

音视频通话 SDK

客户端 API



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

客户端 API

Android

API 概览

TUICallKit

TUICallEngine

TUICallObserver

类型定义

iOS

API 概览

TUICallKit

TUICallEngine

TUICallObserver

类型定义

Web

API 概览

TUICallKit

TUICallEngine

TUICallEvent

小程序

API 概览

TUICallKit

TUICallEngine

TUICallEvent

uni-app 客户端

API 概览

TUICallKit

TUICallEngine

TUICallEvent

Flutter

API 概览

TUICallKit

TUICallEngine

TUICallObserver

类型定义

客户端 API

Android

API 概览

最近更新时间：2024-07-25 17:44:51

TUICallKit (含 UI 接口)

TUICallKit API 是音视频通话组件的**含 UI 接口**，使用TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景。

API	描述
createInstance	创建 TUICallKit 实例（单例模式）
setSelfInfo	设置用户的头像、昵称
call	发起 1v1 通话
call	发起 1v1 通话，支持自定义离线推送消息显示内容
groupCall	发起群组通话
groupCall	发起群组通话，支持自定义离线推送消息显示内容
joinInGroupCall	主动加入当前的群组通话中
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
enableIncomingBanner	开启/关闭来电横幅显示

TUICallEngine (无 UI 接口)

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

API	描述
createInstance	创建 TUICallEngine 实例（单例模式）

<code>destroyInstance</code>	销毁 TUICallEngine 实例（单例模式）
<code>init</code>	完成音视频通话基础能力的鉴权
<code>addObserver</code>	增加事件回调
<code>removeObserver</code>	移除回调接口
<code>call</code>	发起 1v1 通话
<code>groupCall</code>	发起群组通话
<code>accept</code>	接听通话
<code>reject</code>	拒绝通话
<code>hangup</code>	结束通话
<code>ignore</code>	忽略通话
<code>inviteUser</code>	在群组通话中，邀请其他人加入
<code>joinInGroupCall</code>	主动加入当前的群组通话中
<code>switchCallMediaType</code>	切换通话媒体类型，比如视频通话切音频通话
<code>startRemoteView</code>	开始订阅远端用户视频流
<code>stopRemoteView</code>	停止订阅远端用户视频流
<code>openCamera</code>	开启摄像头
<code>closeCamera</code>	关闭摄像头
<code>switchCamera</code>	切换前后摄像头
<code>openMicrophone</code>	打开麦克风
<code>closeMicrophone</code>	关闭麦克风
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/扬声器）
<code>setSelfInfo</code>	设置用户的昵称、头像
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式（ 群组通话版套餐 支持）
<code>setVideoRenderParams</code>	设置用户视频画面的渲染模式

<code>setVideoEncoderParams</code>	设置视频编码的编码参数
<code>getTRTCCloudInstance</code>	高级接口，用于接入第三方美颜等
<code>setBeautyLevel</code>	设置美颜级别，支持关闭默认美颜

TUICallObserver

TUICallObserver 是 TUICallEngine 对应的回调事件类，您可以通过此回调，来监听自己感兴趣的回调事件。

API	描述
<code>onError</code>	通话过程中错误回调
<code>onCallReceived</code>	通话请求的回调
<code>onCallCancelled</code>	通话取消的回调
<code>onCallBegin</code>	通话接通的回调
<code>onCallEnd</code>	通话结束的回调
<code>onCallMediaTypeChanged</code>	通话的媒体类型发生改变的回调
<code>onUserReject</code>	xxxx 用户拒绝通话的回调
<code>onUserNoResponse</code>	xxxx 用户不响应的回调
<code>onUserLineBusy</code>	xxxx 用户忙线的回调
<code>onUserJoin</code>	xxxx 用户加入通话的回调
<code>onUserLeave</code>	xxxx 用户离开通话的回调
<code>onUserVideoAvailable</code>	xxx 用户是否有视频流的回调
<code>onUserAudioAvailable</code>	xxx 用户是否有音频流的回调
<code>onUserVoiceVolumeChanged</code>	所有用户音量大小的反馈回调
<code>onUserNetworkQualityChanged</code>	所有用户网络质量的反馈回调
<code>onKickedOffline</code>	当前用户被踢下线
<code>onUserSigExpired</code>	在线时票据过期

关键类型定义

API	描述
TUICallDefine.MediaType	通话的媒体类型，枚举类型：视频通话、语音通话
TUICallDefine.Role	通话的角色，枚举类型：主叫、被叫
TUICallDefine.Status	通话的状态，枚举类型：空闲、待接听、接听中
TUICallDefine.CallParams	通话扩展参数：离线推送消息显示内容
TUICommonDefine.RoomId	音视频房间 Id，支持数字、字符串两种类型
TUICommonDefine.Camera	摄像头 Id 参数，枚举类型：前摄、后摄
TUICommonDefine.AudioPlaybackDevice	声音的播放设备，枚举类型：扬声器、听筒
TUICommonDefine.NetworkQualityInfo	当前的网络质量信息

TUICallKit

最近更新时间：2024-11-29 09:13:12

TUICallKit API 简介

TUICallKit API 是音视频通话组件的含 UI 接口，使用 TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景，更详细的接入步骤，详情请参见 [快速接入TUICallKit](#)。

API 概览

API	描述
createInstance	创建 TUICallKit 实例（单例模式）
setSelfInfo	设置用户的昵称、头像
call	发起 1v1 通话
call	发起 1v1 通话，支持自定义房间号、通话邀请超时时间，离线推送内容等
groupCall	发起群组通话
groupCall	发起群组通话，支持自定义房间号、通话邀请超时时间，离线推送内容等
joinInGroupCall	主动加入当前的群组通话中
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
enableIncomingBanner	开启/关闭来电横幅显示

API 详情

createInstance

创建 TUICallKit 的单例。

Kotlin

```
fun createInstance(context: Context): TUICallKit
```

Java

```
TUICallKit createInstance(Context context)
```

setSelfInfo

设置用户昵称、头像。用户昵称不能超过500字节，用户头像必须是 URL 格式。

Kotlin

```
fun setSelfInfo(nickname: String?, avatar: String?, callback:
TUICommonDefine.Callback?)
```

Java

```
void setSelfInfo(String nickname, String avatar,
TUICommonDefine.Callback callback)
```

参数如下表所示：

参数	类型	含义
nickname	String	目标用户的昵称
avatar	String	目标用户的头像

call

拨打电话（1v1通话）。

Kotlin

```
fun call(userId: String, callMediaType: TUICallDefine.MediaType)
```

Java

```
void call(String userId, TUICallDefine.MediaType callMediaType)
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话

call

拨打电话（1v1通话），支持自定义房间号、通话邀请超时时间，离线推送内容等。

```
void call(String userId, TUICallDefine.MediaType callMediaType,  
          TUICallDefine.CallParams params, TUICommonDefine.Callback  
          callback)
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话
params	TUICallDefine.CallParams	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等

groupCall

发起群组通话。

⚠ 注意

使用群组通话前需要创建 IM 群组，如果已经创建，请忽略。

群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

Kotlin

```
fun groupCall(groupId: String, userIdList: List<String?>?,
callMediaType: TUICallDefine.MediaType)
```

Java

```
void groupCall(String groupId, List<String> userIdList,
TUICallDefine.MediaType callMediaType);
```

参数如下表所示：

参数	类型	含义
groupId	String	此次群组通话的群 ID
userIdList	List	目标用户的 userId 列表
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话

groupCall

发起群组通话，支持自定义房间号、通话邀请超时时间，离线推送内容等。

⚠ 注意

使用群组通话前需要创建 IM 群组，如果已经创建，请忽略。

群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

Kotlin

```
fun groupCall(
    groupId: String, userIdList: List<String?>?,
    callMediaType: TUICallDefine.MediaType, params: CallParams?,
    callback: TUICommonDefine.Callback?
)
```

Java

```
void groupCall(String groupId, List<String> userIdList,
TUICallDefine.MediaType callMediaType,
```

```
TUICallDefine.CallParams params, TUICommonDefine.Callback  
callback);
```

参数如下表所示：

参数	类型	含义
groupId	String	此次群组通话的群 ID
userIdList	List	目标用户的 userId 列表
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话
params	TUICallDefine.CallParams	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等

joinInGroupCall

加入群组中已有的音视频通话。

⚠ 注意

加入群组中已有的音视频通话前，需要提前创建或加入 IM 群组，并且群组中已有用户在通话中，如果已经创建，请忽略。

群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

Kotlin

```
fun joinInGroupCall(roomId: RoomId?, groupId: String?, callMediaType:  
TUICallDefine.MediaType?)
```

Java

```
void joinInGroupCall(TUICommonDefine.RoomId roomId, String groupId,  
TUICallDefine.MediaType callMediaType);
```

参数如下表所示：

参数	类型	含义
----	----	----

roomId	TUICommonDefine.RoomId	此次通话的音视频房间 ID
groupId	String	此次群组通话的群 ID
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话

setCallingBell

设置自定义来电铃声。

这里仅限传入本地文件地址，需要确保该文件目录是应用可以访问的。

- 铃声设置后与设备绑定，更换用户，铃声依旧会生效。
- 如需恢复默认铃声，`filePath` 传空即可。

Kotlin

```
fun setCallingBell(filePath: String?)
```

Java

```
void setCallingBell(String filePath);
```

enableMuteMode

开启/关闭静音模式。

开启后，收到通话请求，不会播放来电铃声。

Kotlin

```
fun enableMuteMode(enable: Boolean)
```

Java

```
void enableMuteMode(boolean enable);
```

enableFloatWindow

开启/关闭悬浮窗功能。

默认为 `false`，通话界面左上角的悬浮窗按钮隐藏，设置为 `true` 后显示。

Kotlin

```
fun enableFloatWindow(enable: Boolean)
```

Java

```
void enableFloatWindow(boolean enable);
```

enableIncomingBanner

开启/关闭来电横幅显示。

默认为 `false`，被叫端收到邀请后默认弹出全屏通话等待界面，开启后先展示一个横幅，然后根据需要拉起全屏通话界面。

```
fun enableIncomingBanner(enable: Boolean)
```

TUICallEngine

最近更新时间：2024-07-25 17:44:51

TUICallEngine API 简介

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套接口自己封装交互。

API 概览

API	描述
createInstance	创建 TUICallEngine 实例（单例模式）
destroyInstance	销毁 TUICallEngine 实例（单例模式）
init	完成音视频通话基础能力的鉴权
addObserver	增加事件回调
removeObserver	移除回调接口
call	发起 1v1 通话
groupCall	发起群组通话
accept	接听通话
reject	拒绝通话
hangup	结束通话
ignore	忽略通话
inviteUser	在群组通话中，邀请其他人加入
joinInGroupCall	主动加入当前的群组通话中
switchCallMediaType	切换通话媒体类型，比如视频通话切音频通话
startRemoteView	开始订阅远端用户视频流
stopRemoteView	停止订阅远端用户视频流
openCamera	开启摄像头
closeCamera	关闭摄像头

<code>switchCamera</code>	切换前后摄像头
<code>openMicrophone</code>	打开麦克风
<code>closeMicrophone</code>	关闭麦克风
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/扬声器）
<code>setSelfInfo</code>	设置用户的昵称、头像
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式（ 群组通话版套餐 支持）
<code>setVideoRenderParams</code>	设置用户视频画面的渲染模式
<code>setVideoEncoderParams</code>	设置视频编码的编码参数
<code>getTRTCCloudInstance</code>	高级接口，用于接入第三方美颜等
<code>setBeautyLevel</code>	设置美颜级别，支持关闭默认美颜

API 详情

createInstance

创建 TUICallEngine 的单例。

```
TUICallEngine createInstance(Context context)
```

destroyInstance

销毁 TUICallEngine 的单例。

```
void destroyInstance();
```

init

初始化函数，请在使用所有功能之前先调用该函数，以便完成包含通话服务鉴权在内初始化动作。

```
void init(int SDKAppID, String userId, String userSig,
TUICommonDefine.Callback callback)
```

参数如下表所示：

参数	类型	含义
SDKAppID	int	您可以在实时音视频控制台 > 应用管理 > 应用信息中查看 SDKAppID
userId	String	当前用户的 ID，字符串类型，只允许包含英文字母（a-z 和 A-Z）、数字（0-9）、连词符（-）和下划线（_）
userSig	String	腾讯云设计的一种安全保护签名，获取方式请参见 如何计算及使用 UserSig
callback	TUICommonDefine.Callback	初始化回调， <code>onSuccess</code> 表示初始化成功

addObserver

添加回调接口，您可以通过这个接听，监听 `TUICallObserver` 相关的事件回调。

```
void addObserver(TUICallObserver observer);
```

removeObserver

移除回调接口。

```
void removeObserver(TUICallObserver observer);
```

call

拨打电话（1v1通话）。

⚠ 注意：

字符串房间号目前只支持 Android、iOS、Flutter、Uniapp 平台，Web、小程序等其他平台后续支持，敬请期待！

```
void call(String userId, TUICallDefine.MediaType callMediaType,
          TUICallDefine.CallParams params, TUICommonDefine.Callback
          callback);
```

参数如下表所示：

参数	类型	含义
----	----	----

userId	String	目标用户的 userId
callMedia Type	TUICallDefine.Me diaType	通话的媒体类型，例如：视频通话、语音通话
params	TUICallDefine.Cal lParams	通话参数扩展字段，例如：房间号、通话邀请超时时间，离线推送自定义内容等

groupCall

发起群组通话。

⚠ 注意：

使用群组通话前需要创建 IM 群组，如果已经创建，请忽略；

字符串房间号目前只支持 Android、iOS、Flutter、Uniapp 平台，Web、小程序等其他平台后续支持，敬请期待！

```
void groupCall(String groupId, List<String> userIdList,  
TUICallDefine.MediaType callMediaType,  
TUICallDefine.CallParams params, TUICommonDefine.Callback  
callback);
```

参数如下表所示：

参数	类型	含义
groupId	String	此次群组通话的群 ID
userIdLi st	List	目标用户的 userId 列表
callMedi aType	TUICallDefine.Me diaType	通话的媒体类型，例如：视频通话、语音通话
params	TUICallDefine.Cal lParams	通话参数扩展字段，例如：房间号、通话邀请超时时间，离线推送自定义内容等

accept

接受当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数接听来电。

```
void accept(TUICommonDefine.Callback callback);
```

reject

拒绝当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数拒绝来电。

```
void reject(TUICommonDefine.Callback callback);
```

ignore

忽略当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数忽略来电，此时主叫会收到 `onUserLineBusy` 的回调。

备注：如果您的业务中存在直播、会议等场景，在直播/会议中的情况时，也可以调用这个函数来忽略此次来电。

```
void ignore(TUICommonDefine.Callback callback);
```

hangup

挂断当前通话，当您处于通话中，可以调用该函数结束通话。

```
void hangup(TUICommonDefine.Callback callback);
```

inviteUser

邀请用户加入此次群组通话。

使用场景：一个群组通话中的用户主动邀请其他人时使用。

```
void inviteUser(List<String> userIdList, TUICallDefine.CallParams  
params,  
                TUICommonDefine.ValueCallback callback);
```

参数如下表所示：

参数	类型	含义
userIdList	List	目标用户的 userId 列表
params	TUICallDefine.CallParams	通话参数扩展字段，例如：通话邀请超时时间，离线推送自定义内容等

 **注意：**

邀请他人进入当前通话，该场景下，自定义 RoomId 无效，SDK 会邀请他人加入当前邀请者所在的房间。

joinInGroupCall

主动加入此次群组通话。

使用场景：群组内用户主动加入此次群组通话使用。

```
void joinInGroupCall(TUICommonDefine.RoomId roomId, String groupId,
                    TUICallDefine.MediaType callMediaType,
                    TUICommonDefine.Callback callback);
```

参数如下表所示：

参数	类型	含义
roomId	TUICommonDefine.RoomId	此次通话的音视频房间 ID
groupId	String	此次群组通话的群 ID
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话

switchCallMediaType

切换视频通话到语音通话。

```
void switchCallMediaType(TUICallDefine.MediaType callMediaType);
```

参数如下表所示：

参数	类型	含义
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话

startRemoteView

开始订阅远端用户的视频数据，此接口在 setRenderView 之后调用。

```
void startRemoteView(String userId, TUIVideoView videoView,
```

```
TUICommonDefine.PlayCallback callback);
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId
videoView	TUIVideoView	待渲染的视图

stopRemoteView

停止订阅远端用户的视频数据。

```
void stopRemoteView(String userId);
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId

openCamera

开启摄像头。

```
void openCamera(TUICommonDefine.Camera camera, TUIVideoView videoView,  
TUICommonDefine.Callback callback);
```

参数如下表所示：

参数	类型	含义
camera	TUICommonDefine.Camera	前置/后置 摄像头
videoView	TUIVideoView	待渲染的视图

closeCamera

关闭摄像头。

```
void closeCamera();
```

switchCamera

切换前后摄像头。

```
void switchCamera(TUICommonDefine.Camera camera);
```

参数如下表所示：

参数	类型	含义
camera	TUICommonDefine.Camera	前置/后置 摄像头

openMicrophone

打开麦克风。

```
void openMicrophone(TUICommonDefine.Callback callback);
```

closeMicrophone

关闭麦克风。

```
void closeMicrophone();
```

selectAudioPlaybackDevice

选择音频播放设备。

目前支持听筒、扬声器，在通话场景中，可以使用这个接口来开启/关闭免提模式。

```
void selectAudioPlaybackDevice(TUICommonDefine.AudioPlaybackDevice device);
```

参数如下表所示：

参数	类型	含义
device	TUICommonDefine.AudioPlaybackDevice	听筒/扬声器

setSelfInfo

设置用户昵称、头像。

用户昵称不能超过500字节，用户头像必须是 URL 格式。

```
void setSelfInfo(String nickname, String avatar,
    TUICommonDefine.Callback callback);
```

参数如下表所示：

参数	类型	含义
nickname	String	用户昵称
avatar	String	用户头像（格式为 URL）

enableMultiDeviceAbility

开启/关闭 TUICallEngine 的多设备登录模式（[群组通话版套餐](#)支持）。

```
void enableMultiDeviceAbility(boolean enable, TUICommonDefine.Callback
    callback);
```

setVideoRenderParams

设置用户视频画面的渲染模式。

```
void setVideoRenderParams(String userId,
    TUICommonDefine.VideoRenderParams params, TUICommonDefine.Callback
    callback);
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId
params	TUICommonDefine.VideoRenderParams	视频画面渲染参数，例如：画面旋转角度、填充模式

setVideoEncoderParams

设置视频编码的编码参数。

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

```
void setVideoEncoderParams(TUICommonDefine.VideoEncoderParams params,
    TUICommonDefine.Callback callback);
```

参数如下表所示：

参数	类型	含义
params	TUICommonDefine.VideoEncoderParams	视频编码的参数，例如：分辨率、视频宽高比

getTRTCCloudInstance

高级接口。

用于接入第三方美颜等。

