loader: 'native-ext-loader',
      options: {
        rewritePath: targetPlatform === 'win32' ? './resources' :
        '../Resources'
      }
    },
  ],
}
```

注意：

使用 `vue-cli` 创建的项目，`webpack` 配置存放在 `vue.config.js` 文件中的 `configureWebpack` 选项中。

使用 `create-react-app` 创建的项目，`webpack` 配置文件为 `[项目目录]/node_modules/react-scripts/config/webpack.config.js`。

3. 配置 packages.json 文件，添加打包配置和构建脚本。

3.1 添加 `electron-builder` 打包配置（注意大小写）：

```
"build": {
```

```

"省略": "...",
"win": {
  "extraFiles": [
    {
      "from": "node_modules/trtc-electron-sdk/build/Release/",
      "to": "./resources",
      "filter": ["**/*"]
    }
  ]
},
"mac": {
  "extraFiles": [
    {
      "from": "node_modules/trtc-electron-
sdk/build/Release/trtc_electron_sdk.node",
      "to": "./Resources"
    }
  ]
},
"directories": {
  "output": "./bin"
}
},

```

3.2 添加 scripts 构建、打包脚本 `create-react-app` 项目请参考以下配置：

```

"scripts": {
  "build:mac": "react-scripts build --target_platform=darwin",
  "build:win": "react-scripts build --target_platform=win32",
  "compile:mac": "node_modules/.bin/electron-builder --mac",
  "compile:win64": "node_modules/.bin/electron-builder --win --x64",
  "pack:mac": "npm run build:mac && npm run compile:mac",
  "pack:win64": "npm run build:win && npm run compile:win64"
}

```

3.3 `vue-cli` 项目请参考以下配置：

```

"scripts": {
  "build:mac": "vue-cli-service build --target_platform=darwin",
  "build:win": "vue-cli-service build --target_platform=win32",
  "compile:mac": "node_modules/.bin/electron-builder --mac",
  "compile:win64": "node_modules/.bin/electron-builder --win --x64",
  "pack:mac": "npm run build:mac && npm run compile:mac",
  "pack:win64": "npm run build:win && npm run compile:win64"
}

```

找不到入口文件？

使用 `create-react-app` 创建的项目，使用 `electron-builder` 打包时可能会遇到此问题：

```
$ node_modules\\.bin\\electron-builder.cmd
  • electron-builder version=22.6.0 os=6.1.7601
  • loaded configuration file=package.json ("build" field)
  • public/electron.js not found. Please see
https://medium.com/@kitze/%EF%B8%8F-from-react-to-an-electron-app-ready-for-
production-a0468ecb1da3
  • loaded parent configuration preset=react-cra
```

其中 `public/electron.js not found` 指的就是无法找到入口文件。

解决方案

1. 移动并重命名入口文件：

```
$ cd [项目目录]
$ mv main.electron.js ./public/electron.js
```

2. 修改 `pacakge.json` 文件：

```
{
  "main": "public/electron.js",
  "省略": "..."