```
TRTCCloud getTRTCCloudInstance();
```

setBeautyLevel

设置美颜级别，支持关闭默认美颜。

```
void setBeautyLevel(float level, TUICommonDefine.Callback callback);
```

参数如下表所示：

参数	类型	含义
level	float	美颜级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显

TUICallObserver

最近更新时间：2023-12-29 16:01:31

UICallObserver API 简介

TUICallObserver 是 TUICallEngine 对应的回调事件类，您可以通过此回调，来监听自己感兴趣的回调事件。

回调事件概览

API	描述
onError	通话过程中错误回调
onCallReceived	通话请求的回调
onCallCancelled	通话取消的回调
onCallBegin	通话接通的回调
onCallEnd	通话结束的回调
onCallMediaTypeChanged	通话媒体类型发生改变的回调
onUserReject	xxxx 用户拒绝通话的回调
onUserNoResponse	xxxx 用户不响应的回调
onUserLineBusy	xxxx 用户忙线的回调
onUserJoin	xxxx 用户加入通话的回调
onUserLeave	xxxx 用户离开通话的回调
onUserVideoAvailable	xxx 用户是否有视频流的回调
onUserAudioAvailable	xxx 用户是否有音频流的回调
onUserVoiceVolumeChanged	所有用户音量大小的反馈回调
onUserNetworkQualityChanged	所有用户网络质量的反馈回调
onKickedOffline	当前用户被踢下线
onUserSigExpired	在线时票据过期

回调事件详情

onError

错误回调。

说明：
SDK 不可恢复的错误，一定要监听，并分情况给用户适当的界面提示。

```
void onError(int code, String msg);
```

参数如下表所示：

参数	类型	含义
code	int	错误码
msg	String	错误信息

onCallReceived

收到一个新的来电请求回调，被叫会收到，您可以通过监听这个事件，来决定是否显示通话接听界面。

```
void onCallReceived(String callerId, List<String> calleeIdList, String  
groupId,  
                    TUICallDefine.MediaType callMediaType, String  
userData)
```

参数如下表所示：

参数	类型	含义
callerId	String	主叫 ID（邀请方）
calleeIdList	List	被叫 ID 列表（被邀请方）
groupId	String	群组通话 ID
callMediaType	TUICallDefine.MediaType	通话的媒体类型，比如视频通话、语音通话
userData	String	用户增加的扩展字段，详见 TUICallDefine.CallParams

onCallCancelled

表示此次通话主叫取消、被叫超时、拒接等，涉及多个场景，您可以通过监听这个事件来实现类似未接来电、重置 UI 状态等显示逻辑。

- 主叫取消：主叫收到该回调（userId 为自己）；被叫收到该回调（userId 为主叫的 ID）
- 被叫超时：主叫会同时收到 `onUserNoResponse` 和 `onCallCancelled` 回调（userId 是自己的 ID）；被叫收到 `onCallCancelled` 回调（userId 是自己的 ID）
- 被叫拒接：主叫会同时收到 `onUserReject` 和 `onCallCancelled` 回调（userId 是自己的 ID）；被叫收到 `onCallCancelled` 回调（userId 是自己的 ID）
- 被叫忙线：主叫会同时收到 `onUserLineBusy` 和 `onCallCancelled` 回调（userId 是自己的 ID）
- 异常中断：被叫接收通话失败，收到该回调（userId 是自己的 ID）

```
void onCallCancelled(String userId);
```

参数如下表所示：

参数	类型	含义
userId	String	用户的 ID

onCallBegin

表示通话接通，主叫和被叫都可以收到，您可以通过监听这个事件来开启云端录制、内容审核等流程。

```
void onCallBegin(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType callMediaType, TUICallDefine.Role callRole);
```

参数如下表所示：

参数	类型	含义
roomId	<code>TUICommonDefine.RoomId</code>	此次通话的音视频房间 ID
callMediaType	<code>TUICallDefine.MediaType</code>	通话的媒体类型，视频通话、语音通话
callRole	<code>TUICallDefine.Role</code>	角色，枚举类型：主叫、被叫

onCallEnd

表示通话挂断，主叫和被叫都可以收到，您可以通过监听这个事件来显示通话时长、通话类型等信息，或者来停止云端的录制流程。

```
void onCallEnd(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType  
callMediaType, TUICallDefine.Role callRole, long totalTime);
```

参数如下表所示：

参数	类型	含义
roomId	TUICommonDefine.RoomId	此次通话的音视频房间 ID
callMediaType	TUICallDefine.MediaType	通话的媒体类型，视频通话、语音通话
callRole	TUICallDefine.Role	角色，枚举类型：主叫、被叫
totalTime	long	此次通话的时长，单位：秒

⚠ 注意：

客户端的事件一般都会随着杀进程等异常事件丢失掉，如果您需要通过监听通话时长来完成计费逻辑，建议使用 REST API 来完成这类流程。

onCallMediaTypeChanged

表示通话的媒体类型发生变化。

```
void onCallMediaTypeChanged(TUICallDefine.MediaType  
oldCallMediaType, TUICallDefine.MediaType newCallMediaType);
```

参数如下表所示：

参数	类型	含义
oldCallMediaType	TUICallDefine.MediaType	旧的通话类型
newCallMediaType	TUICallDefine.MediaType	新的通话类型

onUserReject

通话被拒绝的回调，在 1v1 通话中，只有主叫方会收到拒绝回调，在群组通话中，所有被邀请者都可以收到该回调。

```
void onUserReject(String userId);
```

参数如下表所示：

参数	类型	含义
userId	String	拒绝用户的 ID

onUserNoResponse

对方无回应的回调。

```
void onUserNoResponse(String userId);
```

参数如下表所示：

参数	类型	含义
userId	String	无响应用户的 ID

onUserLineBusy

通话忙线回调。

```
void onUserLineBusy(String userId);
```

参数如下表所示：

参数	类型	含义
userId	String	忙线用户的 ID

onUserJoin

有用户进入此次通话的回调。

```
void onUserJoin(String userId);
```

参数如下表所示：

参数	类型	含义
----	----	----

userId	String	加入当前通话的用户 ID
--------	--------	--------------

onUserLeave

有用户离开此次通话的回调。

```
void onUserLeave(String userId);
```

参数如下表所示：

参数	类型	含义
userId	String	离开当前通话的用户 ID

onUserVideoAvailable

用户是否开启视频上行回调。

```
void onUserVideoAvailable(String userId, boolean isVideoAvailable);
```

参数如下表所示：

参数	类型	含义
userId	String	通话用户 ID
isVideoAvailable	boolean	用户视频是否可用

onUserAudioAvailable

用户是否开启音频上行回调。

```
void onUserAudioAvailable(String userId, boolean isAudioAvailable);
```

参数如下表所示：

参数	类型	含义
userId	String	用户 ID
isAudioAvailable	boolean	用户音频是否可用

onUserVoiceVolumeChanged

用户通话音量的回调。

```
void onUserVoiceVolumeChanged(Map<String, Integer> volumeMap);
```

参数如下表所示：

参数	类型	含义
volumeMap	Map<String, Integer>	音量表，根据每个 userId 可以获取对应用户的音量大小，音量最小值为 0，音量最大值为 100

onUserNetworkQualityChanged

用户网络质量的回调。

```
void onUserNetworkQualityChanged(List<TUICallDefine.NetworkQualityInfo> networkQualityList);
```

参数如下表所示：

参数	类型	含义
networkQualityList	List	网络状态，根据每个 userId 可以获取对应用户当前的网络质量

onKickedOffline

当前用户被踢下线：此时可以 UI 提示用户，并再次重新调用初始化。

```
void onKickedOffline();
```

onUserSigExpired

在线时票据过期：此时您需要生成新的 userSig，并再次重新调用初始化。

```
void onUserSigExpired();
```

类型定义

最近更新时间：2023-09-04 16:49:01

常用结构

TUICallDefine

类型	描述
CallParams	通话参数
OfflinePushInfo	离线推送厂商配置信息

TUICommonDefine

类型	描述
RoomId	通话中音视频房间 ID
NetworkQualityInfo	用户的网络质量
VideoRenderParams	视频画面的渲染参数
VideoEncoderParams	视频编码参数

枚举定义

TUICallDefine

类型	说明
MediaType	通话类型
Role	通话角色
Status	通话状态
Scene	通话场景
IOSOfflinePushType	iOS 离线推送类型

TUICommonDefine

类型	说明
----	----

AudioPlaybackDevice	音频设备
Camera	摄像头类型
NetworkQuality	网络质量
FillMode	视频画面填充模式
Rotation	视频画面旋转方向
ResolutionMode	视频宽高比模式
Resolution	视频分辨率

CallParams

通话参数

参数	类型	描述
roomId	TUICommonDefine.RoomId	通话中音视频房间ID
offlinePushInfo	OfflinePushInfo	厂商离线推送配置信息
timeout	int	通话超时时间，默认：30s，单位：秒
userData	String	发起通话时自定义扩展字段，被叫端在 onCallReceived 中回调

OfflinePushInfo

厂商离线推送配置信息，详情配置步骤请参考：[离线唤醒](#)

参数	类型	描述
title	String	离线推送展示通知栏标题
desc	String	离线推送展示通知栏内容
ignoreIOSBadge	boolean	离线推送忽略 badge 计数（仅对 iOS 生效），如果设置为 true，在 iOS 接收端，这条消息不会使 APP 的应用图标未读计数增加。
iOSSound	String	离线推送声音设置（仅对 iOS 生效）。当 sound = IOS_OFFLINE_PUSH_NO_SOUND，表示接收时不会播放声音。当 sound =

		IOS_OFFLINE_PUSH_DEFAULT_SOUND，表示接收时播放系统声音。如果要自定义 iOSSound，需要先把语音文件链接进 Xcode 工程，然后把语音文件名（带后缀）设置给 iOSSound
androidSound	String	离线推送声音设置（仅对 Android 生效, IMSDK 6.1 及以上版本支持）。只有华为和谷歌手机支持设置声音提示，小米手机设置声音提示，请您参照： 小米自定义铃声 。另外，谷歌手机 FCM 推送在 Android 8.0 及以上系统设置声音提示，必须调用 setAndroidFCMChannelID 设置好 channelId，才能生效
androidOPPOChannelID	String	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID
androidVIVOClassification	int	VIVO 推送消息分类 (待废弃接口，VIVO 推送服务于 2023 年 4 月 3 日优化消息分类规则，推荐使用 setAndroidVIVOCategory 设置消息类别)。0：运营消息 1：系统消息，默认取值为 1
androidXiaoMiChannelID	String	小米手机 8.0 系统及以上的渠道 ID
androidFCMChannelID	String	FCM 通道手机 8.0 系统及以上的渠道 ID
androidHuaWeiCategory	String	华为推送消息分类，详见： 华为消息分类标准
isDisablePush	boolean	是否关闭推送（默认开启推送）
iOSPushType	IOSOfflinePushType	iOS 离线推送类型，默认：APNs

RoomId

通话中音视频房间ID

- intRoomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 intRoomId 需要填写为 0。若两者都填，SDK 将优先选用 intRoomId。
- 不要混用 intRoomId 和 strRoomId，因为它们之间是不互通的，比如数字123和字符串123两个完全不同的房间。

参数	类型	描述
----	----	----

intRoomId	int	数字房间号 取值范围 1 – 2147483647(2^31-1)
strRoomId	String	字符串房间号 推荐取值 限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： - 大小写英文字母（a-zA-Z）； - 数字（0-9）； - 空格、！、#、\$、%、&、（、）、+、-、:、;、<、=、.、>、?、@、[、]、^、_、{、}、 、~、,、。

⚠ 注意：

字符串房间号目前只支持 Android、iOS、Flutter、Uniapp 平台，Web、小程序等其他平台后续支持，敬请期待！

NetworkQualityInfo

用户网络质量

参数	类型	描述
userId	String	用户 ID
quality	NetworkQuality	网络质量

VideoRenderParams

视频画面的渲染参数

参数	类型	描述
fillMode	VideoRenderParams.FillMode	视频画面填充模式
rotation	VideoRenderParams.Rotation	视频画面旋转方向

VideoEncoderParams

视频编码参数

参数	类型	描述
resolution	VideoEncoderParams.Resolution	视频分辨率
resolutionMode	VideoEncoderParams.ResolutionMode	视频宽高比

MediaType

通话类型

类型	取值	描述
Unknown	0	未知类型
Audio	1	语音通话
Video	2	视频通话

Role

通话角色

类型	取值	描述
None	0	未知类型
Caller	1	主叫（邀请方）
Called	2	被叫（被邀请方）

Status

通话状态

类型	取值	描述
None	0	未定义
Waiting	1	通话等待中
Accept	2	通话已接听

Scene

通话场景

类型	取值	描述
GROUP_CALL	0	群组通话
MULTI_CALL	1	匿名群组通话（暂不支持，敬请期待）
SINGLE_CALL	2	单人通话

IOSOfflinePushType

iOS离线推送类型

类型	取值	描述
APNs	0	普通推送
VoIP	1	VoIP 推送

AudioPlaybackDevice

音频设备

类型	取值	描述
Earpiece	0	听筒
Speakerphone	1	扬声器

Camera

前置/后置 摄像头

类型	取值	描述
Front	0	前置摄像头
Back	1	后置摄像头

NetworkQuality

网络质量

类型	取值	描述
Unknown	0	未定义

Excellent	1	当前网络非常好
Good	2	当前网络比较好
Poor	3	当前网络一般
Bad	4	当前网络较差
Vbad	5	当前网络很差
Down	6	当前网络不满足通话的最低要求

FillMode

视频画面填充模式

类型	取值	描述
Fill	0	填充模式：即将画面内容居中等比缩放以充满整个显示区域，超出显示区域的部分将会被裁剪掉，此模式下画面可能不完整。
Fit	1	适应模式：即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边。

Rotation

视频画面旋转方向

类型	取值	描述
Rotation_0	0	不旋转
Rotation_90	1	顺时针旋转90度
Rotation_180	2	顺时针旋转180度
Rotation_270	3	顺时针旋转270度

ResolutionMode

视频宽高比模式

类型	取值	描述
Landscape	0	横屏分辨率，例如：Resolution.Resolution_640_360 + ResolutionMode.Landscape = 640 × 360。

Portrait	1	竖屏分辨率，例如：Resolution.Resolution_640_360 + ResolutionMode.Portrait = 360 × 640。
----------	---	---

Resolution

视频分辨率

类型	取值	描述
Resolution_640_360	108	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps
Resolution_640_480	62	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps
Resolution_960_540	110	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps
Resolution_960_720	64	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps
Resolution_1280_720	112	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps
Resolution_1920_1080	114	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps

iOS

API 概览

最近更新时间：2024-07-25 17:44:51

TUICallKit (含 UI 接口)

TUICallKit API 是音视频通话组件的**含 UI 接口**，使用 TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景。

API	描述
createInstance	创建 TUICallKit 实例（单例模式）
setSelfInfo	设置用户的头像、昵称
call	发起 1v1 通话
call	发起 1v1 通话，支持自定义房间号、通话邀请超时时间，离线推送内容等
groupCall	发起群组通话
groupCall	发起群组通话，支持自定义房间号、通话邀请超时时间，离线推送内容等
joinInGroupCall	主动加入当前的群组通话中
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
enableIncomingBanner	开启/关闭来电横幅显示

TUICallEngine (无 UI 接口)

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

API	描述
createInstance	创建 TUICallEngine 实例（单例模式）

<code>destroyInstance</code>	销毁 TUICallEngine 实例（单例模式）
<code>init</code>	完成音视频通话基础能力的鉴权
<code>addObserver</code>	增加事件回调
<code>removeObserver</code>	移除回调接口
<code>call</code>	发起 1v1 通话
<code>groupCall</code>	发起群组通话
<code>accept</code>	接听通话
<code>reject</code>	拒绝通话
<code>hangup</code>	结束通话
<code>ignore</code>	忽略通话
<code>inviteUser</code>	在群组通话中，邀请其他人加入
<code>joinInGroupCall</code>	主动加入当前的群组通话中
<code>switchCallMediaType</code>	切换通话媒体类型，比如视频通话切音频通话
<code>startRemoteView</code>	开始订阅远端用户视频流
<code>stopRemoteView</code>	停止订阅远端用户视频流
<code>openCamera</code>	开启摄像头
<code>closeCamera</code>	关闭摄像头
<code>switchCamera</code>	切换前后摄像头
<code>openMicrophone</code>	打开麦克风
<code>closeMicrophone</code>	关闭麦克风
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/免提）
<code>setSelfInfo</code>	设置用户的头像、昵称
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式（ 群组通话版套餐 支持）
<code>setVideoRenderParams</code>	设置用户视频画面的渲染模式

<code>setVideoEncoderParams</code>	设置视频编码的编码参数
<code>getTRTCCloudInstance</code>	高级接口，用于接入第三方美颜等

TUICallObserver

TUICallObserver 是 TUICallEngine 对应的回调事件类，您可以通过此回调，来监听自己感兴趣的回调事件。

API	描述
<code>onError</code>	通话过程中错误回调
<code>onCallReceived</code>	通话请求的回调
<code>onCallCancelled</code>	通话取消的回调
<code>onCallBegin</code>	通话接通的回调
<code>onCallEnd</code>	通话结束的回调
<code>onCallMediaTypeChanged</code>	通话的媒体类型发生改变的回调
<code>onUserReject</code>	xxxx 用户拒绝通话的回调
<code>onUserNoResponse</code>	xxxx 用户不响应的回调
<code>onUserLineBusy</code>	xxxx 用户忙线的回调
<code>onUserJoin</code>	xxxx 用户加入通话的回调
<code>onUserLeave</code>	xxxx 用户离开通话的回调
<code>onUserVideoAvailable</code>	xxxx 用户是否有视频流的回调
<code>onUserAudioAvailable</code>	xxxx 用户是否有音频流的回调
<code>onUserVoiceVolumeChanged</code>	所有用户音量大小的反馈回调
<code>onUserNetworkQualityChanged</code>	所有用户网络质量的反馈回调
<code>onKickedOffline</code>	当前用户被踢下线
<code>onUserSigExpired</code>	在线时票据过期

关键类型定义

API	描述
TUICallMediaType	通话的媒体类型，枚举类型：视频通话、语音通话
TUICallRole	通话的角色，枚举类型：主叫、被叫
TUICallStatus	通话的状态，枚举类型：空闲、待接听、接听中
TUIRoomId	音视频房间Id，支持数字、字符串两种类型
TUICamera	摄像头Id参数，枚举类型：前摄、后摄
TUIAudioPlaybackDevice	声音的播放设备，枚举类型：扬声器、听筒
TUINetworkQuality	当前的网络质量信息

TUICallKit

最近更新时间：2024-11-29 09:13:12

TUICallKit API 简介

TUICallKit API 是音视频通话组件的含 UI 接口，使用TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景，更详细的接入步骤，详情请参见 [快速接入TUICallKit](#)。

API 概览

API	描述
createInstance	创建 TUICallKit 实例（单例模式）
setSelfInfo	设置用户的昵称、头像
call	发起 1v1 通话
call	发起 1v1 通话，支持自定义房间号、通话邀请超时时间，离线推送内容等
groupCall	发起群组通话
groupCall	发起群组通话，支持自定义房间号、通话邀请超时时间，离线推送内容等
joinInGroupCall	主动加入当前的群组通话中
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
enableIncomingBanner	开启/关闭来电横幅显示

API 详情

createInstance

创建 TUICallKit 的单例。

Objective-C

```
- (instancetype)createInstance;
```

Swift

```
public static func createInstance() -> TUICallKit
```

setSelfInfo

设置用户昵称、头像。用户昵称不能超过500字节，用户头像必须是 URL 格式。

Objective-C

```
- (void)setSelfInfo:(NSString * _Nullable)nickname avatar:(NSString *  
_Nullable)avatar succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

Swift

```
public func setSelfInfo(nickname: String, avatar: String, succ:@escaping  
TUICallSucc, fail: @escaping TUICallFail)
```

参数	类型	含义
nickname	NSString	用户昵称
avatar	NSString	用户头像（格式为 URL）
succ	TUICallSucc	成功回调
fail	TUICallFail	失败回调

call

拨打电话（1v1通话）

Objective-C

```
- (void)call:(NSString *)userId callMediaType:  
(TUICallMediaType)callMediaType;
```

Swift

```
public func call(userId: String, callMediaType: TUICallMediaType)
```

参数如下表所示：

参数	类型	含义
userId	NSString	目标用户的 userId
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话

call

拨打电话（1v1通话），支持自定义房间号、通话邀请超时时间，离线推送内容等。

Objective-C

```
- (void)call:(NSString *)userId callMediaType:
(TUICallMediaType)callMediaType params:(TUICallParams *)params succ:
(TUICallSucc __nullable)succ fail:(TUICallFail __nullable)fail;
```

Swift

```
public func call(userId: String, callMediaType: TUICallMediaType,
params: TUICallParams,
succ: @escaping TUICallSucc, fail: @escaping
TUICallFail)
```

参数如下表所示：

参数	类型	含义
userId	NSString	目标用户的 userId
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话
params	TUICallParams	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等
succ	TUICallSucc	成功回调

fail	TUICallFail	失败回调
------	-------------	------

groupCall

发起群组通话

⚠ 注意:

使用群组通话前需要创建 IM 群组，如果已经创建，请忽略。

群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

Objective-C

```
- (void)groupCall:(NSString *)groupId userIdList:(NSArray<NSString *>
*)userIdList callMediaType:(TUICallMediaType)callMediaType;
```

Swift

```
public func groupCall(groupId: String, userIdList: [String],
callMediaType: TUICallMediaType)
```

参数	类型	含义
groupId	NSString	此次群组通话的群 ID
userIdList	NSArray	目标用户的 userId 列表
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话

groupCall

发起群组通话，支持自定义房间号、通话邀请超时时间，离线推送内容等。

⚠ 注意:

使用群组通话前需要创建 IM 群组，如果已经创建，请忽略。

群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

Objective-C

```
- (void)groupCall:(NSString *)groupId userIdList:(NSArray<NSString *>
*)userIdList callMediaType:(TUICallMediaType)callMediaType params:
(TUICallParams *)params succ:(TUICallSucc __nullable)succ fail:
(TUICallFail __nullable)fail;
```

Swift

```
public func groupCall(groupId: String, userIdList: [String],
callMediaType: TUICallMediaType, params: TUICallParams, succ: @escaping
TUICallSucc, fail: @escaping TUICallFail)
```

参数	类型	含义
groupId	NSString	此次群组通话的群 ID
userIdList	NSArray	目标用户的 userId 列表
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话
params	TUICallParams	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等
succ	TUICallSucc	成功回调
fail	TUICallFail	失败回调

joinInGroupCall

发起群组通话

⚠ 注意：

加入群组中已有的音视频通话前，需要提前创建或加入 IM 群组，并且群组中已有用户在通话中，如果已经创建，请忽略。

群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

Objective-C

```
- (void)joinInGroupCall:(TUIRoomId *)roomId groupId:(NSString *)groupId
callMediaType:(TUICallMediaType)callMediaType;
```

Swift

```
public func joinInGroupCall(roomId: TUIRoomId, groupId: String,
callMediaType: TUICallMediaType)
```

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间 ID
groupId	NSString	此次群组通话的群 ID
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话

setCallingBell

设置自定义来电铃声。

这里仅限传入本地文件地址，需要确保该文件目录是应用可以访问的。

- 铃声设置后与设备绑定，更换用户，铃声依旧会生效。
- 如需恢复默认铃声，`filePath` 传空即可。

Objective-C

```
- (void)setCallingBell:(NSString *)filePath;
```

Swift

```
public func setCallingBell(filePath: String)
```

enableMuteMode

开启后，收到通话请求，不会播放来电铃声。

Objective-C

```
- (void)enableMuteMode:(BOOL)enable;
```

Swift

```
public func enableMuteMode(enable: Bool)
```

enableFloatWindow

开启/关闭悬浮窗功能。

默认为 `false`，通话界面左上角的悬浮窗按钮隐藏，设置为 `true` 后显示。

Objective-C

```
- (void)enableFloatWindow:(BOOL)enable;
```

Swift

```
public func enableFloatWindow(enable: Bool)
```

enableIncomingBanner

开启/关闭来电横幅显示。

默认为 `false`，被叫端收到邀请后默认弹出全屏通话等待界面，开启后先展示一个横幅，然后根据需要拉起全屏通话界面。

```
public func enableIncomingBanner(enable: Bool)
```

TUICallEngine

最近更新时间：2024-07-25 17:44:51

TUICallEngine API 简介

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套接口自己封装交互。

API 概览

API	描述
createInstance	创建 TUICallEngine 实例（单例模式）
destroyInstance	销毁 TUICallEngine 实例（单例模式）
init	完成音视频通话基础能力的鉴权
addObserver	增加事件回调
removeObserver	移除回调接口
call	发起 1v1 通话
groupCall	发起群组通话
accept	接听通话
reject	拒绝通话
hangup	结束通话
ignore	忽略通话
inviteUser	在群组通话中，邀请其他人加入
joinInGroupCall	主动加入当前的群组通话中
switchCallMediaType	切换通话媒体类型，比如视频通话切音频通话
startRemoteView	开始订阅远端用户视频流
stopRemoteView	停止订阅远端用户视频流
openCamera	开启摄像头
closeCamera	关闭摄像头

<code>switchCamera</code>	切换前后摄像头
<code>openMicrophone</code>	打开麦克风
<code>closeMicrophone</code>	关闭麦克风
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/扬声器）
<code>setSelfInfo</code>	设置用户的昵称、头像
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式（ 群组通话版套餐 支持）
<code>setVideoRenderParams</code>	设置用户视频画面的渲染模式
<code>setVideoEncoderParams</code>	设置视频编码的编码参数
<code>getTRTCCloudInstance</code>	高级接口，用于接入第三方美颜等
<code>setBeautyLevel</code>	设置美颜级别，支持关闭默认美颜

API 详情

createInstance

创建 TUICallEngine 的单例。

```
- (TUICallEngine *)createInstance;
```

destroyInstance

销毁 TUICallEngine 的单例。

```
- (void)destroyInstance;
```

init

初始化函数，请在所有功能之前先调用该函数，以便完成包含通话服务鉴权在内初始化动作。

```
- (void)init:(NSString *)sdkAppID userId:(NSString *)userId userSig:
```

```
(NSString *)userSig succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

参数如下表所示：

参数	类型	含义
sdkAppID	NSString	您可以在 实时音视频控制台 > 应用管理 > 应用信息中查看 SDKAppID
userId	NSString	当前用户的 ID，字符串类型，只允许包含英文字母（a-z 和 A-Z）、数字（0-9）、连词符（-）和下划线（_）
userSig	NSString	腾讯云设计的一种安全保护签名，获取方式请参见 如何计算及使用 UserSig

addObserver

添加回调接口，您可以通过这个接听，监听 [TUICallObserver](#) 相关的事件回调。

```
- (void)addObserver:(id<TUICallObserver>)observer;
```

removeObserver

移除回调接口。

```
- (void)removeObserver:(id<TUICallObserver>)observer;
```

call

拨打电话（1v1通话）。

⚠ 注意：

字符串房间号目前只支持 Android 和 iOS 平台，Web、小程序、Flutter、Uniapp 等其他平台后续支持，敬请期待！

```
- (void)call:(NSString *)userId callMediaType:
(TUICallMediaType)callMediaType params:(TUICallParams *)params succ:
(TUICallSucc)succ fail:(TUICallFail)fail
```

参数如下表所示：

参数	类型	含义
userId	NSString	目标用户的 userId
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话
params	TUICallParams	通话参数扩展字段，例如：房间号、通话邀请超时时间，离线推送自定义内容等

groupCall

发起群组通话。

⚠ 注意：

使用群组通话前需要创建 IM 群组，如果已经创建，请忽略；

字符串房间号目前只支持 Android 和 iOS 平台，Web、小程序、Flutter、Uniapp 等其他平台后续支持，敬请期待！

```
- (void)groupCall:(NSString *)groupId userIdList:(NSArray <NSString *>
*)userIdList callMediaType:(TUICallMediaType)callMediaType params:
(TUICallParams *)params succ:(TUICallSucc)succ fail:(TUICallFail)fail
```

参数	类型	含义
groupId	NSString	此次群组通话的群 ID
userIdList	NSArray	目标用户的 userId 列表
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话
params	TUICallParams	通话参数扩展字段，例如：房间号、通话邀请超时时间，离线推送自定义内容等

accept

接受当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数接听来电。

```
- (void)accept:(TUICallSucc)succ fail:(TUICallFail)fail;
```

reject

拒绝当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数拒绝来电。

```
- (void)reject:(TUICallSucc)succ fail:(TUICallFail)fail;
```

ignore

忽略当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数忽略来电，此时主叫会收到 `onUserLineBusy` 的回调。

备注：如果您的业务中存在直播、会议等场景，在直播/会议中的情况时，也可以调用这个函数来忽略此次来电。

```
- (void)ignore:(TUICallSucc)succ fail:(TUICallFail)fail;
```

hangup

挂断当前通话，当您处于通话中，可以调用该函数结束通话。

```
- (void)hangup:(TUICallSucc)succ fail:(TUICallFail)fail;
```

inviteUser

邀请用户加入此次群组通话。

使用场景：一个群组通话中的用户主动邀请其他人时使用。

```
- (void)inviteUser:(NSArray<NSString*>*)userIdList params:
(TUICallParams*)params succ:(void(^)(NSArray<NSString*>
*userIdList))succ fail:(TUICallFail)fail
```

参数	类型	含义
userIdList	NSArray	目标用户的 userId 列表
params	TUICallParams	通话参数扩展字段，例如：离线推送自定义内容

joinInGroupCall

主动加入此次群组通话。

使用场景：群组内用户主动加入此次群组通话使用。

```
- (void)joinInGroupCall:(TUIRoomId*)roomId groupId:(NSString*)groupId
callMediaType:(TUICallMediaType)callMediaType succ:(TUICallSucc)succ
```

```
fail:(TUICallFail) fail;
```

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间 ID
groupId	NSString	此次群组通话的群 ID
callMediaType	TUICallMediaTy e	通话的媒体类型，比如视频通话、语音通话

switchCallMediaType

切换视频通话到语音通话。

```
- (void)switchCallMediaType:(TUICallMediaType) newType;
```

参数	类型	含义
callMediaType	TUICallMediaTy e	通话的媒体类型，比如视频通话、语音通话

startRemoteView

设置显示视频画面的 View 对象。

```
- (void)startRemoteView:(NSString *)userId videoView:(TUIVideoView  
*)videoView onPlaying:(void (^)(NSString *userId))onPlaying onLoading:  
(void (^)(NSString *userId))onLoading onError:(void (^)(NSString *userId,  
int code, NSString *errMsg))onError;
```

参数	类型	含义
userId	NSString	目标用户的 userId
videoView	TUIVideoView	待渲染的视图

stopRemoteview

停止订阅远端用户的视频数据。

```
- (void)stopRemoteView:(NSString *)userId;
```

参数	类型	含义
userId	NSString	目标用户的 userId

openCamera

开启摄像头。

```
- (void)openCamera:(TUICamera)camera videoView:(TUIVideoView *)videoView  
succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

参数	类型	含义
camera	TUICamera	前置/后置 摄像头
videoView	TUIVideoView	待渲染的视图

closeCamera

关闭摄像头。

```
- (void)closeCamera;
```

switchCamera

切换前后摄像头。

```
- (void)switchCamera:(TUICamera)camera;
```

参数	类型	含义
camera	TUICamera	前置/后置 摄像头

openMicrophone

打开麦克风。

```
- (void)openMicrophone:(TUICallSucc)succ fail:(TUICallFail)fail;
```

closeMicrophone

关闭麦克风。

```
- (void)closeMicrophone;
```

selectAudioPlaybackDevice

选择音频播放设备。

目前支持听筒、扬声器，在通话场景中，可以使用这个接口来开启/关闭免提模式。

```
- (void)selectAudioPlaybackDevice:(TUIAudioPlaybackDevice) device;
```

参数	类型	含义
device	TUIAudioPlaybac kDevice	听筒/扬声器

setSelfInfo

设置用户昵称、头像。

用户昵称不能超过500字节，用户头像必须是 URL 格式。

```
- (void)setSelfInfo:(NSString * _Nullable)nickname avatar:(NSString *  
_Nullable)avatar succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

参数	类型	含义
nickname	NSString	用户昵称
avatar	NSString	用户头像（格式为 URL）

enableMultiDeviceAbility

开启/关闭 TUICallEngine 的多设备登录模式（[群组通话版套餐](#) 支持）。

```
- (void)enableMultiDeviceAbility:(BOOL)enable succ:(TUICallSucc)succ  
fail:(TUICallFail)fail;
```

setVideoRenderParams

设置用户视频画面的渲染模式。

```
- (void)setVideoRenderParams:(NSString *)userId params:  
(TUIVideoRenderParams *)params succ:(TUICallSucc)succ fail:
```

```
(TUICallFail) fail;
```

参数	类型	含义
userId	NSString	目标用户的 userId
params	TUIVideoRenderParams	视频画面渲染参数，例如：画面旋转角度、填充模式

setVideoEncoderParams

设置视频编码的编码参数。

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

```
- (void)setVideoEncoderParams:(TUIVideoEncoderParams *)params succ:
(TUICallSucc)succ fail:(TUICallFail)fail;
```

参数	类型	含义
params	TUIVideoEncoderParams	视频编码的参数，例如：分辨率、视频宽高比

getTRTCCloudInstance

高级接口。

用于接入第三方美颜等。

```
- (TRTCCloud *)getTRTCCloudInstance;
```

setBeautyLevel

设置美颜级别，支持关闭默认美颜。

```
- (void)setBeautyLevel:(CGFloat)level succ:(TUICallSucc)succ fail:
(TUICallFail)fail;
```

参数如下表所示：

参数	类型	含义
----	----	----

level	CGFloat	美颜级别，美颜级别，取值范围 0 – 9； 0 表示关闭，9 表示效果最明显
-------	---------	--

TUICallObserver

最近更新时间：2023-12-29 16:01:31

TUICallObserver API 简介

TUICallObserver 是 TUICallEngine 对应的回调事件类，您可以通过此回调，来监听自己感兴趣的回调事件。

回调事件概览

API	描述
onError	通话过程中错误回调
onCallReceived	通话请求的回调
onCallCancelled	通话取消的回调
onCallBegin	通话接通的回调
onCallEnd	通话结束的回调
onCallMediaTypeChanged	通话媒体类型发生改变的回调
onUserReject	xxxx 用户拒绝通话的回调
onUserNoResponse	xxxx 用户不响应的回调
onUserLineBusy	xxxx 用户忙线的回调
onUserJoin	xxxx 用户加入通话的回调
onUserLeave	xxxx 用户离开通话的回调
onUserVideoAvailable	xxxx 用户是否有视频流的回调
onUserAudioAvailable	xxxx 用户是否有音频流的回调
onUserVoiceVolumeChanged	所有用户音量大小的反馈回调
onUserNetworkQualityChanged	所有用户网络质量的反馈回调
onKickedOffline	当前用户被踢下线

`onUserSigExpired`

在线时票据过期

回调事件详情

onError

错误回调。

❗ 说明

SDK 不可恢复的错误，一定要监听，并分情况给用户适当的界面提示。

```
- (void)onError:(int)code message:(NSString * _Nullable)message;
```

参数如下表所示：

参数	类型	含义
code	int	错误码
message	NSString	错误信息

onCallReceived

收到一个新的来电请求回调，被叫会收到，您可以通过监听这个事件，来决定是否显示通话接听界面。

```
- (void)onCallReceived:(NSString *)callerId calleeIdList:
(NSArray<NSString *> *)calleeIdList groupId:(NSString *
_Nullable)groupId callMediaType:(TUICallMediaType)callMediaType
userData:(NSString * _Nullable)userData
```

参数如下表所示：

参数	类型	含义
callerId	NSString	主叫 ID（邀请方）
calleeIdList	NSArray	被叫 ID 列表（被邀请方）
groupId	NSString	群组通话 ID
callMediaType	TUICallMediaType	通话的媒体类型，比如视频通话、语音通话
userData	NSString	用户增加的扩展字段，详见 TUICallParams

onCallCancelled

表示此次通话主叫取消、被叫超时、拒接等，涉及多个场景，您可以通过监听这个事件来实现类似未接来电、重置 UI 状态等显示逻辑。

- 主叫取消：主叫收到该回调（userId 为自己）；被叫收到该回调（userId 为主叫的 ID）
- 被叫超时：主叫会同时收到 [onUserNoResponse](#) 和 [onCallCancelled](#) 回调（userId 是自己的 ID）；被叫收到 [onCallCancelled](#) 回调（userId 是自己的 ID）
- 被叫拒接：主叫会同时收到 [onUserReject](#) 和 [onCallCancelled](#) 回调（userId 是自己的 ID）；被叫收到 [onCallCancelled](#) 回调（userId 是自己的 ID）
- 被叫忙线：主叫会同时收到 [onUserLineBusy](#) 和 [onCallCancelled](#) 回调（userId 是自己的 ID）
- 异常中断：被叫接收通话失败，收到该回调（userId 是自己的 ID）

```
- (void)onCallCancelled:(NSString *)userId;
```

参数如下表所示：

参数	类型	含义
userId	NSString	用户的 ID

onCallBegin

表示通话接通，主叫和被叫都可以收到，您可以通过监听这个事件来开启云端录制、内容审核等流程。