}
```

在执行打包时，出现 `fs-extra` 模块的语法错误？

```
[项目目录]\\node_modules\\electron-builder\\node_modules\\fs-
extra\\lib\\empty\\index.js:33
  } catch {
    ^

SyntaxError: Unexpected token {
    at new Script (vm.js:51:7)
```

可以升级到最新的 `node`，具体请参见 [Node.js 官方网站](#)。

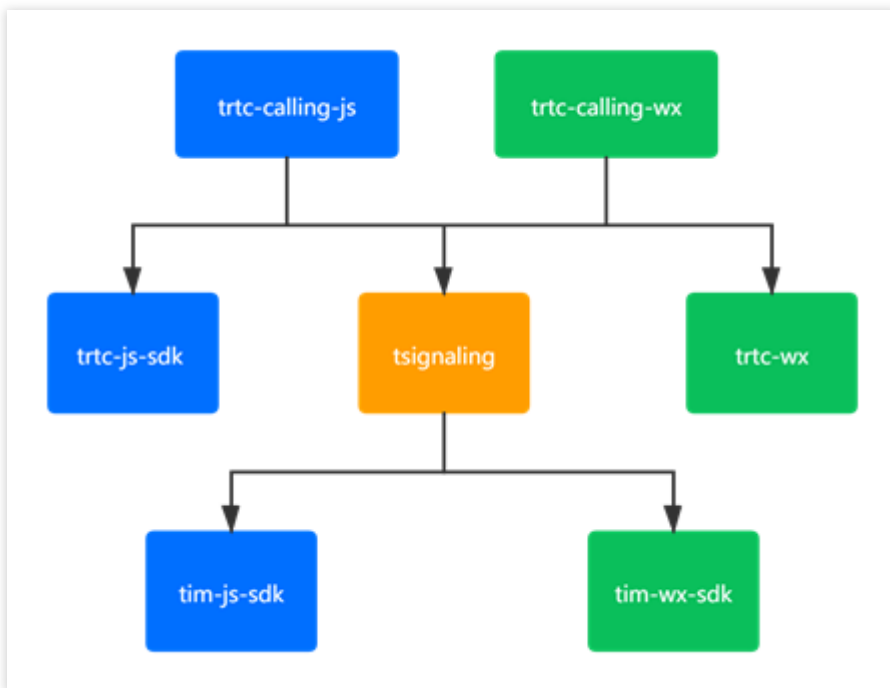
TRTCCalling Web 相关

最近更新时间：2024-03-11 17:04:23

基础问题

什么是 TRTCCalling？

TRTCCalling 是在 TRTC 和 TIM 的基础上诞生的一款快速集成的音视频的解决方案。支持1v1和多人视频/语音通话。



TRTCCalling 是否支持接受 roomID 为字符串？

roomID 可以 string，但只限于数字字符串。

环境问题

Web 端 SDK 支持哪些浏览器？

TRTC Web SDK 对浏览器的详细支持度，请参见 [TRTC Web SDK 对浏览器支持情况](#)。

对于上述没有列出的环境，您可以在当前浏览器打开 [TRTC 能力测试](#) 测试是否完整的支持 WebRTC 的功能。

如何实时检测当前网络的情况？

具体操作请参见 [通话前的网络质量检测](#)。

IM H5 Demo 项目本地跑通功能正常，但放在服务器上用 IP 访问后无法正常视频/语音通话？

背景：IM 的 H5 Demo 在本地跑通后，使用 localhost 能正常实现消息发送、视频/语音通话功能。将项目放到服务器上，用 IP 访问后，文字消息收发、控制台请求返回正常且控制台没有任何错误，但无法正常运行语音/视频通话，获取不到视频图像。

原因：IM 中语音/通话视频使用的是 TRTCCalling SDK，用户使用 IP 访问时，使用的是 HTTP 协议。

解决方案：TRTCCalling SDK 需在 HTTPS 或 localhost 环境下运行。

集成问题

calling 线上 Demo 无法进入 NO_RESP？

原因：NO_RESP 事件触发条件：1-邀请者超时，2-被邀请者不在线。

解决方案：请根据触发条件进行事件处理。

calling 在 iPhone 微信的浏览器打开无法听到对方的声音？

原因：自动播放受限。

解决方案：calling 在 1.0.0 版本时，进行了处理。建议您升级 calling 至 1.0.0 及之后的版本。

TRTCCalling hangup() 报错：“uncaught (in promise) TypeError: cannot read property 'stop' of null”？

原因：用户在监听事件中多次调用 hangup()，导致 hangup 未执行完成又一次触发。

解决方案：hangup() 只需要执行一次，监听事件的后续操作，TRTCCalling 内部已进行处理，不需要再执行 hangup() 方法，只需做自己业务相关操作即可。

最新版本 90 的 Chrome 浏览器，trtccalling.js 提示：“不支持，TRTCCliinet.您的浏览器不兼容此应用”？

原因：IM 版本过低，检测机制有所缺失。

解决方案：建议升级 IM 版本。

在接通过程中报：“TypeError: Cannot read property 'getVideoTracks' of null”？

原因：用户在接受时，还没有获取使用用户视频和麦克风到权限导致的。

解决方案：在使用 startRemoteView、startLocalView 等操作设备方法时，建议使用异步方法。或者升级 TRTCCaling 版本至 1.0.0。

sdkappid 用 script 方式引入时报：“TSignaling._onMessageReceived unknown bussinessID=undefined”？

详情：同一个 sdkAppid 用 script 方式引入的，与 script 引入的能互通，与 npm 引入的或 Android/iOS 的不能互通，且返回警告信息：`TSignaling._onMessageReceived unknown bussinessID=undefined`。

原因：`bussinessID=undefined` 表示该版本 tsignaling 版本为旧版本，旧版本信令有问题。