```
- (void)onCallBegin:(TUIRoomId *)roomId callMediaType:
(TUICallMediaType)callMediaType callRole:(TUICallRole)callRole;
```

参数如下表所示：

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间 ID
callMediaType	TUICallMediaType	通话的媒体类型，视频通话、语音通话
callRole	TUICallRole	角色，枚举类型：主叫、被叫

onCallEnd

表示通话挂断，主叫和被叫都可以收到，您可以通过监听这个事件来显示通话时长、通话类型等信息，或者来停止云端的录制流程。

```
- (void)onCallEnd:(TUIRoomId *)roomId callMediaType:
(TUICallMediaType)callMediaType callRole:(TUICallRole)callRole
totalTime:(float)totalTime;
```

参数如下表所示：

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间 ID
callMediaType	TUICallMediaType	通话的媒体类型，视频通话、语音通话
callRole	TUICallRole	角色，枚举类型：主叫、被叫
totalTime	float	此次通话的时长，单位：秒

⚠ 注意

客户端的事件一般都会随着杀进程等异常事件丢失掉，如果您需要通过监听通话时长来完成计费逻辑，建议使用 REST API 来完成这类流程。

onCallMediaTypeChanged

表示通话的媒体类型发生变化。

```
- (void)onCallMediaTypeChanged:(TUICallMediaType)oldCallMediaType
newCallMediaType:(TUICallMediaType)newCallMediaType;
```

参数如下表所示：

参数	类型	含义
oldCallMediaType	TUICallMediaType	旧的通话类型
newCallMediaType	TUICallMediaType	新的通话类型

onUserReject

通话被拒绝的回调，在1v1 通话中，只有主叫方会收到拒绝回调，在群组通话中，所有被邀请者都可以收到该回调。

```
- (void)onUserReject:(NSString *)userId;
```

参数如下表所示：

参数	类型	含义
userId	NSString	拒绝用户的 ID

onUserNoResponse

对方无回应的回调。

```
- (void)onUserNoResponse:(NSString *)userId;
```

参数如下表所示：

参数	类型	含义
userId	NSString	无响应用户的 ID

onUserLineBusy

通话忙线回调。

```
- (void)onUserLineBusy:(NSString *)userId;
```

onUserJoin

有用户进入此次通话的回调。

```
- (void)onUserJoin:(NSString *)userId;
```

onUserLeave

有用户离开此次通话的回调。

```
- (void)onUserLeave:(NSString *)userId;
```

onUserAudioAvailable

用户是否开启音频上行回调。

```
- (void)onUserAudioAvailable:(NSString *)userId isAudioAvailable:
```

```
(BOOL)isAudioAvailable;
```

参数如下表所示：

参数	类型	含义
userId	NSString	用户 ID
isAudioAvailable	BOOL	用户音频是否可用

onUserVideoAvailable

用户是否开启视频上行回调。

```
- (void)onUserVideoAvailable:(NSString *)userId isVideoAvailable:
(BOOL)isVideoAvailable;
```

参数如下表所示：

参数	类型	含义
userId	NSString	通话用户 ID
isVideoAvailable	BOOL	用户视频是否可用

onUserVoiceVolumeChanged

用户通话音量的回调。

```
- (void)onUserVoiceVolumeChanged:(NSDictionary <NSString *, NSNumber *>
*)volumeMap;
```

参数如下表所示：

参数	类型	含义
volumeMap	NSDictionary <NSString *, NSNumber *>	音量表，根据每个 userId 可以获取对应的音量大小，音量最小值为0，音量最大值为100
volumeMap	NSDictionary	音量表，根据每个 userId 可以获取对应的音量大小，音量最小值为0，音量最大值为100

onUserNetworkQualityChanged

用户网络质量的回调。

```
- (void)onUserNetworkQualityChanged:(NSArray<TUINetworkQualityInfo *>
*)networkQualityList;
```

参数如下表所示：

参数	类型	含义
networkQualityList	NSArray	网络状态，根据每个 userId 可以获取对应用户当前的网络质量

onKickedOffline

当前用户被踢下线：此时可以 UI 提示用户，并再次重新调用初始化。

```
- (void)onKickedOffline;
```

onUserSigExpired

在线时票据过期：此时您需要生成新的 userSig，并再次重新调用初始化。

```
- (void)onUserSigExpired;
```

类型定义

最近更新时间：2023-09-11 17:33:31

常用结构

TUICallDefine

类型	描述
TUICallParams	通话参数
TUIOfflinePushInfo	离线推送厂商配置信息

TUICommonDefine

类型	描述
TUIRoomId	通话中音视频房间 ID
TUIVideoRenderParams	视频画面的渲染参数
TUIVideoEncoderParams	视频编码参数

枚举定义

TUICallDefine

类型	说明
TUICallMediaType	通话类型
TUICallRole	通话角色
TUICallStatus	通话状态
TUICallScene	通话场景
TUICallIOSOfflinePushType	iOS 离线推送类型

TUICommonDefine

类型	说明
TUIAudioPlaybackDevice	音频设备

TUICamera	摄像头类型
TUINetworkQuality	网络质量
TUIVideoRenderParamsFillMode	视频画面填充模式
TUIVideoRenderParamsRotation	视频画面旋转方向
TUIVideoEncoderParamsResolutionMode	视频宽高比模式
TUIVideoEncoderParamsResolution	视频分辨率

TUICallParams

通话参数

参数	类型	描述
offlinePushInfo	TUIOfflinePushInfo	厂商离线推送配置信息
timeout	int	通话超时时间，默认：30s，单位：秒
userData	NSString	发起通话时自定义扩展字段，被叫端在 onCallReceived 中回调

TUIOfflinePushInfo

厂商离线推送配置信息，详情配置步骤请参考：[离线唤醒](#)

参数	类型	描述
title	NSString	离线推送展示通知栏标题
desc	NSString	离线推送展示通知栏内容
ignoreIOSBadge	BOOL	离线推送忽略 badge 计数（仅对 iOS 生效），如果设置为 true，在 iOS 接收端，这条消息不会使 APP 的应用图标未读计数增加。
iOSSound	NSString	离线推送声音设置（仅对 iOS 生效）。当 sound = IOS_OFFLINE_PUSH_NO_SOUND，表示接收时不会播放声音。当 sound = IOS_OFFLINE_PUSH_DEFAULT_SOUND，表示接收时播放系统声音。如果要自定义 iOSSound，需要先把

		语音文件链接进 Xcode 工程，然后把语音文件名（带后缀）设置给 iOSSound
androidSound	NSString	离线推送声音设置（仅对 Android 生效, IMSDK 6.1 及以上版本支持）。只有华为和谷歌手机支持设置声音提示，小米手机设置声音提示，请您参照： 小米自定义铃声 另外，谷歌手机 FCM 推送在 Android 8.0 及以上系统设置声音提示，必须调用 setAndroidFCMChannelID 设置好 channelId，才能生效
androidOPPOChannelID	NSString	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID
androidVIVOClassification	NSInteger	VIVO 推送消息分类 (待废弃接口，VIVO 推送服务于 2023 年 4 月 3 日优化消息分类规则，推荐使用 setAndroidVIVOCategory 设置消息类别)。0：运营消息 1：系统消息，默认取值为 1
androidXiaoMiChannelID	NSString	小米手机 8.0 系统及以上的渠道 ID
androidFCMChannelID	NSString	FCM 通道手机 8.0 系统及以上的渠道 ID
androidHuaWeiCategory	NSString	华为推送消息分类，详见： 华为消息分类标准
isDisablePush	BOOL	是否关闭推送（默认开启推送）
iOSPushType	TUICallIOSOfflinePushType	iOS 离线推送类型，默认：APNs

TUIRoomId

通话中音视频房间ID

- intRoomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 intRoomId 需要填写为 0。若两者都填，SDK 将优先选用 intRoomId。
- 不要混用 intRoomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 123 两个完全不同的房间。

参数	类型	描述
intRoomId	UInt32	数字房间号 取值范围 1 – 2147483647(2 ³¹ –1)

strRoomId	NSString	字符串房间号 推荐取值 限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： 大小写英文字母（a-zA-Z）； 数字（0-9）； 空格、！、#、\$、%、&、（、）、+、-、 :、;、<、=、.、>、?、@、[、]、^ 、_、{、}、 、~、,、。
-----------	----------	---

注意：
字符串房间号目前只支持 Android、iOS、Flutter、Uniapp 平台，Web、小程序等其他平台后续支持，敬请期待！

TUIVideoRenderParams

视频画面的渲染参数

参数	类型	描述
fillMode	TUIVideoRenderParamsFillMode	视频画面填充模式
rotation	TUIVideoRenderParamsRotation	视频画面旋转方向

TUIVideoEncoderParams

视频编码参数

参数	类型	描述
resolution	TUIVideoEncoderParamsResolution	视频分辨率
resolutionMode	TUIVideoEncoderParamsResolutionMode	视频宽高比

TUICallMediaType

通话类型

类型	取值	描述
----	----	----

TUICallMediaTypeUnknown	0	未知类型
TUICallMediaTypeAudio	1	语音通话
TUICallMediaTypeVideo	2	视频通话

TUICallRole

通话角色

类型	取值	描述
TUICallRoleNone	0	未知类型
TUICallRoleCall	1	主叫（邀请方）
TUICallRoleCalled	2	被叫（被邀请方）

TUICallStatus

通话状态

类型	取值	描述
TUICallStatusNone	0	未定义
TUICallStatusWaiting	1	通话等待中
TUICallStatusAccept	2	通话已接听

TUICallScene

通话场景

类型	取值	描述
TUICallSceneGroup	0	群组通话
TUICallSceneMulti	1	匿名群组通话（暂不支持，敬请期待）
TUICallSceneSingle	2	单人通话

TUICallIOSOfflinePushType

iOS离线推送类型

类型	取值	描述
TUICallIOSOfflinePushTypeAPNs	0	普通推送
TUICallIOSOfflinePushTypeVoIP	1	VoIP 推送

TUIAudioPlaybackDevice

音频设备

类型	取值	描述
TUIAudioPlaybackDeviceSpeakerphone	0	扬声器
TUIAudioPlaybackDeviceEarpiece	1	听筒

TUICamera

前置/后置 摄像头

类型	取值	描述
TUICameraFront	0	前置摄像头
TUICameraBack	1	后置摄像头

TUINetworkQuality

网络质量

类型	取值	描述
TUINetworkQualityUnknown	0	未定义
TUINetworkQualityExcellent	1	当前网络非常好

TUINetworkQualityGood	2	当前网络比较好
TUINetworkQualityPoor	3	当前网络一般
TUINetworkQualityBad	4	当前网络较差
TUINetworkQualityVbad	5	当前网络很差
TUINetworkQualityDown	6	当前网络不满足通话的最低要求

TUIVideoRenderParamsFillMode

视频画面填充模式

类型	取值	描述
TUIVideoRenderParamsFillModeFill	0	填充模式：即将画面内容居中等比缩放以充满整个显示区域，超出显示区域的部分将会被裁剪掉，此模式下画面可能不完整
TUIVideoRenderParamsFillModeFit	1	适应模式：即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边

TUIVideoRenderParamsRotation

视频画面旋转方向

类型	取值	描述
TUIVideoRenderParamsRotation_0	0	不旋转
TUIVideoRenderParamsRotation_90	1	顺时针旋转90度
TUIVideoRenderParamsRotation_180	2	顺时针旋转180度
TUIVideoRenderParamsRotation_270	3	顺时针旋转270度

TUIVideoEncoderParamsResolutionMode

视频宽高比模式

类型	取值	描述
TUIVideoEncoderParamsResolutionModeLandscape	0	横屏分辨率，例如： Resolution.Resolution_640_360 + ResolutionMode.Landscape = 640 × 360
TUIVideoEncoderParamsResolutionModePortrait	1	竖屏分辨率，例如： Resolution.Resolution_640_360 + ResolutionMode.Portrait = 360 × 640

TUIVideoEncoderParamsResolution

视频分辨率

类型	取值	描述
TUIVideoEncoderParamsResolution_640_360	108	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps
TUIVideoEncoderParamsResolution_640_480	62	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps
TUIVideoEncoderParamsResolution_960_540	110	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps
TUIVideoEncoderParamsResolution_960_720	64	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps
TUIVideoEncoderParamsResolution_1280_720	112	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps
TUIVideoEncoderParamsResolution_1920_1080	114	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps

Web

API 概览

最近更新时间：2025-05-16 17:17:42

TUICallKit（含 UI 组件）

TUICallKit API 是音视频通话组件的含 UI 接口，使用 TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景。

API	描述
<code>init</code>	初始化 TUICallKit。
<code>calls</code>	发起单人或多人通话。
<code>join</code>	主动加入通话。
<code>setCallingBell</code>	设置自定义来电铃声。
<code>setSelfInfo</code>	设置用户昵称和头像。
<code>enableMuteMode</code>	开启/关闭来电铃声。
<code>enableFloatWindow</code>	开启/关闭悬浮窗功能。
<code>enableVirtualBackground</code>	开启/关闭虚拟背景的功能按钮。
<code>setLanguage</code>	设置通话语言。
<code>hideFeatureButton</code>	隐藏按钮。
<code>setLocalViewBackgroundImage</code>	设置本地用户通话界面背景图。
<code>setRemoteViewBackgroundImage</code>	设置远端用户通话界面背景图。
<code>setLayoutMode</code>	设置通话界面布局模式。
<code>setCameraDefaultState</code>	设置摄像头默认状态。
<code>destroyed</code>	销毁 TUICallKit。
<code>getTUICallEngineInstance</code>	获取 TUICallEngine 实例。

TUICallEngine（无 UI 接口）

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 CallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

API	描述
<code>createInstance</code>	创建 TUICallEngine 实例（单例模式）。
<code>destroyInstance</code>	销毁 TUICallEngine 实例（单例模式）。
<code>on</code>	监听事件。
<code>off</code>	取消监听事件。
<code>login</code>	登录接口。
<code>logout</code>	登出接口。
<code>calls</code>	发起单人或多人通话。
<code>join</code>	主动加入通话。
<code>setSelfInfo</code>	设置用户昵称和头像。
<code>accept</code>	接听通话。
<code>reject</code>	拒绝通话。
<code>hangup</code>	结束通话。
<code>switchCallMediaType</code>	通话类型切换。
<code>startRemoteView</code>	启动远端画面渲染。
<code>stopRemoteView</code>	停止远端画面渲染。
<code>openCamera</code>	开启摄像头。
<code>closeCamara</code>	关闭摄像头。
<code>switchCamera</code>	切换前后置摄像头，注意：仅移动端支持。
<code>openMicrophone</code>	打开麦克风。
<code>closeMicrophone</code>	关闭麦克风。
<code>setVideoQuality</code>	设置视频分辨率。
<code>getDeviceList</code>	获取设备列表。

<code>switchDevice</code>	切换摄像头或麦克风设备。
<code>enableAIVoice</code>	开启/关闭 AI 降噪。
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式。
<code>setBlurBackground</code>	开关/设置背景虚化。
<code>setVirtualBackground</code>	开关/设置图片背景虚化。

TUICallEvent

TUICallEvent 是 TUICallEngine 对应的事件类，您可以通过监听如下事件，来监听自己感兴趣的事件。

EVENT	描述
<code>TUICallEvent.ERROR</code>	通话过程中错误事件。
<code>TUICallEvent.SDK_READY</code>	SDK 进入 ready 状态时收到该事件。
<code>TUICallEvent.KICKED_OUT</code>	用户重复登录，被踢下线。
<code>TUICallEvent.USER_ACCEPT</code>	xxxx 用户接通的事件。
<code>TUICallEvent.USER_ENTER</code>	xxxx 用户加入通话的事件。
<code>TUICallEvent.USER_LEAVE</code>	xxxx 用户离开通话的事件。
<code>TUICallEvent.REJECT</code>	xxxx 用户拒绝通话的事件。
<code>TUICallEvent.NO_RESP</code>	xxxx 用户不响应的事件。
<code>TUICallEvent.LINE_BUSY</code>	xxxx 用户忙线的事件。
<code>TUICallEvent.USER_VIDEO_AVAILABLE</code>	xxxx 用户是否有视频流的事件。
<code>TUICallEvent.USER_AUDIO_AVAILABLE</code>	xxxx 用户是否有音频流的事件。
<code>TUICallEvent.USER_VOICE_VOLUME</code>	所有用户音量大小的反馈事件。
<code>TUICallEvent.GROUP_CALL_INVITE_LIST_UPDATE</code>	群聊更新邀请列表收到该事件。
<code>TUICallEvent.ON_CALL_BEGIN</code>	通话接通的事件。

TUICallEvent.ON_CALL_RECEIVED	通话请求的事件。
TUICallEvent.ON_CALL_CANCELED	通话取消的事件。
TUICallEvent.CALLING_END	通话结束的事件。
TUICallEvent.DEVICED_UPDATED	设备列表更新的事件。
TUICallEvent.CALL_TYPE_CHANGED	通话的媒体类型发生改变的事件。
TUICallEvent.ON_USER_NETWORK_QUALITY_CHANGED	所有用户网络质量的事件。

TUICallKit

最近更新时间：2025-05-16 17:17:42

TUICallKit 是音视频通话组件的含 UI 接口，使用 TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景；更详细的接入步骤，详见：[快速接入 TUICallKit](#)。

- ❗ 说明：
- 为提供更优质的音视频通信能力，TUICallKit npm 包 [@tencentcloud/call-uikit-react](#)、[@tencentcloud/call-uikit-vue](#)、[@tencentcloud/call-uikit-vue2](#)、[@tencentcloud/call-uikit-vue2.6](#) 在 4.0.0 版本进行了重大升级。
 - v4.0.0+ 对接口进行了优化调整，废弃了部分接口，同步提供更简洁、稳定的新接口方案。

API 概览

API	描述
<TUICallKit />	TUICallKit 通话 UI 组件。
init	初始化 TUICallKit 组件实例。
calls	发起单人或多人通话。
join	主动加入通话。
setCallingBell	设置自定义来电铃声。
setSelfInfo	设置用户昵称和头像。
enableMuteMode	开启/关闭来电铃声。
enableFloatWindow	开启/关闭悬浮窗功能。
enableVirtualBackground	开启/关闭模糊背景的功能按钮。
setLanguage	设置 TUICallKit 组件通话语言。
hideFeatureButton	隐藏按钮。
setLocalViewBackgroundImage	设置本地用户通话界面背景图。
setRemoteViewBackgroundImage	设置远端用户通话界面背景图。

<code>setLayoutMode</code>	设置通话界面布局模式。
<code>setCameraDefaultState</code>	设置摄像头是否默认打开。
<code>destroyed</code>	销毁 TUICallKit 组件实例。
<code>getTUICallEngineInstance</code>	获取 TUICallEngine 实例。

< TUICallKit /> 组件

属性概览

属性	描述	类型	是否必填	默认值	支持 vue	支持 react
<code>allowedMinimized</code>	是否允许悬浮窗	boolean	否	false	✓	✓
<code>allowedFullScreen</code>	是否允许通话界面全屏	boolean	否	true	✓	✓
<code>videoDisplayMode</code>	通话界面显示模式	VideoDisplayMode	否	VideoDisplayMode.COVER	✓	✓
<code>videoResolution</code>	通话分辨率	VideoResolution	否	VideoResolution.RESOLUTION_480P	✓	✓
<code>beforeCalling</code>	拨打电话前与收到通话邀请前会执行此函数	function(type, error)	否	-	✓	✓
<code>afterCalling</code>	结束通话后会执行此函数	function()	否	-	✓	✓
<code>onMinimized</code>	组件切换最小化状态时会执行此函数， STATUS 值说明	function(oldStatus, newStatus)	否	-	✓	✓
<code>kickedOut</code>	组件抛出的事件，当前登录用户被踢出登录时	function()	否	-	✓	×

	会触发该事件， 通话也会自动结束					
statusChanged	组件抛出的事件，当通话状态发生变化时，会触发该事件，通话状态种类详见， STATUS 值说明	function({ oldStatus, newStatus})	否	-	✓	×

示例代码

React

```
import { TUICallKit, VideoDisplayMode, VideoResolution } from
"@tencentcloud/call-uikit-react";

<TUICallKit
  videoDisplayMode={VideoDisplayMode.CONTAIN}
  videoResolution={VideoResolution.RESOLUTION_1080P}
  beforeCalling={handleBeforeCalling}
  afterCalling={handleAfterCalling}
/>

function handleBeforeCalling(type: string, error: any) {
  console.log("[TUICallkit Demo] handleBeforeCalling:", type, error);
}
function handleAfterCalling() {
  console.log("[TUICallkit Demo] handleAfterCalling");
}
```

Vue3

```
<template>
  <TUICallKit
    :allowedMinimized="true"
    :allowedFullScreen="true"
    :videoDisplayMode="VideoDisplayMode.CONTAIN"
    :videoResolution="VideoResolution.RESOLUTION_1080P"
    :beforeCalling="beforeCalling"
```

```

      :afterCalling="afterCalling"
      :onMinimized="onMinimized"
      :kickedOut="handleKickedOut"
      :statusChanged="handleStatusChanged"
    />
  </template>

  <script lang="ts" setup>
    import { TUICallKit, TUICallKitServer, VideoDisplayMode,
    VideoResolution, STATUS } from "@tencentcloud/call-uikit-vue";

    function beforeCalling(type: string, error: any) {
      console.log("[TUICallkit Demo] beforeCalling:", type, error);
    }
    function afterCalling() {
      console.log("[TUICallkit Demo] afterCalling");
    }
    function onMinimized(oldStatus: string, newStatus: string) {
      console.log("[TUICallkit Demo] onMinimized: " + oldStatus + " -> " +
      newStatus);
    }
    function kickedOut() {
      console.log("[TUICallkit Demo] kickedOut");
    }
    function statusChanged(args: { oldStatus: string; newStatus: string; })
    {
      const { oldStatus, newStatus } = args;
      if (newStatus === STATUS.CALLING_C2C_VIDEO) {
        console.log(`[TUICallkit Demo] statusChanged: ${oldStatus} ->
        ${newStatus}`);
      }
    }
  </script>

```

Vue2.7

```

<template>
  <TUICallKit
    :allowedMinimized="true"
    :allowedFullScreen="true"
    :videoDisplayMode="VideoDisplayMode.CONTAIN"

```

```

      :videoResolution="VideoResolution.RESOLUTION_1080P"
      :beforeCalling="beforeCalling"
      :afterCalling="afterCalling"
      :onMinimized="onMinimized"
      :kickedOut="handleKickedOut"
      :statusChanged="handleStatusChanged"
    />
  </template>

  <script lang="ts" setup>
    import { TUICallKit, TUICallKitServer, VideoDisplayMode,
    VideoResolution, STATUS } from "@tencentcloud/call-uikit-vue2";

    function beforeCalling(type: string, error: any) {
      console.log("[TUICallkit Demo] beforeCalling:", type, error);
    }
    function afterCalling() {
      console.log("[TUICallkit Demo] afterCalling");
    }
    function onMinimized(oldStatus: string, newStatus: string) {
      console.log("[TUICallkit Demo] onMinimized: " + oldStatus + " -> " +
      newStatus);
    }
    function kickedOut() {
      console.log("[TUICallkit Demo] kickedOut");
    }
    function statusChanged(args: { oldStatus: string; newStatus: string; })
    {
      const { oldStatus, newStatus } = args;
      if (newStatus === STATUS.CALLING_C2C_VIDEO) {
        console.log(`[TUICallkit Demo] statusChanged: ${oldStatus} ->
        ${newStatus}`);
      }
    }
  </script>

```

Vue2.6

```

<template>
  <TUICallKit
    :allowedMinimized="true"
    :allowedFullScreen="true"

```

```
:videoDisplayMode="VideoDisplayMode.CONTAIN"
:videoResolution="VideoResolution.RESOLUTION_1080P"
:beforeCalling="beforeCalling"
:afterCalling="afterCalling"
:onMinimized="onMinimized"
:kickedOut="handleKickedOut"
:statusChanged="handleStatusChanged"
/>
</template>

<script lang="ts" setup>
import { TUICallKit, TUICallKitServer, VideoDisplayMode,
VideoResolution, STATUS } from "@tencentcloud/call-uikit-vue2.6";

function beforeCalling(type: string, error: any) {
  console.log("[TUICallkit Demo] beforeCalling:", type, error);
}
function afterCalling() {
  console.log("[TUICallkit Demo] afterCalling");
}
function onMinimized(oldStatus: string, newStatus: string) {
  console.log("[TUICallkit Demo] onMinimized: " + oldStatus + " -> " +
newStatus);
}
function kickedOut() {
  console.log("[TUICallkit Demo] kickedOut");
}
function statusChanged(args: { oldStatus: string; newStatus: string; })
{
  const { oldStatus, newStatus } = args;
  if (newStatus === STATUS.CALLING_C2C_VIDEO) {
    console.log(`[TUICallkit Demo] statusChanged: ${oldStatus} ->
${newStatus}`);
  }
}
</script>
```

TUICallKitServer API 详情

React

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-react";
```

Vue3

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-vue";
```

Vue2.7

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-vue2";
```

Vue2.6

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-vue2.6";
```

init

初始化 TUICallKit。

```
try {
  await TUICallKitServer.init({ SDKAppID, userID, userSig });
  // If you already have a tim instance in your project, you need to
  pass it in here.
  // await TUICallKitServer.init({ tim, SDKAppID, userID, userSig});
  console.log("[TUICallKit] init succeeds.");
} catch (error: any) {
  console.error(`[TUICallKit] init failed. Reason: ${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
SDKAppID	Number	是	应用的 SDKAppID，您可以在实时音视频控制台找到您的 SDKAppID，具体详见： 开通服务 。

userId	String	是	即用户名，只允许包含英文字母（a-z 和 A-Z）、数字（0-9）、连词符（-）和下划线（_）。 注意： 不支持同一个 userID 在两台不同的设备上同时进入同一个房间通话，否则会相互干扰。
userSig	String	是	使用 SecretKey 对 SDKAppID、UserID 等信息进行加密，就可以得到 userSig。 它是一个鉴权用的票据，用于腾讯云识别当前用户是否能够使用 TRTC 的服务，获取方式请参见 UserSig
tim	TencentCloudChat	否	tim 是 TencentCloudChat SDK 的实例

calls

发起单人或多人通话。

说明：
v4.0.0+ 支持。

```
try {
  await TUICallKitServer.calls({
    userIDList: ['jack', 'tom'],
    type: TUICallType.VIDEO_CALL
  });
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the groupCall API. Reason:${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
userIDList	Array<String>	是	被呼叫的用户列表。
type	TUICallType	是	通话的媒体类型，参数值说明参见 TUICallType 通话类型 。 兼容旧版本：语音通话(type = 1)、视频通话(type = 2)。
chatGroupID	String	是	与 chat 结合使用时，chat 群组的群 ID。

roomId	Number	否	数字房间号，范围 [1, 2147483647]。
strRoomID	String	否	字符串房间号。 v3.3.1+ 支持 1. roomId 与 strRoomID 是互斥的，若您选用 strRoomID，则 roomId 需要填写为 0。若两者都填，SDK 将优先选用 roomId。 2. 不要混用 roomId 和 strRoomID，因为它们之间是不互通的，例如数字 123 和字符串 "123" 是两个完全不同的房间。
timeout	Number	否	通话超时时间，默认：30s，单位：秒。设置范围：10s ~ 600s。如果传入0，默认30s。
userData	String	否	发起通话时自定义扩展字段，被呼叫用户在 ON_CALL_RECEIVED 事件中有该参数。
offlinePushInfo	Object	否	自定义离线消息推送参数。

join

加入群组中已有的音视频通话。

说明：
v4.0.0+ 支持。

```
try {
  await TUICallKitServer.join({callId: xx});
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the join API. Reason:
${error}`);
}
```

参数列表：

参数	类型	是否必填	含义
callId	string	是	此次通话的唯一 ID。

setLanguage

设置语言，目前支持：中文、英文、日文。

```
TUICallKitServer.setLanguage("zh-cn"); // "en" | "zh-cn" | "ja_JP"
```

参数如下表所示：

参数	类型	是否必填	含义
lang	String	是	语言类型 <code>en</code> 、 <code>zh-cn</code> 和 <code>ja_JP</code> 。

setSelfInfo

设置当前用户昵称和头像。

⚠ 注意：

v2.2.0+ 支持。 通话中使用该接口修改用户信息，UI 不会立即更新，需要等到下次通话才能看到变化。

```
try {
  await TUICallKitServer.setSelfInfo({ nickName: "xxx", avatar:
"http://xxx" });
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the setSelfInfo API.
Reason: ${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
nickName	String	是	自己的昵称。
avatar	String	是	自己头像地址。

setCallingBell

- 自定义用户的来电铃声。
- 仅限传入本地 MP3 格式的文件地址，需要确保该文件目录是应用可以访问的。
- 使用 import 方式引入铃声文件。
- 如需恢复默认铃声，`filePath` 传空即可。

```
try {
  await TUICallKitServer.setCallingBell(filePath?: string)
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the setCallingBell API.
Reason: ${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
filePath	String	是	铃声文件地址。

enableFloatWindow

开启/关闭悬浮窗功能。默认为 false，通话界面左上角的悬浮窗按钮隐藏，设置为 true 后显示。

```
try {
  await TUICallKitServer.enableFloatWindow(enable: Boolean)
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the enableFloatWindow API.
Reason: ${error}`);
}
```

enableMuteMode

开启/关闭来电铃声。开启后，收到通话请求时，不会播放来电铃声。

```
try {
  await TUICallKitServer.enableMuteMode(enable: boolean)
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the enableMuteMode API.
Reason: ${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
----	----	------	----

enable	Boolean	是	开启/关闭来电铃声。默认 false。
--------	---------	---	---------------------

enableVirtualBackground

开启/关闭模糊背景功能。如果想设置图片背景模糊参见 [Web](#)。通过调用接口，您可以在 UI 上显示模糊背景的功能按钮，点击按钮可直接启用模糊背景功能。

注意：
v3.2.4+ 支持。

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-react";
const enable = true;
TUICallKitServer.enableVirtualBackground(enable);
```

参数列表：

参数	类型	是否必填	含义
enable	boolean	是	- enable = true 显示模糊背景按钮。 - enable = false 不显示模糊背景按钮。

destroyed

销毁 TUICallKit 实例。

该方法不会自动退出 `tim`，需要手动退出 `tim.logout()`。

```
try {
  await TUICallKitServer.destroyed();
} catch (error: any) {
  alert(`[TUICallKit] Failed to call the destroyed API. Reason: ${error}`);
}
```

hideFeatureButton

隐藏功能按钮，目前仅支持 摄像头，麦克风和切换前后置按钮。

注意：
v3.2.9+ 支持。

Vue

```
import { TUICallKitServer, FeatureButton } from "@tencentcloud/call-uikit-vue";

TUICallKitServer.hideFeatureButton(FeatureButton.Camera);
```

React

```
import { TUICallKitServer, FeatureButton } from "@tencentcloud/call-uikit-react";

TUICallKitServer.hideFeatureButton(FeatureButton.Camera);
```

参数列表：

参数	类型	是否必填	含义
buttonName	FeatureButton	是	按钮名称。

setLocalViewBackgroundImage

设置本地用户通话背景。

注意：
v3.2.9+ 支持。

```
TUICallKitServer.setLocalViewBackgroundImage('http://xxx.png');
```

参数列表：

参数	类型	是否必填	含义
url	string	是	图片地址（支持本地路径和网络地址）。

setRemoteViewBackgroundImage

设置远端用户通话背景。

注意：
v3.2.9+ 支持。

```
const userId = 'xxx';
const url = 'http://xxx.png';
TUICallKitServer.setRemoteViewBackgroundImage(userId, url);
```

参数列表：

参数	类型	是否必填	含义
userId	string	是	远端用户 userId，设置为 '*' 表示对所有远端用户生效。
url	string	是	图片地址（支持本地路径和网络地址）。

setLayoutMode

设置通话界面布局模式。

注意：
v3.3.0+ 支持。

Vue

```
import { TUICallKitServer, LayoutMode } from "@tencentcloud/call-uikit-vue";
TUICallKitServer.setLayoutMode(LayoutMode.LocalInLargeView);
```

React

```
import { TUICallKitServer, LayoutMode } from "@tencentcloud/call-uikit-react";
TUICallKitServer.setLayoutMode(LayoutMode.LocalInLargeView);
```

参数列表：

参数	类型	是否必填	含义
layoutMode	Layout Mode	是	用户流的布局模式。

setCameraDefaultState

设置摄像头是否默认打开。

注意：
v3.3.0+ 支持。

```
TUICallKitServer.setCameraDefaultState(true);
```

参数列表：

参数	类型	是否必填	含义
isOpen	boolean	是	是否开启摄像头。

getTUICallEngineInstance

获取 TUICallEngine 实例。

注意：
v2.4.3+ 支持。

```
TUICallKitServer.getTUICallEngineInstance();
```

TUICallKit 类型定义

videoDisplayMode

画面显示模式 videoDisplayMode 属性有三个值：

- VideoDisplayMode.CONTAIN
- VideoDisplayMode.COVER
- VideoDisplayMode.FILL ，默认值是 VideoDisplayMode.COVER 。

属性	值	描述
videoDisplayMode	VideoDisplayMode.CONTAIN	• 优先保证视频内容全部显示。 • 视频尺寸等比缩放，直至视频窗口的一边与视窗边框对齐。 • 如果视频尺寸与显示视窗尺寸不一致，在保持长宽比的前提下，将视频进行缩放后填满视窗，缩放后的视频四周会有一圈黑边。
	VideoDisplayMode.COVER	• 优先保证视窗被填满。 • 视频尺寸等比缩放，直至整个视窗被视频填满。 • 如果视频长宽与显示窗口不同，则视频流会按照显示窗口的比例进行周边裁剪或图像拉伸后填满视窗。
	VideoDisplayMode.FILL	• 保证视窗被填满的同时保证视频内容全部显示，但是不保证视频尺寸比例不变。 • 视频的宽高会被拉伸至和视窗尺寸一致。

videoResolution

分辨率 `videoResolution` 有三个值：

- `VideoResolution.RESOLUTION_480P`
- `VideoResolution.RESOLUTION_720P`
- `VideoResolution.RESOLUTION_1080P`，默认值是 `VideoResolution.RESOLUTION_480P`。

分辨率说明：

视频 Profile	分辨率（宽 × 高）	帧率（fps）	码率（kbps）
480p	640 × 480	15	900
720p	1280 × 720	15	1500
1080p	1920 × 1080	15	2000

常见问题：

- iOS 13&14不支持编码高于 720P 的视频，建议在这两个系统版本限制最高采集 720P。参见 [iOS Safari 已知问题 case 12](#)。
- Firefox 不支持自定义视频帧率（默认为 30fps）。
- 受系统性能占用，摄像头采集能力和浏览器限制等因素的影响，视频分辨率、帧率、码率的实际值不一定能够完全匹配设定值，在这种情况下，浏览器会自动调整 Profile 尽可能匹配设定值。

STATUS

属性值	描述
STATUS.IDLE	闲置状态。
STATUS.BE_INVITED	收到通话邀请。
STATUS.DIALING_C2C	正在 1v1 呼叫。
STATUS.DIALING_GROUP	正在群组呼叫。
STATUS.CALLING_C2C_AUDIO	正在 1v1 语音通话。
STATUS.CALLING_C2C_VIDEO	正在 1v1 视频通话。
STATUS.CALLING_GROUP_AUDIO	正在群组语音通话。
STATUS.CALLING_GROUP_VIDEO	正在群组视频通话。

TUICallType

TUICallType 类型	描述
TUICallType.AUDIO_CALL	语音通话。
TUICallType.VIDEO_CALL	视频通话。

offlinePushInfo

参数	类型	是否必填	含义
offlinePushInfo.title	String	否	离线推送标题（选填）。
offlinePushInfo.description	String	否	离线推送内容（选填）。
offlinePushInfo.androidOPPOChannelID	String	否	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID（选填）。

offlinePushInfo.extension	String	否	离线推送透传内容，可以用于设置 Android Notification 模式 和 VoIP 模式 。默认： Notification 模式 ，会是系统的通知； VoIP 模式 是需要传字段。
offlinePushInfo.ignoreIOSBadge	Boolean	否	离线推送忽略 badge 计数（仅对 iOS 生效），如果设置为 true，在 iOS 接收端，这条消息不会使 APP 的应用图标未读计数增加。 v3.3.0+ 支持。
offlinePushInfo.iOSSound	String	否	离线推送声音设置（仅对 iOS 生效）。 v3.3.0+ 支持。
offlinePushInfo.androidSound	String	否	离线推送声音设置（仅对 Android 生效）。 v3.3.0+ 支持。
offlinePushInfo.androidVIVOClassification	Number	否	VIVO 推送消息分类。 v3.3.0+ 支持。
offlinePushInfo.androidXiaoMiChannelID	String	否	小米手机 8.0 系统及以上的渠道 ID。 v3.3.0+ 支持。
offlinePushInfo.androidFCMChannelID	String	否	FCM 通道手机 8.0 系统及以上的渠道 ID。 v3.3.0+ 支持。
offlinePushInfo.androidHuaWeiCategory	String	否	华为推送消息分类。 v3.3.0+ 支持。
offlinePushInfo.isDisablePush	Boolean	否	是否关闭推送（默认开启推送）。 v3.3.0+ 支持。
offlinePushInfo.iOSPushType	Number	否	iOS 离线推送类型。0-普通推送；1-VoIP推送。默认：0。 v3.3.0+ 支持。

Android Notification 模式