解决方案：升级 tsignaling 版本，且在引入过程中需注意新版本 tsignaling 的文件名称为 `tsignaling-js`。

提醒：“Uncaught (in promise) Error: createCustomMessage 接口需要 SDK 处于 ready 状态后才能调用”？

原因：未按正确步骤完成初始化。

解决方案：升级 TRTCCalling 版本至 1.0.0，监听 SDK_READY 事件进行后续操作。

提醒：“Uncaught (in promise) RTCError: duplicated play() call observed, please stop() firstly <INVALID_OPERATION 0x1001>”？

原因：在语音通过过程中，调用 startRemoteView 接口。

解决方案：在语音通话过程中，取消 startRemoteView 操作。

提醒：“Uncaught (in promise) Error: inviteID is invalid or invitation has been processed”？

详情：Web 端 trtccalling 与 native 端互通，web 呼叫 native 后，native 接听而 web 端摄像头还未开启，本地预览还没有画面就点挂断，native 还在通话页面。返回错误信息：`Uncaught (in promise) Error: inviteID is invalid or invitation has been processed`。

原因：在获取用户设备时，若用户未授权音视频设备，可以进入音视频通话房间，但挂断时，native 无法收到挂断信令。

解决方案：calling 的 1.0.0 版本，进行前置获取并获取不成功时，不允许用户进入通话。建议您升级 calling 至 1.0.0 及之后的版本。

主叫呼叫成功后，被叫打印了日志（应该收到了呼叫），但没有走回调 handleNewInvitationReceived 回调？

原因：TRTCCalling <= 0.6.0 和 Tsignaling <= 0.3.0 版本过低。

解决方案：升级 TRTCCalling 和 Tsignaling 到最新版本。

TRTCCalling 在 CALL 之后 主动 reject 后无法在呼叫？

原因：call 之后主动 reject 后，calling 状态没重置导致。

解决方案：升级 TRTCCalling 版本 >= 1.0.3。

Error: TRTCCalling.call - 获取用户设备权限失败？

原因：TRTCCalling 没有设备权限或者没有对于设备。

解决方案：

使用 [TRTC 设备检测](#) 进行检查。

访问 **Chrome** 的网站设置（<chrome://settings/content>）查看使用 TRTCCalling 的网站是否开启摄像头/麦克风权限。

TRTCCalling web 是否支持接收离线消息？

不支持接收离线消息。

支持离线消息推送，可以通过 `call / groupCall` 中的 [offlinePushInfo](#) 添加需要推送的消息。

音视频质量相关

最近更新时间：2022-09-06 16:56:27

一、视频问题

TRTC 视频画面出现黑边怎么去掉？

设置 `TRTCVideoFillMode_Fill`（填充）即可解决，TRTC 视频渲染模式分为填充和适应，本地渲染画面可以通过 `setLocalViewFillMode()` 设置，远端渲染画面可以通过 `setRemoteViewFillMode` 设置：

`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。

`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。

TRTC 出现卡顿怎么排查？

可以通过对应的 RoomID、UserID 在实时音视频控制台的 [监控仪表盘](#) 页面查看通话质量：

通过接受端视角查看发送端和接收端用户情况。

查看发送端和接收端是否丢包率比较高，如果丢包率过高一般是网络状况不稳定导致卡顿。

查看帧率和 CPU 占用率，帧率比较低和 CPU 使用率过高都会导致卡顿现象。

TRTC 出现画质不佳，模糊、马赛克等现象怎么排查？

清晰度主要和码率有关，检查 SDK 码率是否配置的比较低，如果高分辨率低码率容易产生马赛克现象。

TRTC 会通过云端 QOS 流控策略，根据网络状况动态调整码率、分辨率，网络比较差时容易降低码率导致清晰度下降。