```
const extension = {
  timPushFeatures: {
    fcmPushType: 0, // 0, VoIP; 1, notification
  }
};
offlinePushInfo.extension = JSON.stringify(extension);
```

Android VoIP 模式

```
const extension = {
  timPushFeatures: {
    fcmPushType: 0, // 0, data; 1, notification
    fcmNotificationType: 1, // 0, TIMPush implementation; 1, business
    implementation after transparent transmission
  }
};
offlinePushInfo.extension = JSON.stringify(extension);
```

FeatureButton

FeatureButton 类型	描述
FeatureButton.Camera	摄像头按钮。
FeatureButton.Microphone	麦克风按钮。
FeatureButton.SwitchCamera	切换前后置摄像头按钮。
FeatureButton.InviteUser	邀请他人按钮。

LayoutMode

LayoutMode 类型	描述
LayoutMode.LocalInLargeView	本地用户在大窗显示。
LayoutMode.RemoteInLargeView	远端用户在大窗显示。

废弃接口

<TUICallKitMini />

<TUICallKitMini/> : TUICallKit UI 通话悬浮窗，当 <TUICallKit :allowedMinimized="true"/> 时，<TUICallKitMini/> 必须被放置在页面中。

❗ 说明：

- <TUICallKitMini/> UI 通话悬浮窗，≥ v2.3.2 废弃，不再提供该组件。

- 由 `<TUICallKit />` UI 通话组件主体，实现悬浮窗样式，减少使用复杂度。

属性	描述	类型	是否必填	默认值	支持 vue	支持 react
@kicked-out	 注意： ≥ v2.3.2 废弃 组件抛出的事件，当前登录用户被踢出登录时会触发该事件，通话也会自动结束	function()	否	—	✓	×
@status-changed	 注意： ≥ v2.3.2 废弃 组件抛出的事件，当通话状态发生变化时，会触发该事件，通话状态种类详见，STATUS 值说明	function({ oldStatus, newStatus })	否	—	✓	×

call

拨打 1v1 通话，建议使用 **calls** 接口。

```
import { TUICallKitServer, TUICallType } from '@tencentcloud/call-uikit-react';
try {
  await TUICallKitServer.call({
    userID: 'mike', // username being called
    type: TUICallType.VIDEO_CALL,
```

```
});
} catch (error: any) {
  console.error(`[TUICallKit] Call failed. Reason: ${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
userID	String	是	被呼叫用户的用户名。
type	TUICallType	是	通话的媒体类型，参数值说明参见 TUICallType 通话类型 。 兼容旧版本：语音通话(type = 1)、视频通话(type = 2)。
roomID	Number	否	数字房间号，范围 [1, 2147483647]。
strRoomID	String	否	字符串房间号。 v3.3.1+ 支持 1. roomID 与 strRoomID 是互斥的，若您选用 strRoomID，则 roomID 需要填写为 0。若两者都填，SDK 将优先选用 roomID。 2. 不要混用 roomID 和 strRoomID，因为它们之间是不互通的，例如数字 123 和字符串 "123" 是两个完全不同的房间。
timeout	Number	否	通话超时时间，默认：30s，单位：秒。timeout = 0，设置为不超时。
userData	String	否	发起通话时自定义扩展字段，被呼叫用户在 ON_CALL_RECEIVED 事件中有该参数。
offlinePushInfo	Object	否	自定义离线消息推送参数。

groupCall

发起群组通话，建议使用 `calls` 接口。

```
try {
  await TUICallKitServer.groupCall({
    userIDList: ['jack', 'tom'],
    groupID: "xxx",
    type: TUICallType.VIDEO_CALL
  });
} catch (error: any) {
```

```
console.error(`[TUICallKit] Failed to call the groupCall API.
Reason:${error}`);
}
```

参数如下表所示：

参数	类型	是否必填	含义
userIDList	Array<String>	是	被呼叫的用户列表。
type	TUICallType	是	通话的媒体类型，参数值说明参见 TUICallType 通话类型 。 兼容旧版本：语音通话(type = 1)、视频通话(type = 2)。
groupID	String	是	通话群组 ID，groupID 的创建可参见 chat-createGroup API 。
roomID	Number	否	数字房间号，范围 [1, 2147483647]。
strRoomID	String	否	字符串房间号。 v3.3.1+ 支持 1. roomID 与 strRoomID 是互斥的，若您选用 strRoomID，则 roomID 需要填写为 0。若两者都填，SDK 将优先选用 roomID。 2. 不要混用 roomID 和 strRoomID，因为它们之间是不互通的，例如数字 123 和字符串 "123" 是两个完全不同的房间。
timeout	Number	否	通话超时时间，默认：30s，单位：秒。timeout = 0，设置为不超时。
userData	String	否	发起通话时自定义扩展字段，被呼叫用户在 ON_CALL_RECEIVED 事件中有该参数。
offlinePushInfo	Object	否	自定义离线消息推送参数。

joinInGroupCall

加入群组中已有的音视频通话，**建议使用 join 接口**。

⚠ 注意：
v3.1.2+ 支持。

说明：

- 加入群组中已有的音视频通话前，需要提前创建或加入IM 群组，并且群组中已有用户在通话中，如果已经创建，请忽略。
- 群组的创建详见：[IM 群组管理](#)，或者您也可以直接使用 [IM TUIKit](#)，一站式集成聊天、通话等场景。

```
try {
  const params = {
    type: TUICallType.VIDEO_CALL,
    groupID: "xxx",
    roomID: 0,
  };
  await TUICallKitServer.joinInGroupCall(params);
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the enableMuteMode API.
Reason: ${error}`);
}
```

参数列表：

参数	类型	是否必填	含义
type	TUICallType	是	通话的媒体类型，例如视频通话、语音通话。
groupID	string	是	此次群组通话的群 ID。
roomID	number	是	此次通话的音视频房间 ID。

TUICallEngine

最近更新时间：2025-05-16 17:17:42

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

- ❗ 说明：
- 为提供更优质的音视频通信能力，原 `tuicall-engine-webrtc` 包已正式停止维护，建议开发者迁移至 `@tencentcloud/call-engine-js` 新包。
 - 新包对接口进行了优化调整，部分旧版接口不再兼容，同步提供更简洁、稳定的新接口方案。

API 概览

API	描述
<code>createInstance</code>	创建 TUICallEngine 实例（单例模式）。
<code>destroyInstance</code>	销毁 TUICallEngine 实例（单例模式）。
<code>on</code>	监听事件。
<code>off</code>	取消监听事件。
<code>login</code>	登录接口。
<code>logout</code>	登出接口。
<code>setSelfInfo</code>	设置用户昵称和头像。
<code>calls</code>	发起单人或多人通话。
<code>join</code>	主动加入通话。
<code>accept</code>	接听通话。
<code>reject</code>	拒绝通话。
<code>hangup</code>	结束通话。
<code>startRemoteView</code>	启动远端画面渲染。
<code>stopRemoteView</code>	停止远端画面渲染。
<code>openCamera</code>	开启摄像头。

<code>closeCamara</code>	关闭摄像头。
<code>switchCamera</code>	切换前后置摄像头，仅移动端支持。
<code>openMicrophone</code>	打开麦克风。
<code>closeMicrophone</code>	关闭麦克风。
<code>setVideoQuality</code>	设置视频分辨率。
<code>getDeviceList</code>	获取设备列表。
<code>switchDevice</code>	切换摄像头或麦克风设备。
<code>enableAIVoice</code>	开启/关闭 AI 降噪。
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式。
<code>setBlurBackground</code>	开关/设置背景虚化。
<code>setVirtualBackground</code>	开关/设置图片背景虚化。

废弃接口

API	描述
<code>call</code>	拨打电话（1v1通话），建议使用 calls 接口。
<code>groupCall</code>	发起群组通话，建议使用 calls 接口。
<code>startLocalView</code>	启动本地画面渲染，建议使用 openCamera。
<code>stopLocalView</code>	停止本地画面渲染，建议使用 closeCamera。
<code>joinInGroupCall</code>	主动加入此次群组通话，建议使用 join 接口。
<code>switchCallMediaType</code>	通话类型切换。

TUICallEvent

最近更新时间：2024-06-28 11:51:31

TUICallEvent API 简介

TUICallEvent API 是音视频通话组件的**事件接口**。

事件列表

EVENT	描述
TUICallEvent.ERROR	通话过程中错误事件
TUICallEvent.SDK_READY	SDK 进入 ready 状态时收到该事件
TUICallEvent.KICKED_OUT	用户重复登录，被踢下线
TUICallEvent.USER_ACCEPT	xxxx 用户接通的事件
TUICallEvent.USER_ENTER	xxxx 用户加入通话的事件
TUICallEvent.USER_LEAVE	xxxx 用户离开通话的事件
TUICallEvent.REJECT	xxxx 用户拒绝通话的事件
TUICallEvent.NO_RESP	xxxx 用户不响应的事件
TUICallEvent.LINE_BUSY	xxxx 用户忙线的事件
TUICallEvent.USER_VIDEO_AVAILABLE	xxxx 用户是否有视频流的事件
TUICallEvent.USER_AUDIO_AVAILABLE	xxxx 用户是否有音频流的事件
TUICallEvent.USER_VOICE_VOLUME	所有用户音量大小的反馈事件
TUICallEvent.GROUP_CALL_INVITE_LIST_UPDATE	群聊更新邀请列表收到该事件
TUICallEvent.ON_CALL_BEGIN	通话接通的事件，v1.4.6+ 支持
TUICallEvent.INVITED	通话请求的事件， 后续废弃，建议使用 TUICallEvent.ON_CALL_RECEIVED

<code>TUICallEvent.ON_CALL_RECEIVED</code>	通话请求的事件，v1.4.6+ 支持
<code>TUICallEvent.CALLING_CANCEL</code>	通话取消的事件， 后续废弃，建议使用 <code>TUICallEvent.ON_CALL_CANCELED</code>
<code>TUICallEvent.ON_CALL_CANCELED</code>	通话取消的事件，v1.4.6+ 支持
<code>TUICallEvent.ON_CALL_BEGIN</code>	通话接通时抛出的事件
<code>TUICallEvent.CALLING_END</code>	通话结束的事件
<code>TUICallEvent.DEVICED_UPDATED</code>	设备列表更新的事件
<code>TUICallEvent.CALL_TYPE_CHANGED</code>	通话的媒体类型发生改变的回调
<code>TUICallEvent.ON_USER_NETWORK_QUALITY_CHANGED</code>	所有用户网络质量的事件，v3.0.7+ 支持

ERROR

通话过程中错误事件，可以通过监听该事件，捕获通话过程中的内部错误。

```
let onError = function(error) {
  console.log(error.code, error.msg);
};
tuiCallEngine.on(TUICallEvent.ERROR, onError);
```

参数如下表所示：

参数	类型	含义
code	Number	错误码
msg	String	错误信息

SDK_READY

TUICallEngine 依赖 [@tencentcloud/chat](#) SDK，登录成功后才会出发 `SDK_READY` 事件，然后才能使用 SDK 的各项功能。

```
let onSDKReady = function(event) {
  console.log(event);
};
```

```
};  
tuiCallEngine.on(TUICallEvent.SDK_READY, onSDKReady);
```

KICKED_OUT

当前用户被踢下线，此时可以 UI 提示用户，并再次重新调用登录。

```
let handleOnKickedOut = function(event) {  
  console.log(event);  
};  
tuiCallEngine.on(TUICallEvent.KICKED_OUT, handleOnKickedOut);
```

USER_ACCEPT

如果有用户接通，其他用户都会抛出该事件，userID 就是接通的用户名。

- 1v1 通话下：被叫接通，主叫会抛出该事件。
- 群组通话下：A 呼叫 B/C，B 接通，A/C 都会抛出该事件，事件的 userID 都是 B。同理 C 接通，A、B 都会抛出该事件，事件的 userID 都是 C。

```
let handleUserAccept = function(event) {  
  console.log(event.userID);  
};  
tuiCallEngine.on(TUICallEvent.USER_ACCEPT, handleUserAccept);
```

参数如下表所示：

参数	类型	含义
userID	String	接通通话的用户 ID

USER_ENTER

如果有用户进入通话，其他用户都会抛出该事件，userID 就是进入通话的用户名。

```
let handleUserEnter = function(event) {  
  console.log(event.userID);  
};  
tuiCallEngine.on(TUICallEvent.USER_ENTER, handleUserEnter);
```

参数如下表所示：

参数	类型	含义
userID	String	进入通话的用户 ID

USER_LEAVE

有用户离开此次通话时，**通话中的其他用户**都会抛出该事件，userID 就是离开通话的用户名。

```
let handleUserLeave = function(event) {  
  console.log(event.userID);  
};  
tuiCallEngine.on(TUICallEvent.USER_LEAVE, handleUserLeave);
```

参数如下表所示：

参数	类型	含义
userID	String	离开当前通话的用户 ID

REJECT

通话被拒绝时抛出该事件

1. 在 1v1 通话中，只有主叫方会收到拒绝事件，userID 就是被叫用户名。
2. 在群组通话中，当有被邀请者拒绝通话时，群组通话的其他人都会抛出该事件，userID 就是拒绝通话的用户名。

```
let handleInviteeReject = function(event) {  
  console.log(event.userID);  
};  
tuiCallEngine.on(TUICallEvent.REJECT, handleInviteeReject);
```

参数如下表所示：

参数	类型	含义
userID	String	拒绝通话的用户 ID

NO_RESP

被呼叫者没有回应时，其他通话用户都会抛出该事件。

- 在 1v1 通话中，只有主叫方会抛出无人应答的事件。例如 A 邀请 B，B 不应答，A 抛出该事件。

- 在群组通话中，当有被邀请者不回应时，群组通话的其他人都会抛出该事件。例如 A 邀请 B、C 进入通话，B 不应答，A、C 均会抛出该事件。

```
let handleNoResponse = function(event) {
  console.log(event.sponsor, event.userIDList);
};
tuiCallEngine.on(TUICallEvent.NO_RESP, handleNoResponse);
```

参数如下表所示：

参数	类型	含义
sponsor	String	发起方的用户 ID
userIDList	Array<String>	无响应的用户 ID 列表

LINE_BUSY

通话忙线事件。例如：B 正在通话时，此时 A 呼叫 B，A 会抛出事件。

```
let handleLineBusy = function(event) {
  console.log(event);
};
tuiCallEngine.on(TUICallEvent.LINE_BUSY, handleLineBusy);
```

参数如下表所示：

参数	类型	含义
userID	String	忙线用户 ID

USER_VIDEO_AVAILABLE

视频通话中一个用户开启/关闭了摄像头，通话的其他用户都会抛出该事件。例如：A 和 B 在进行视频通话，A 开/关摄像头，B 会抛出该事件。

```
let handleUserVideoChange = function(event) {
  console.log(event.userID, event.isVideoAvailable);
};
tuiCallEngine.on(TUICallEvent.USER_VIDEO_AVAILABLE,
  handleUserVideoChange);
```

参数如下表所示：

参数	类型	含义
userID	String	开/关摄像头的用户 ID
isVideoAvailable	Boolean	true 用户打开摄像头；false 用户关闭摄像头

USER_AUDIO_AVAILABLE

音视频通话中一个用户开启/关闭了麦克风，通话的其他用户都会抛出该事件。例如：A 和 B 在进行音视频通话，A 开/关麦克风，B 会抛出该事件。

```
let handleUserAudioChange = function(event) {  
  console.log(event.userID, event.isAudioAvailable);  
};  
tuiCallEngine.on(TUICallEvent.USER_AUDIO_AVAILABLE,  
handleUserAudioChange);
```

参数如下表所示：

参数	类型	含义
userID	String	开/关麦克风的用户 ID
isAudioAvailable	Boolean	true 用户打开麦克风；false 用户关闭麦克风

USER_VOICE_VOLUME

音视频通话中用户音量变化时，通话的其他用户都会抛出该事件。例如：A 和 B 在进行音视频通话，A 音量变化，B 会抛出该事件。

```
let handleUserVoiceVolumeChange = function(event) {  
  console.log(event.volumeMap);  
};  
tuiCallEngine.on(TUICallEvent.USER_VOICE_VOLUME,  
handleUserVoiceVolumeChange);
```

参数如下表所示：

参数	类型	含义
volumeMap	Array<Object>	音量表，根据每个 userid 可以获取对应的音量大小，音量范

>

围: [0, 100]

GROUP_CALL_INVITEE_LIST_UPDATE

群聊更新邀请列表收到该事件。

```
let handleGroupInviteeListUpdate = function(event) {  
    console.log(event.userIDList);  
};  
tuiCallEngine.on(TUICallEvent.GROUP_CALL_INVITEE_LIST_UPDATE,  
handleGroupInviteeListUpdate);
```

参数如下表所示:

参数	类型	含义
userIDList	Array<String>	群组更新邀请列表

INVITED

收到一个新的来电请求事件, 被叫会收到, 您可以通过监听这个事件, 来决定是否显示通话接听界面。

注意:
计划后续版本废弃, 建议使用: [ON_CALL_RECEIVED](#)。

ON_CALL_RECEIVED

收到一个新的来电请求事件, 被叫会收到, 您可以通过监听这个事件, 来决定是否显示通话接听界面。

注意:
v1.4.6+ 版本支持。

```
let handleOnCallReceived = function(event) {  
    console.log(event);  
};  
tuiCallEngine.on(TUICallEvent.ON_CALL_RECEIVED, handleOnCallReceived);
```

参数如下表所示:

参数	类型	含义
----	----	----

sponsor	String	邀请者
userIDList	Array<String>	同时还被邀请的人
isFromGroup	Boolean	是否是群组通话
inviteData	Object	呼叫数据
inviteID	String	邀请 ID，标识一次邀请
userData	String	扩展字段：用于在邀请信令中增加扩展信息
callId	String	本次通话的唯一 ID
roomId	Number	此次通话的音视频房间 ID
callMediaType	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，枚举类型：主叫、被叫

CALLING_CANCEL

如果通话未建立，都会抛出该事件。您可以通过监听该事件来实现类似未接来电、重置 UI 状态等显示逻辑。通话未建立的场景如下：

⚠ 注意：

计划后续版本废弃，建议使用：[ON_CALL_CANCELED](#)。

ON_CALL_CANCELED

如果通话未建立，都会抛出该事件。您可以通过监听该事件来实现类似未接来电、重置 UI 状态等显示逻辑。通话未建立的场景如下：

- 主叫取消：主叫抛出该事件，userID 为主叫；被叫也会抛出该事件，userID 为被叫；
- 被叫超时：主叫会同时抛出 [NO_RESP](#) 和 [CALLING_CANCEL](#) 事件，userID 为主叫；被叫抛出 [CALLING_CANCEL](#) 事件，userID 为被叫；
- 被叫拒接：主叫会同时抛出 [REJECT](#) 和 [CALLING_CANCEL](#) 事件，userID 为主叫；被叫抛出 [CALLING_CANCEL](#) 事件，userID 为被叫；
- 被叫忙线：主叫会同时抛出 [LINE_BUSY](#) 和 [CALLING_CANCEL](#) 事件，userID 为主叫；被叫抛出 [CALLING_CANCEL](#) 事件，userID 为被叫；

⚠ 注意：

v1.4.6+ 版本支持。

```
let handleOnCallCanceled = function(event) {  
    console.log(event.userID);  
};  
tuiCallEngine.on(TUICallEvent.ON_CALL_CANCELED, handleOnCallCanceled);
```

参数如下表所示：

参数	类型	含义
userID	String	通话取消的用户 ID
callId	String	本次通话的唯一 ID
roomId	Number	此次通话的音视频房间 ID
callMediaType	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，枚举类型：主叫、被叫

ON_CALL_BEGIN

表示通话接通，主叫和被叫都会抛出该事件，您可以通过监听这个事件来开启云端录制、内容审核等流程。

⚠ 注意：**v1.4.6+ 版本支持。**

```
let handleOnCallBegin = function(event) {  
    console.log(event);  
};  
tuiCallEngine.on(TUICallEvent.ON_CALL_BEGIN, handleOnCallBegin);
```

参数如下表所示：

参数	类型	含义
callId	String	本次通话的唯一 ID
roomId	Number	此次通话的音视频房间 ID

callMedia Type	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，类型：主叫、被叫

CALLING_END

通话结束时，主叫和被叫都会抛出该事件。您可以通过监听这个事件来显示通话时长、通话类型等信息，或者来停止云端的录制流程。

```
let handleCallingEnd = function(event) {
  console.log(event.userID, event.);
};
tuiCallEngine.on(TUICallEvent.CALLING_END, handleCallingEnd);
```

参数如下表所示：

参数	类型	含义
roomId	Number	此次通话的音视频房间 ID
callMediaType	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，枚举类型：主叫 ('inviter')、被叫 ('invitee')、未知 ('')
totalTime	Number	此次通话的时长，单位: 秒
userID	String	通话结束的 userID。
callId	String	本次通话的唯一 ID。 v1.4.6+ 版本支持
callEnd	Number	此次通话的时长（ 后续将废弃，请使用 totalTime ），单位: 秒

DEVICED_UPDATED

设备列表更新时抛出该事件。

```
let handleDeviceUpdated = function({ microphoneList, cameraList,
currentMicrophoneID, currentCameraID }) {
  console.log(microphoneList, cameraList, currentMicrophoneID,
currentCameraID);
};
```

```
};
tuiCallEngine.on(TUICallEvent.DEVICED_UPDATED, handleDeviceUpdated);
```

CALL_TYPE_CHANGED

通话类型切换时抛出该事件。

```
let handleCallTypeChanged = function({ oldCallType, newCallType }) {
  console.log(oldCallType, newCallType);
};
tuiCallEngine.on(TUICallEvent.CALL_TYPE_CHANGED, handleDeviceUpdated);
```

参数如下表所示：

参数	类型	含义
oldCallType	Number	旧通话类型
newCallType	Number	新通话类型

ON_USER_NETWORK_QUALITY_CHANGED

所有用户网络质量的事件

注意：
v3.0.7+ 支持。

```
let handleOnUserNetworkQualityChange = function(event) {
  console.log(event.networkQualityList);
};
tuiCallEngine.on(TUICallEvent.ON_USER_NETWORK_QUALITY_CHANGED,
handleOnUserNetworkQualityChange);
```

参数如下表所示：

参数	类型	含义
networkQualityList	Array<Object>	网络状态，根据 userID 可以获取对应用户当前的网络质量（仅表示本地的上下行）。例如： <pre>networkQualityList: [{ userId: quality }]</pre> 网络质量说明： quality = 0，网络状况未知

	quality = 1, 网络状况极佳 quality = 2, 网络状况较好 quality = 3, 网络状况一般 quality = 4, 网络状况差 quality = 5, 网络状况极差 quality = 6, 网络链接已断开
--	--

小程序

API 概览

最近更新时间：2025-05-16 17:17:42

TUICallKit (含 UI 接口)

TUICallKit API 是音视频通话组件的含 UI 接口，使用 TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景。

API	描述
init	初始化 TUICallKit。
calls	发起单人或多人通话。
join	主动加入通话。
setSelfInfo	设置用户的头像、昵称。
setCallingBell	设置自定义来电铃声。
setLogLevel	设置日志级别。
enableFloatWindow	开启/关闭悬浮窗功能。
enableMuteMode	开启/关闭来电铃声。
joinInGroupCall	加入群组中已有的音视频通话。
hideFeatureButton	隐藏按钮。
setLocalViewBackgroundImage	设置本地用户通话界面背景图。
setRemoteViewBackgroundImage	设置远端用户通话界面背景图。
setLayoutMode	设置通话界面布局模式。
setCameraDefaultState	设置摄像头是否默认打开。
destroyed	销毁 TUICallKit。
getTUICallEngineInstance	获取 TUICallEngine 实例。

TUICallEngine (无 UI 接口)

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

API	描述
<code>createInstance</code>	创建 TUICallEngine 实例（静态方法）。
<code>destroyInstance</code>	销毁 TUICallEngine 实例（静态方法）。
<code>on</code>	增加事件监听。
<code>off</code>	取消事件监听。
<code>calls</code>	发起单人或多人通话。
<code>join</code>	主动加入通话。
<code>accept</code>	接听通话。
<code>reject</code>	拒绝通话。
<code>hangup</code>	结束通话。
<code>switchCallMediaType</code>	切换通话媒体类型，比如视频通话切音频通话。
<code>openCamera</code>	开启摄像头。
<code>closeCamera</code>	关闭摄像头。
<code>switchCamera</code>	切换前后摄像头。
<code>openMicrophone</code>	打开麦克风。
<code>closeMicrophone</code>	关闭麦克风。
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/扬声器）。
<code>setSelfInfo</code>	设置用户的头像、昵称。

TUICallEvent

TUICallEvent 是 TUICallEngine 对应的事件类，您可以通过监听如下事件，来监听自己感兴趣的事件。

事件名	说明
<code>ON_CALL_RECEIVED</code>	通话请求的事件。

USER_ACCEPT	用户接听。
USER_ENTER	用户进入通话。
USER_LEAVE	用户离开通话。
USER_UPDATE	用户更新。
REJECT	用户拒绝通话。
NO_RESP	用户无响应。
LINE_BUSY	用户忙线。
ON_CALL_CANCELED	通话未建立时，通话的各端都会抛出该事件。
ON_CALL_BEGIN	通话接通时抛出的事件。
CALL_END	通话结束。
SDK_READY	SDK Ready 回调。
KICKED_OUT	被踢下线。
CALL_MODE	切换通话模式。
ERROR	错误信息。
ON_USER_NETWORK_QUALITY_CHANGED	所有用户网络质量的事件。

关键常量定义

CALL_STATUS

参数	说明
CALL_STATUS.IDLE	默认。
CALL_STATUS.CALLING	呼叫中/被呼叫中。
CALL_STATUS.CONNECTED	接通中。

CallMediaType

参数	说明
----	----

CallMediaType.AUDIO	音频。
CallMediaType.VIDEO	视频。

CALL_MEDIA_TYPE

参数	说明
CALL_MEDIA_TYPE.AUDIO_CALL	音频。
CALL_MEDIA_TYPE.VIDEO_CALL	视频。

AUDIO_PLAYBACK_DEVICE

参数	说明
AUDIO_PLAYBACK_DEVICE.EAR	听筒。
AUDIO_PLAYBACK_DEVICE.SPEAKER	扬声器。

offlinePushInfo

参数	类型	是否必填	含义
offlinePushInfo.title	String	否	离线推送标题（选填）。
offlinePushInfo.description	String	否	离线推送内容（选填）。
offlinePushInfo.androidOPPOChannelID	String	否	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID（选填）。
offlinePushInfo.extension	String	否	离线推送透传内容（选填）（tsignaling 版本 $\geq$ 0.9.0）。

TUICallKit

最近更新时间：2025-05-16 17:17:42

TUICallKit 是音视频通话组件的含 UI 接口，使用 TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景；更详细的接入步骤，详见 [快速接入TUICallKit](#)。

❗ 说明：

- 为提供更优质的音视频通信能力，原 @tencentcloud/call-uikit-wechat 包已正式停止维护，建议开发者迁移至 @tencentcloud/call-uikit-wx-uniapp、@tencentcloud/call-uikit-wx 新包。
- 新包对接口进行了优化调整，部分旧版接口不再兼容，同步提供更简洁、稳定的新接口方案。

API 概览

API	描述
<TUICallKit />	TUICallKit 通话 UI 组件。
init	初始化 TUICallKit
calls	发起单人或多人通话。
join	主动加入通话。
setSelfInfo	设置用户的头像、昵称。
setCallingBell	设置自定义来电铃声。
setLogLevel	设置日志级别。
enableFloatWindow	开启/关闭悬浮窗功能。
enableMuteMode	开启/关闭来电铃声。
hideFeatureButton	隐藏按钮。
setLocalViewBackgr oundImage	设置本地用户通话界面背景图。
setRemoteViewBack groundImage	设置远端用户通话界面背景图。
setLayoutMode	设置通话界面布局模式。

<code>setCameraDefaultState</code>	设置摄像头是否默认打开。
<code>destroyed</code>	销毁 TUICallKit。
<code>getTUICallEngineInstance</code>	获取 TUICallEngine 实例。

API 详情

uni-app 小程序

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-wx-uniapp";
```

微信小程序

```
import { TUICallKitServer } from "@tencentcloud/call-uikit-wx";
```

init

初始化 TUICallKit。

```
TUICallKitServer.init({
  sdkAppID: 0, // 替换为您自己账号下的 SDKAppId
  userID: 'jane', // 填写当前用的 userID
  userSig: 'xxxxxxxxxxxxx',
  tim: null, // 如果您不需要 TIM 实例，可忽略
})
```

参数如下表所示：

参数	类型	说明	是否必填
sdkAppID	Number	您可以在 实时音视频控制台 > 应用管理 > 应用信息 中查看 SDKAppID。	是
userID	String	当前用户的 ID，字符串类型，只允许包含英文字母（a-z 和 A-Z）、数字（0-9）、连词符（-）和下划线（_）。	是
userSig	String	腾讯云设计的一种安全保护签名，获取方式请参见： 如何计算	是

		及使用 UserSig。	
tim	ChatSDK	TIM 实例。	否

这里详细介绍一下 init 中的几个参数：

- **userSig**：使用步骤三中获取的 SecretKey 对 sdkAppID、userID 等信息进行加密，就可以得到 userSig，它是一个鉴权用的票据，用于腾讯云识别当前用户是否能够使用 TRTC 的服务，更多信息请参见 [如何计算及使用 UserSig](#)。
- **tim** 可以将外部的 tim 实例通过 init 透传给 callkit，tim 参数适用于业务中已存在 TIM 实例，为保证 TIM 实例唯一性。

calls

发起群组通话。

```
TUICallKitServer.calls({
  userIDList: ["jane", "mike", "tommy"],
  type: 1,
})
```

参数如下表所示：

参数	类型	含义	是否必填
userIDList	Array<String>	目标用户的 userID 列表，示例：["jane", "mike", "tommy"]。	是
type	CallMediaType	通话的媒体类型，参数值说明参见 CallMediaType 通话类型 。	是
chatGroupID	String	与 chat 结合使用时，chat 群组的群ID。	是
roomID	Number	数字房间号，范围 [1, 2147483647]。	否
userData	String	扩展字段：用于在邀请信令中增加扩展信息。	否
timeout	Number	• 通话的超时时间，0为不超时，单位 s(秒)（选填）。设置范围：10s ~ 600s。如果传入0，默认30s。 • 被叫需要在30s内处于登录状态，timeout 值才会有效，具体参见 timeout 字段设置无效。	否

offlinePushInfo	Object	自定义离线消息推送（选填）。	否
-----------------	--------	----------------	---

join

发起群组通话。

```
TUICallKitServer.join({callId: xx});
```

参数如下表所示：

参数	类型	含义	是否必填
callId	string	此次通话的唯一 ID。	是

setSelfInfo

设置用户头像、昵称。


注意：

通话中使用该接口修改用户信息，UI 不会立即更新，需要等到下次通话才能看到变化。

```
TUICallKitServer.setSelfInfo({ nickName: "xxx", avatar: "http://xxx" });
```

参数如下表所示：

参数	类型	含义
nickName	String	设置昵称。
avatar	String	头像地址。

setCallingBell

设置自定义来电铃声。

这里仅限传入本地文件地址，需要确保该文件目录是应用可以访问的。

- 传入路径应为本地铃声文件的绝对地址。
- 如需恢复默认铃声，filePath 传空即可。
- 支持的铃声文件格式：

格式	iOS	Android
m4a	✓	✓
mp3	✓	✓
wav	✓	✓
aac	✓	✓

```
TUICallKitServer.setCallingBell("filePath")
```

参数如下表所示：

参数	类型	含义
filePath	String	铃声地址。

setLogLevel

设置日志级别，低于 level 的日志将不会输出。

```
TUICallKitServer.setLogLevel(level)
```

参数如下表所示：

参数	值	含义
level	0	普通级别，日志量较多，接入时建议使用。
	1	release 级别，SDK 输出关键信息，生产环境时建议使用。
	2	告警级别，SDK 只输出告警和错误级别的日志。
	3	错误级别，SDK 只输出错误级别的日志。
	4	无日志级别，SDK 将不打印任何日志。

enableFloatWindow

开启/关闭悬浮窗功能。

默认为 `false`，通话界面左上角的悬浮窗按钮隐藏，设置为 `true` 后显示。

```
TUICallKitServer.enableFloatWindow(enable?: Boolean)
```

enableMuteMode

开启/关闭来电铃声。

开启后，收到通话请求时，不会播放来电铃声。

```
try {
  await TUICallKitServer.enableMuteMode(enable: boolean)
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the enableMuteMode API.
Reason: ${error}`);
}
```

hideFeatureButton

隐藏功能按钮，目前仅支持 摄像头，麦克风和切换前后置按钮。

```
TUICallKitServer.hideFeatureButton(buttonName: FeatureButton);
```

参数列表：

参数	类型	是否必填	含义
buttonName	FeatureButton	是	按钮名称。

setLocalViewBackgroundImage

设置本地用户通话背景。

```
TUICallKitServer.setLocalViewBackgroundImage(url: string);
```

参数列表：

参数	类型	是否必填	含义
url	string	是	图片地址（支持本地路径和网络地址）。

**注意：**

如果使用网络地址，需要将网络地址添加到 request 合法域名。

setRemoteViewBackgroundImage

设置远端用户通话背景。

```
TUICallKitServer.setRemoteViewBackgroundImage(userId: string, url: string);
```

参数列表：

参数	类型	是否必填	含义
userId	string	是	远端用户 userId，设置为 '*' 表示对所有远端用户生效。
url	string	是	图片地址（支持本地路径和网络地址）。 注意： 如果使用网络地址，需要将网络地址添加到 request 合法域名。

setLayoutMode

设置通话界面布局模式。

```
import { TUICallKitServer, LayoutMode } from
"../../TUICallKit/src/index";
TUICallKitServer.setLayoutMode(LayoutMode.LocalInLargeView);
```

参数列表：

参数	类型	是否必填	含义
layoutMode	Layout Mode	是	用户流的布局模式。

setCameraDefaultState

设置摄像头是否默认打开。

```
TUICallKitServer.setCameraDefaultState(true);
```

参数列表：

参数	类型	是否必填	含义
isOpen	boolean	是	是否开启摄像头。

destroyed

销毁 TUICallKit。

```
TUICallKitServer.destroyed();
```

getTUICallEngineInstance

获取 TUICallEngine 实例。

```
TUICallKitServer.getTUICallEngineInstance();
```

TUICallKit 类型定义

offlinePushInfo

参数	类型	是否必填	含义
offlinePushInfo.title	String	否	离线推送标题（选填）。
offlinePushInfo.description	String	否	离线推送内容（选填）。
offlinePushInfo.androidOPPOChannelID	String	否	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID（选填）。
offlinePushInfo.extensi	Strin	否	离线推送透传内容（选填）（tsignaling 版本 ≥

on	g		0.9.0)。
----	---	--	---------

FeatureButton

FeatureButton 类型	描述
FeatureButton.Camera	摄像头按钮。
FeatureButton.Microphone	麦克风按钮。
FeatureButton.SwitchCamera	切换前后置摄像头按钮。
FeatureButton.InviteUser	邀请他人按钮。

LayoutMode

LayoutMode 类型	描述
LayoutMode.LocalInLargeView	本地用户在大窗显示。
LayoutMode.RemoteInLargeView	远端用户在大窗显示。

废弃接口

call

拨打电话（1v1 通话）。

```
TUICallKitServer.call({
  userID: "jane",
  type: 1
})
```

参数如下表所示：

参数	类型	含义	是否必填
userID	String	被叫用户的 ID。	是

type	Number	通话的媒体类型：1：语音通话 2：视频通话。	是
roomID	Number	数字房间号, 范围 [1, 2147483647]。	否
strRoomID	String	字符串房间号。 v3.3.1+ 支持 1. roomID 与 strRoomID 是互斥的，若您选用 strRoomID，则 roomID 需要填写为 0。若两者都填，SDK 将优先选用 roomID。 2. 不要混用 roomID 和 strRoomID，因为它们之间是不互通的，例如数字 123 和字符串 "123" 是两个完全不同的房间。	否
userData	String	扩展字段：用于在邀请信令中增加扩展信息。	否
timeout	Number	- 通话的超时时间，0为不超时，单位 s(秒)（选填），默认 30s。 - 被叫需要在30s内处于登录状态，timeout 值才会有效，具体参见 timeout 字段设置无效。	否
offlinePushInfo	Object	自定义离线消息推送（选填）。	否

groupCall

发起群组通话。

```
TUICallKitServer.groupCall({
  userIDList: ["jane", "mike", "tommy"],
  type: 1,
  groupID: "12345678"
})
```

参数如下表所示：

参数	类型	含义	是否必填
userIDList	Array<String>	目标用户的 userId 列表，示例：["jane", "mike", "tommy"]。	是
type	CallMediaType	通话的媒体类型，参数值说明参见 CallMediaType 通话类型 。	是

groupID	String	此次群组通话的群 ID。	是
roomID	Number	数字房间号, 范围 [1, 2147483647]。	否
strRoomID	String	字符串房间号。 v3.3.1+ 支持 1. roomID 与 strRoomID 是互斥的, 若您选用 strRoomID, 则 roomID 需要填写为 0。若两者都填, SDK 将优先选用 roomID。 2. 不要混用 roomID 和 strRoomID, 因为它们之间是不互通的, 例如数字 123 和字符串 "123" 是两个完全不同的房间。	否
userData	String	扩展字段: 用于在邀请信令中增加扩展信息。	否
timeout	Number	- 通话的超时时间, 0为不超时, 单位 s(秒) (选填), 默认30s。 - 被叫需要在30s内处于登录状态, timeout 值才会有效, 具体参见 timeout 字段设置无效。	否
offlinePushInfo	Object	自定义离线消息推送 (选填)。	否

joinInGroupCall

加入群组中已有的音视频通话。

❗ 说明:

- 加入群组中已有的音视频通话前, 需要提前创建或加入IM 群组, 并且群组中已有用户在通话中, 如果已经创建, 请忽略。
- 群组的创建详见: [IM 群组管理](#), 或者您也可以直接使用 [IM TUIKIT](#), 一站式集成聊天、通话等场景。