检查进房时使用的 VideoCall 模式还是 Live 模式，针对通话场景 VideoCall 模式主打低延时和保流畅，所以在弱网情况下会更容易牺牲画质确保流畅，对画质更加看重的场景建议使用 Live 模式。

TRTC 自己的本地画面和远端画面左右相反？

本地默认采集的画面是镜像的。App 端可以通过 `setLocalViewMirror` 接口设置，该接口只改变本地摄像头预览画面的镜像模式；也可以通过 `setVideoEncoderMirror` 接口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（以及服务器录制下来的）画面效果。Web 端可以通过 `createStream` 接口修改 `mirror` 参数来设置。

TRTC 设置视频编码输出的方向没有效果？

需要设置 `setGSensorMode()` 为 `TRTCGSensorMode_Disable`，关闭重力感应，否则调用 `setVideoEncoderRotation` 后远端用户观看到的画面不会变化。

TRTC 正常上行有数据，为什么旁路拉流失败看不到画面？

请确认是否已在 [应用管理>功能配置](#) 中开启自动旁路推流。

预览/播放画面出现旋转？

使用 TRTCSDK 摄像头采集：

建议更新 SDK 版本到最新版本。

如果是特殊设备，可以使用本地预览画面渲染角度接口 `setLocalViewRotation`，远端视频画面的渲染角度接口 `setRemoteViewRotation`，设置编码器输出的画面渲染角度接口 `setVideoEncoderRotation` 进行调整。具体接口使用说明请参见 [视频画面旋转](#)。

使用自定义视频采集：

建议更新 SDK 版本到最新版本。

确认下将采集的视频画面的角度是正确。

将视频数据填充给 TRTCSDK，检查是否给 `TRTCCLoudDef.TRTCVideoFrame` 设置旋转角度

如果是特殊设备，可以使用本地预览画面渲染角度接口 `setLocalViewRotation`，远端视频画面的渲染角度接口 `setRemoteViewRotation`，设置编码器输出的画面渲染角度接口 `setVideoEncoderRotation` 进行调整。具体接口使用说明请参见 [视频画面旋转](#)。

视频出现镜像问题？

使用前摄像头进行视频通话，会有镜像效果，所以本地预览和远端观众画面是左右颠倒。

如何横屏推流？

开发者可能用设备是电视或者因场景需要横屏推流，具体实现请参见[视频画面旋转和缩放](#)

直播中出现播放黑屏是什么原因呢？

播放失败或解码失败，参看播放失败的处理办法。

Metadata 问题，例如 `metadata` 中只有音频流信息，但实际数据中即有音频也有视频;或者是开始的数据只有音频，但播放一段时间后又加入了视频信息。这种情况一般建议修改源流的 `metadata` 信息。

视频编码数据里没有画面信息，只有类似 SEI 之类的帧时，解码也没有画面，自然就黑屏了，这种一般是定制类的视频数据。

直播中出现播放花屏、绿屏，是什么原因呢？

一般是丢失了 I 帧导致，因为 P 帧和 B 帧的解码都依赖于 I 帧，如果 I 帧丢失了，P 帧和 B 帧都会解码失败，于是画面就会花屏、拖影、绿屏等情况，先使用不同播放器如 ffplay、VLC、Potplayer 同时播放一下同一路流，如果播放器都出现花屏、绿屏的情况，通常就是音视频源码流就存在问题，需要检查源码流。

Metadata 发生变化，大多数的播放器一般只会在开始解码之前解析一次 metadata 设置解码参数，当画面发生变化，例如分辨率变化了，但播放器解码参数没有重新配置，就有可能导致花屏、绿屏，这种情况下，最好的办法是推流端在直播过程中不改变编码参数，这样就不会引起 metadata 信息的修改。

硬件编解码的兼容性问题，这类情况通常出现在 Android 设备上，一些 Android 设备的硬件编解码器实现不好，兼容性不佳。这种情况，最好的办法，换软编软解对比。

推流端和播放端颜色格式不一致导致，例如推流端使用的是 NV12，而播放端支持的是 I420，则解码时就会因为颜色格式不一致而显示为花屏、绿屏等情况。这种情况下，统一推拉流两端的颜色格式。

二、音频问题

TRTC 的通话同时用了点播播放器 TXVodPlayer 播放，为什么播放声音很小？

通过 `setSystemVolumeType` 接口设置通话时使用的系统音量类型，设置为媒体音量模式 `TRTCSystemVolumeTypeMedia` 即可解决。

如何选择媒体音量和通话音量？

通过 `setSystemVolumeType` 接口，支持自主选择通话音量和媒体音量。

`TRTCAudioVolumeTypeAuto`：默认类型，麦上通话音量、麦下媒体音量。

`TRTCAudioVolumeTypeVOIP`：始终使用通话音量。

`TRTCAudioVolumeTypeMedia`：始终使用媒体音量。

声音小怎么处理？

若所有观众听到的声音都小，则是上行因素导致：

检查 Windows 和 mac 的 `setCurrentDeviceVolume`、全平台的 `setAudioCaptureVolume` 接口的 `volume` 值是否小于 50，可以适当调大音量。

检查 3A 处理的 AGC 自动增益是否开启。

检查是否是由于蓝牙耳机导致。

若部分观众听到声音小，则是下行因素导致：

检查 `setAudioPlaybackVolume`、`setCurrentDeviceVolume` 接口的 `volume` 值是否小于 50，可以适当调大音量。

手机端可以检查是否调用 `setAudioRoute` API 切换了听筒播放。

声音卡顿断续？

打开 [监控仪表盘](#)，在音频选项卡中查看：

若接收端和发送端“设备状态”的 CPU 超过90%，建议关闭其他后台程序。

若音频上下行存在明显丢包，rtt 值波动较大，表明当前用户网络质量不佳，建议切换稳定网络。

为什么有回声？

通话双方的设备相距太近的时候，属于正常现象，测试时请相互距离远一点；是否误关了 3A 处理的 AEC 回声消除。

声音音质差或者声音忽大忽小？

若您外接了声卡，开了耳返，当连麦时就会引发此问题，建议您在外接声卡时关闭耳返，因为声卡一般自带了耳返功能。

Web 通话过程中出现回声、杂音、噪声、声音小？

通话双方的设备相距太近的时候，属于正常现象，测试时请相互距离远一点。当其他端听到 Web 端的声音存在回声、噪声、杂音等情况时，说明 Web 端的 3A 处理没有生效。若您使用了浏览器原生 [getUserMedia](#) API 进行自定义采集，则需要手动设置 3A 参数：

echoCancellation：回声消除开关。

noiseSuppression：噪声抑制开关。

autoGainControl：自动增益开关。

若您使用 [TRTC.createStream](#) 接口进行采集，则无需手动设置 3A 参数，SDK 默认开启 3A。

三、其他问题

TRTC 怎么监测网络状态，实现信号强弱展示功能？

可以使用 [onNetworkQuality\(\)](#) 监听当前网络的上行和下行质量，以Android为例，可参考 [TRTC-API-Example](#) 实现信号强弱功能。

为什么设备摄像头或麦克风出现被占用等异常现象？

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等，硬件设备的释放是异步操作，待资源释放完毕，SDK 会通过 `TRTCCLoudListener` 中的 `onExitRoom()` 回调通知上层。如果您要再次调用 `enterRoom()` 或切换到其他的音视频 SDK，请等待 `onExitRoom()` 回调到来之后再执行相关操作。

如何判断打开摄像头成功？

通过回调方法 `onCameraDidReady`，当收到该回调时表示摄像头已经准备就绪。

如何判断打开麦克风成功？

通过回调方法 `onMicDidReady`，当收到该回调时表示麦克风已经准备就绪。

打开摄像头失败？

确认摄像头权限是否授予

如果设备是电视、盒子等，使用的摄像头是外接的。目前 TRTCSDK 是支持识别外接摄像头的。因此，需要确认摄像头接头和设备是否接触良好。

TRTC 有哪些技术统计指标？

注意：

该场景适用于 iOS/Mac、Android、Windows 平台。

SDK 提供回调方法 `onStatistics(TRTCStatistics statics)`，每 2 秒回一次技术指标。包括当前 `appCpu`（App 的 CPU 使用率）、`systemCpu`（当前系统的 CPU 使用率）、`rtt`（延迟）、`upLoss`（上行丢包率）、`downLoss`（下行丢包率）以及本地成员和远端成员的音视频统计信息，具体参数请参见 [TRTCStatistics](#) 类型说明。

其他问题

最近更新时间：2022-09-27 15:24:28

直播、互动直播、实时音视频以及旁路直播有什么区别和关系？

直播（关键词：一对多，RTMP/HLS/HTTP-FLV，CDN）

直播分为推流端、播放端以及直播云服务，云服务使用 CDN 进行直播流的分发。推流使用的是通用标准的协议 RTMP，经过 CDN 分发后，播放时一般可以选择 RTMP、HTTP-FLV 或 HLS（H5 支持）等方式进行观看。

互动直播（关键词：连麦、PK）

互动直播是一种业务形式，指主播与观众之间进行互动连麦，主播与主播之间进行互动PK的一种直播类型。

实时音视频（关键词：多人互动，UDP 私有协议，低延时）

实时音视频（Tencent Real-Time Communication，TRTC）主要应用场景是音视频互动和低延时直播，使用基于 UDP 的私有协议，其延迟可低至100ms，典型的场景就是 QQ 电话、腾讯会议、大班课等。腾讯云实时音视频（TRTC）覆盖全平台，除了iOS/Android/Windows之外，还支持小程序以及 WebRTC 互通，并且支持通过云端混流的方式将画面旁路直播到 CDN。

旁路直播（关键词：云端混流，RTC 旁路转推，CDN）

旁路直播是一种技术，指的是将低延时连麦房间里的多路推流画面复制出来，在云端将画面混合成一路，并将混流后的画面推流给直播 CDN 进行分发播放。

两台设备同时运行 Demo，为什么看不到彼此的画面？

请确保两台设备在运行 Demo 时使用的是不同的 UserID，TRTC 不支持同一个 UserID（除非 SDKAppID 不同）在两个设备同时使用。

为什么使用 CDN 直播观看时房间里只有一个人时画面又卡又模糊？

请将 `enterRoom` 中 `TRTCAppScene` 参数指定为 `TRTCAppSceneLIVE`。

VideoCall 模式针对视频通话做了优化，所以在房间中只有一个用户时，画面会保持较低的码率和帧率以节省用户的网络流量，看起来会感觉又卡又模糊。

为什么线上的房间都进不去了？

可能因为房间权限控制已开启。房间权限控制开启后，当前 SDKAppID 下的房间将需要在 `TRTCParamEnc` 中设置 `privateMapKey` 才能进入。如果您线上业务正在运营中，并且线上版本并没有加入 `privateMapKey` 的相关逻辑，请不要开启该功能，更多详情请参见 [进房权限保护](#)。

如何查看 TRTC 日志？

TRTC 的日志默认压缩加密，后缀为 .xlog。日志是否加密是可以通过 `setLogCompressEnabled` 来控制，生成的文件名里面含 C(compressed) 的就是加密压缩的，含 R(raw) 的就是明文的。

iOS&Mac：`sandbox的Documents/log`

Android：

6.7及之前的版本：`/sdcard/log/tencent/liteav`

6.8之后的版本：`/sdcard/Android/data/包名/files/log/tencent/liteav/`

8.5之后的版本：`/sdcard/Android/data/包名/files/log/liteav/`

Windows：

8.8之前的版本：`%appdata%/tencent/liteav/log`

8.8以及8.8之后的版本：`%appdata%/liteav/log`

Web：打开浏览器控制台，或使用 vConsole 记录 SDK 打印信息

说明：

查看 .xlog 文件需要下载解密工具，在python 2.7环境中放到 xlog 文件同目录下直接使用 `python decode_mars_log_file.py` 运行即可。

日志解密工具下载地址：`dldir1.qq.com/hudongzhibo/log_tool/decode_mars_log_file.py`。


出现10006 error 该如何处理？

如果出现"Join room failed result: 10006 error: service is suspended,if charge is overdue,renew it"，请确认您的实时音视频应用的服务状态是否为可用状态。

登录 [实时音视频控制台](#) > [应用管理](#)，选择您创建的应用，单击 [应用信息](#)，在应用信息面板即可确认服务状态。

TRTC Service Status

Status **Normal**

You have enabled pay-as-you-go. To avoid losses caused by service suspension, make sure your account has sufficient balance. You can view your account balance in [Billing Center](#) .

More ▾

进入房间返回错误码-100018是什么原因？

原因是 UserSig 校验失败，可能情况如下：

参数 SDKAppID 传入不正确，可登录实时音视频控制台，选择 [应用管理](#) 并查看对应的 SDKAppID。

参数 UserID 对应的验证签名 UserSig 传入不正确，可登录实时音视频控制台，选择 [开发辅助](#) > [UserSig生成&校验](#) 校验 UserSig。

如何跨房连麦（主播 PK）？

可以使用 connectOtherRoom 接口。主播调用 connectOtherRoom() 后，可以通过 onConnectOtherRoom 回调得到跨房 PK 的结果。主播一所在房间里的所有人，都会通过 onUserEnter 的回调，得到主播二进房的通知。主播二所在房间的所有人，也都会通过 onUserEnter 的回调得到主播一进房的通知。

离开房间接口 exitRoom() 是否必须调用？

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。

旁路录制的各种场景下生成录制文件是什么格式的？

以 [实时音视频控制台](#) 中配置录制文件格式为准。

如何判断音视频通话是否推流成功？

通过回调方法 onSendFirstLocalVideoFrame，在 enterRoom 并 startLocalPreview 成功后开始摄像头采集，并将采集到的画面进行编码。当 SDK 成功向云端送出第一帧视频数据后，会抛出这个回调事件。

如何判断纯音频通话是否推流成功？

通过回调方法 onSendFirstLocalAudioFrame，在 enterRoom 并 startLocalPreview 成功后开始麦克风采集，并将采集到的声音进行编码。当 SDK 成功向云端送出第一帧音频数据后，会抛出这个回调事件。

我能否查询所有的 UserID？

暂不支持统计所有 UserID，可以在客户端用户注册帐号成功后将用户信息一并写入 SQL 中进行管理或查询。

相同 UserID 是否支持同时进入多个房间？

TRTC 同一时间不支持两个相同的 userId 进入房间，否则会相互干扰。

为什么调用 `setAudioRoute` 设置音频路由（听筒/扬声器）不生效？

只能在通话音量模式下切换听筒/扬声器，即仅在2个及以上用户连麦时调用才生效。

TRTC 只支持腾讯云控制台开启自动录制吗？怎么实现手动开启录制？

TRTC 支持手动录制，具体操作办法如下：

1. 进入 [应用管理](#) > [功能配置](#)，开启 [自动旁路推流](#)，不开启 [启动云端录制](#)。
2. 用户进房间后，按照流 ID 生成规则，计算出 `userid` 对应的 `streamid`。
3. 使用云直播的 [创建录制任务 API](#)，对 `streamid` 启动录制任务。

DomainName 为 `[bizid].livepush.myqcloud.com`。

AppName 为 `trtc_[sdkappid]`。

StreamName 为 `streamid`。

4. 录制任务完成后，云直播会将文件写入到云点播，并通过 [录制回调事件通知](#)。

TRTC 怎么校验生成的 UserSig 是否正确？进房报错-3319、-3320错误怎么排查？

可登录实时音视频控制台，选择 [开发辅助](#)>[UserSig生成&校验](#) 校验 UserSig。

TRTC 如何查看通话时长和使用量？

可在实时音视频控制台的 [用量统计](#) 页面查看。

TRTC 如何维护用户列表、统计直播间的观看人数？

如果开发者项目工程中有集成 [即时通信IM](#)，可以直接通过 IM 群人数统计接口进行统计。但该方案统计出的人数不是很准确，如果开发者对在线人数要求不高，可以直接上述方案。

如果开发者需要很准确的统计在线人数，建议自行实现统计逻辑：

1. 增加观众数（Client -> Server）当有新的观众加入时，意味着某个房间的观众数要 + 1，可以让 App 的观众端在进入房间时向 Server 发送一次累加请求。
2. 减少观众数（Client -> Server）当有观众退出房间时，意味着某个房间的观众数要 - 1，可以让 App 的观众端在退出房间时向 Server 发送一次累减请求。

进房时报-100013错误码，错误信息为 `ERR_SERVER_INFO_SERVICE_SUSPENDED`，是什么问题？

该错误表示服务不可用。请检查：

套餐包剩余分钟数是否大于0。

腾讯云账号是否欠费。

TRTC 启动云端录制但没有生成录制文件怎么排查？

1. 确保已在 [实时音视频控制台](#) 开启 [自动旁路推流](#) 和 [启动云端录制](#)。
2. TRTC 房间中有用户正常上行音视频数据才会开始录制。
3. 旁路 CDN 拉流正常才会有录制文件生成。
4. 若刚开始只有音频，中途切换了视频，根据录制模板的不同可能会只生成视频时间段的录制文件，或只生成音频时间段的录制文件。

邀请嘉宾连线如何告知嘉宾房间号？

告知嘉宾房间号的操作可以加在自定义消息中，解析消息内容获取 roomid，相关说明请参见 [创建自定义消息](#)、[TIMMsgSendNewMsg](#)。

是否可以至少两个人进入房间，才开始录音？

可以。若您需获取录制混流后的音频数据，您可在 [启动云端混流](#) 后，制定输出流 ID，并调用直播接口 [创建录制任务](#)。

Windows 端怎么采集到被分享应用播放的声音？

通过调用 [startSystemAudioLoopback](#) 接口，可打开系统声音采集。

Windows 会议模式中，如何实现主播对观众发起音视频连线的功能？

需要搭配另一个云产品 [即时通信 IM](#) 达成连线需求。

呼叫的大致逻辑为：A 给 B 发送自定义消息 X 并唤起呼叫页面，X 展示效果自行处理，B 接收到 X 后调起被呼叫页面，B 单击 [enterRoom](#) 进入房间，并发送自定义消息 X1 给 A，A 收到 X1（自行决定是否展示）同时调用 [enterRoom](#) 进入房间。使用 IM 来发送自定义消息。

观众如何查看房间里连线的画面？

当观众使用直播模式时，观众进入房间观看会通过 TRTCCloudDelegate 中的 [onUserVideoAvailable](#) 回调获知主播的 userid（连麦的人也会 [enterRoom](#) 进房，对于观众来说也是主播）。然后观众可以调用 [startRemoteView](#) 方法来显示主播的视频画面。

更多详细操作，请参见 [跑通直播模式\(Windows\)](#)。

TRTC 有 Linux SDK 吗？

Linux SDK 暂未完全开放，若您需咨询或使用相关服务，请联系：colleenyu@tencent.com。

TRTC 是否支持在视频通话（或连麦互动）时进行屏幕分享吗？

支持。TRTC 连麦互动或者视频通话时，摄像头采集画面作为主流画面，同时还支持屏幕分享作为辅流画面。分享出去的屏幕就是当前手机屏幕画面，包含连麦互动或者视频通话的窗口。