```
try {
  const params = {
    type: 2, // 视频通话
    groupID: "xxx",
    roomID: 0,
  };
  await TUICallKitServer.joinInGroupCall(params);
} catch (error: any) {
  console.error(`[TUICallKit] Failed to call the enableMuteMode API. Reason: ${error}`);
}
```

```
}

```

参数列表：

参数	类型	是否必填	含义
type	CallMediaType	是	通话的媒体类型，参数值说明参见 CallMediaType 通话类型 。
groupID	string	是	此次群组通话的群 ID。
roomID	number	是	此次通话的音视频房间 ID。

TUICallEngine

最近更新时间：2025-05-16 17:17:42

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

由于小程序开发的限制，需要先给 live-pusher 绑定事件，具体请参见 [TUICallkit](#)。

- ❗ 说明：
- 为提供更优质的音视频通信能力，原 tuicall-engine-wx 包已正式停止维护，建议开发者迁移至 @tencentcloud/tuicall-engine-wx 新包。
 - 新包对接口进行了优化调整，部分旧版接口不再兼容，同步提供更简洁、稳定的新接口方案。

API 概览

创建实例和事件回调

API	描述
createInstance	创建 TUICallEngine 实例（静态方法）
destroyInstance	销毁 TUICallEngine 实例（静态方法）
on	增加事件监听
off	取消事件监听

通话操作相关接口函数

API	描述
calls	发起 1v1 通话
inviteUser	在群组通话中，邀请其他人加入
join	主动加入当前的群组通话中
accept	接听通话
reject	拒绝通话
hangup	结束通话

设备控制相关接口函数

API	描述
openCamera	开启摄像头
closeCamera	关闭摄像头
switchCamera	切换前后摄像头
openMicrophone	打开麦克风
closeMicrophone	关闭麦克风
selectAudioPlaybackDevice	选择音频播放设备（听筒/扬声器）

其他接口函数

API	描述
setSelfInfo	设置用户的头像、昵称
setLogLevel	设置日志等级
setBeautyLevel	设置美颜级别，支持关闭默认美颜

API 详情

```
import TUICallEngine, { TUICallEvent, CALL_MEDIA_TYPE } from
 '@tencentcloud/tuicall-engine-wx';
```

createInstance

创建 TUICallEngine 的单例。

```
wx.$TUICallEngine = TUICallEngine.createInstance({
  tim,
  sdkAppID,
});
```

说明

参数	类型	说明	是否必传
sdkAppID	Number	SDKAppId 是腾讯云用于区分客户的唯一标识	是

tim	ChatSDK	TIM 实例	否
-----	---------	--------	---

destroyInstance

销毁 TUICallEngine 的单例。

```
TUICallEngine.destroyInstance();
```

on

通过 on 方法可以监听 TUICallEngine 所有的事件。

```
let handleNewInvitationReceived = function(event) {  
    // 收到邀请  
};  
wx.$TUICallEngine.on(TUICallEvent.INVITED, handleNewInvitationReceived,  
this);
```

说明

参数	类型	说明	是否必传
eventName	TUICall Event	事件名	是
handler	Function	事件的回调函数	是
context	Any	当前执行上下文（this）	是

off

移除事件监听。

```
let handleNewInvitationReceived = function(event) {  
    // 收到邀请  
};  
wx.$TUICallEngine.off(TUICallEvent.INVITED,  
handleNewInvitationReceived);
```

说明

参数	类型	说明	是否必传
eventName	TUICallEvent	事件名	是
handler	Function	事件的回调函数	是

login

登录接口。

```
const params = {
  userID: 'john', // 您的用户 ID
  userSig: 'xxxx', // 签名
};
let promise = wx.$TUICallEngine.login(params);
promise.then(() => {
  //success
}).catch(error => {
  console.warn('login error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
userID	String	当前用户的 ID，字符串类型，只允许包含英文字母（a-z 和 A-Z）、数字（0-9）、连词符（-）和下划线（_）	是
userSig	String	腾讯云设计的一种安全保护签名，获取方式请参考 如何计算 UserSig 。	是

logout

登出接口。

```
let promise = wx.$TUICallEngine.logout();
promise.then(() => {
  // logout success
}).catch(error => {
  console.warn('logout error:', error);
});
```

calls

发起单人或多人通话。

⚠ 注意

离线推送仅适用于终端（Android 或 iOS），Web 和 微信小程序不支持。

```
import { MEDIA_TYPE } from 'tuicall-engine-wx';
let promise = wx.$TUICallEngine.groupCall({
  userIDList: ['user1', 'user2'],
  type: CALL_MEDIA_TYPE.AUDIO_CALL,
});
promise.then(() => {
  // groupCall success
}).catch(error => {
  console.warn('groupCall error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
userIDList	Array<String>	被呼叫者的 userID 列表	是
type	MEDIA_TYPE	通话的媒体类型	是
chatGroupID	String	与 chat 结合使用时，chat 群组的群ID	是
timeout	Number	超时时长，单位：秒	否
roomID	Number	此次通话的音视频房间 ID，目前仅支持数字房间号，后续版本会支持字符串房间号	否
userData	String	扩展字段: 用于在邀请信令中增加扩展信息	否

inviteUser

邀请用户加入此次群组通话。

使用场景：一个群组通话中的用户主动邀请其他人时使用。

```
const userIDList = ['jack', 'john'];
```

```
const params = {
  userIDList
};
wx.$TUICallEngine.inviteUser(params).then(() => {
  // success
}).catch(error => {
  console.error('inviteUser error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
params	Object	邀请参数	是
params.userIDList	Array<String>	被叫用户的 userID 列表	是

join

主动加入此次群组通话。

```
const params = { callId: xxx };
wx.$TUICallEngine.join(params).then(() => {
  // success
}).catch(error => {
  console.error('join error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
params	Object	主动加入参数	是
params.callId	string	是	此次通话的唯一 ID。

accept

接受当前通话，当您作为被叫方收到 `EVENT.INVITED` 事件的回调时，可以调用该接口接听来电。

```
wx.$TUICallEngine.on(TUICallEvent.INVITED, () => {
  wx.$TUICallEngine.accept().promise.then(() => {
```

```
        // accept success
    }).catch(error => {
        console.warn('accept error:', error);
    });
});
```

reject

拒绝当前通话，当您作为被叫收到 `EVENT.INVITED` 的回调时，可以调用该函数拒绝来电。

```
wx.$TUICallEngine.on(TUICallEvent.INVITED, () => {
    wx.$TUICallEngine.reject().then(() => {
        // reject success
    }).catch(error => {
        console.warn('reject error:', error);
    });
});
```

hangup

挂断当前通话，当您处于通话中，可以调用该函数结束通话。

```
wx.$TUICallEngine.hangup().then(() => {
    // hangup success
}).catch(error => {
    console.warn('hangup error:', error);
});
```

openCamera

您可以调用该函数开启摄像头。

```
wx.$TUICallEngine.openCamera().then(() => {
    // openCamera success
}).catch(error => {
    console.warn('openCamera error:', error);
});
```

closeCamera

您可以调用该函数关闭摄像头，处于通话中的用户会收到回调。

```
wx.$TUICallEngine.closeCamera().then(() => {  
  // closeCamera success  
}).catch(error => {  
  console.warn('closeCamera error:', error);  
});
```

switchCamera

切换前后摄像头。

```
wx.$TUICallEngine.switchCamera().then(() => {  
  // switchCamera success  
}).catch(error => {  
  console.warn('openMicrophone error:', error);  
});
```

openMicrophone

您可以调用该函数打开麦克风，处于通话中的用户会收到回调。

```
wx.$TUICallEngine.openMicrophone().then(() => {  
  // openMicrophone success  
}).catch(error => {  
  console.warn('openMicrophone error:', error);  
});
```

closeMicrophone

您可以调用该函数关闭麦克风，处于通话中的用户会收到回调。

```
wx.$TUICallEngine.closeMicrophone().then(() => {  
  // closeMicrophone success  
}).catch(error => {  
  console.warn('closeMicrophone error:', error);  
});
```

selectAudioPlaybackDevice

选择音频播放设备，目前支持听筒、扬声器，在通话场景中，可以使用这个接口来开启/关闭免提模式。

```
wx.$TUICallEngine.selectAudioPlaybackDevice(AUDIO_PLAYBACK_DEVICE.EAR).then(() => {  
    // selectAudioPlaybackDevice success  
}).catch(error => {  
    console.warn('selectAudioPlaybackDevice error:', error);  
});
```

参数如下表所示：

参数	类型	含义	是否必传
type	AUDIO_PLAYBACK_DEVICE	扬声器： AUDIO_PLAYBACK_DEVICE.SPEAKER 听筒：AUDIO_PLAYBACK_DEVICE.EAR	是

setSelfInfo

设置用户头像、昵称。

```
wx.$TUICallEngine.setSelfInfo("昵称", "头像 URL").then(() => {  
    // setSelfInfo success  
}).catch(error => {  
    console.warn('setSelfInfo error:', error);  
});
```

参数如下表所示：

参数	类型	含义
nickName	String	设置昵称
avatar	String	头像地址

setLogLevel

设置日志级别，低于 level 的日志将不会输出。

```
wx.$TUICallEngine.setLogLevel(level)
```

参数如下表所示：

参数	值	含义
level	0	普通级别，日志量较多，接入时建议使用
	1	release 级别，SDK 输出关键信息，生产环境时建议使用
	2	告警级别，SDK 只输出告警和错误级别的日志
	3	错误级别，SDK 只输出错误级别的日志
	4	无日志级别，SDK 将不打印任何日志

setBeautyLevel

 **注意：**
≥ v2.1.0 支持

设置美颜级别，支持关闭默认美颜。

```
wx.$TUIEngine.setBeautyLevel (beautyLevel: number);
```

参数如下表所示：

参数	类型	含义
beautyLevel	number	美颜级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显

废弃接口

API	描述
call	拨打电话（1v1通话），建议使用 calls 接口。
groupCall	发起群组通话，建议使用 calls 接口。
startLocalView	启动本地画面渲染，建议使用 openCamera。
stopLocalView	停止本地画面渲染，建议使用 closeCamera。
joinInGroupCall	主动加入此次群组通话，建议使用 join 接口。
switchCallMediaType	切换通话媒体类型，比如视频通话切音频通话

call

拨打电话（1v1通话）。

⚠ 注意：

离线推送仅适用于终端（Android 或 iOS），Web 和 微信小程序不支持。

```
let promise = wx.$TUICallEngine.call({
  userID: 'user1',
  type: MEDIA_TYPE.AUDIO,
});
promise.then(() => {
  // call success
}).catch(error => {
  console.warn('call error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
userID	String	被呼叫者的 userID	是
type	MEDIA_TYPE	通话的媒体类型 MEDIA_TYPE.AUDIO：语音通话 MEDIA_TYPE.VIDEO：视频通话	是
roomId	Number	数字房间号, 范围 [1, 2147483647]	否
userData	String	扩展字段: 用于在邀请信令中增加扩展信息	否

groupCall

发起群组通话。

⚠ 注意：

离线推送仅适用于终端（Android 或 iOS），Web 和 微信小程序不支持。

```
import { MEDIA_TYPE } from 'tuicall-engine-wx';
let promise = wx.$TUICallEngine.groupCall({
  userIDList: ['user1', 'user2'],
  type: MEDIA_TYPE.AUDIO,
  groupID: '123345',
});
```

```
promise.then(() => {
  // groupCall success
}).catch(error => {
  console.warn('groupCall error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
userIDList	Array<String>	被呼叫者的 userID 列表	是
type	MEDIA_TYPE	通话的媒体类型 MEDIA_TYPE.AUDIO: 语音通话 MEDIA_TYPE.VIDEO: 视频通话	是
groupID	String	此次群组通话的群 ID	是
timeout	Number	超时时长，单位：秒	否
roomID	Number	此次通话的音视频房间 ID，目前仅支持数字房间号，后续版本会支持字符串房间号	否
userData	String	扩展字段: 用于在邀请信令中增加扩展信息	否

joinInGroupCall

主动加入此次群组通话。

使用场景：群组内用户主动加入此次群组通话使用。

```
import { MEDIA_TYPE } from 'tuicall-engine-wx';
const params = {
  roomID: 123,
  type: MEDIA_TYPE.AUDIO,
  groupID: 111
};
wx.$TUICallEngine.joinInGroupCall(params).then(() => {
  // success
}).catch(error => {
  console.error('joinInGroupCall error:', error);
});
```

```
});
```

参数如下表所示：

参数	类型	含义	是否必传
params	Object	主动加入参数	是
roomId	Number	此次通话的音视频房间 ID，目前仅支持数字房间号，后续版本会支持字符串房间号	是
groupId	String	此次群组通话的群 ID	是
type	MEDIA_TYPE	通话的媒体类型 MEDIA_TYPE.AUDIO：语音通话 MEDIA_TYPE.VIDEO：视频通话	是

switchCallMediaType

当前通话类型切换。

仅支持 1v1 通话过程中使用

```
wx.$TUICallEngine.switchCallMediaType(MEDIA_TYPE.VIDEO).then(() => {
  // switchCallMediaType success
}).catch(error => {
  console.warn('switchCallMediaType error:', error);
});
```

参数如下表所示：

参数	类型	含义	是否必传
type	MEDIA_TYPE	通话的媒体类型 MEDIA_TYPE.AUDIO：语音通话 MEDIA_TYPE.VIDEO：视频通话	是

TUICallEvent

最近更新时间：2025-05-16 17:17:42

TUICallEvent API 是音视频通话组件的事件接口。

- ❗ 说明：
- 为提供更优质的音视频通信能力，原 `tuicall-engine-webrtc` 包已正式停止维护，建议开发者迁移至 `@tencentcloud/tuicall-engine-js` 新包。
 - 新包对接口进行了优化调整，部分旧版接口不再兼容，同步提供更简洁、稳定的新接口方案。

TUICallEvent 概览

事件名	说明
ON_CALL_RECEIVED	通话请求的事件
USER_ACCEPT	用户接听
USER_ENTER	用户进入通话
USER_LEAVE	用户离开通话
USER_UPDATE	用户更新
REJECT	用户拒绝通话
NO_RESP	用户无响应
LINE_BUSY	用户忙线
ON_CALL_CANCELED	通话未建立时，通话的各端都会抛出该事件
ON_CALL_BEGIN	通话接通时抛出的事件
CALL_END	通话结束
SDK_READY	SDK Ready 回调
KICKED_OUT	被踢下线
CALL_MODE	切换通话模式
ERROR	错误信息

ON_USER_NETWORK_QUALITY_CHANGED

所有用户网络质量的事件

TUICallEvent 详情

```
import TUICallEngine, { TUICallEvent } from '@tencentcloud/tuicall-engine-wx';
```

ON_CALL_RECEIVED

收到一个新的来电请求事件，被叫会收到，您可以通过监听这个事件，来决定是否显示通话接听界面。

```
let handleOnCallReceived = function(event) {  
  console.log(event)  
};  
tuiCallEngine.on(TUICallEvent.ON_CALL_RECEIVED, handleOnCallReceived,  
this);
```

参数如下表所示：

参数	类型	含义
sponsor	String	邀请者
userIDList	Array<String>	同时还被邀请的人
isFromGroup	Boolean	是否是群组通话
inviteData	Object	呼叫数据
inviteID	String	邀请 ID，标识一次邀请
userData	String	扩展字段：用于在邀请信令中增加扩展信息
callId	String	本次通话的唯一 ID。
roomId	Number	此次通话的音视频房间 ID。
callMediaType	Number	通话的媒体类型，视频通话、语音通话。
callRole	String	角色，枚举类型：主叫、被叫。

USER_ACCEPT

如果有用户接听，其他用户会收到该事件。

```
let handleUserAccept = function(event) {  
    console.log('用户接听')  
}  
tuiCallEngine.on(TUICallEvent.USER_ACCEPT, this.handleUserAccept, this);
```

参数如下表所示：

参数	类型	含义
userID	String	接听的用户 ID
userList	Array	未接听用户的信息（后续将废弃）

USER_ENTER

如果有用户同意进入通话，其他用户会收到该事件。

```
let handleUserEnter = function(event) {  
    console.log('用户进入通话')  
}  
tuiCallEngine.on(TUICallEvent.USER_ENTER, this.handleUserEnter, this);
```

参数如下表所示：

参数	类型	含义
userID	String	进房的用户 ID
playerList	Array	房间中用户的推流信息

USER_LEAVE

有用户离开此次通话时，通话中的其他用户会收到该事件。

```
let handleUserLeave = function(event) {  
    console.log('用户离开通话')  
}  
tuiCallEngine.on(TUICallEvent.USER_LEAVE, this.handleUserLeave, this);
```

参数如下表所示：

参数	类型	含义
userID	String	退房的用户 ID
playerList	Array	房间中用户的推流信息

USER_UPDATE

用户更新。

```
let handleUserUpdate = function(event) {  
  console.log('用户更新')  
}  
tuiCallEngine.on(TUICallEvent.USER_UPDATE, this.handleUserUpdate, this)
```

参数如下表所示：

参数	类型	含义
pusher	String	更新后的自身推流信息
playerList	Array	更新后其他用户的推流信息

REJECT

通话被拒绝事件，在 1v1 通话中，只有主叫方会收到拒绝事件，在群组通话中，所有被邀请者都可以收到该事件。

```
let handleInviteeReject = function(event) {  
  console.log('用户拒绝通话')  
}  
tuiCallEngine.on(TUICallEvent.REJECT, this.handleInviteeReject, this);
```

参数如下表所示：

参数	类型	含义
userID	String	拒绝通话的用户 ID
invitee	String	拒绝通话的用户 ID（后续将废弃）
inviteID	String	邀请 ID，标识一次邀请

reason	string	reject 表示拒绝
--------	--------	-------------

NO_RESP

邀请用户无应答。

- 在C2C通话中，只有发起方会收到无人应答的回调 例如 A 邀请 B、C 进入通话，B不应答，A可以收到该回调，但C不行。
- 在IM群组通话中，所有被邀请人均能收到该回调 例如 A 邀请 B、C 进入通话，B不应答，A、C均能收到该回调。

```
let handleNoResponse = function(event) {  
    console.log('用户无响应')  
}  
tuiCallEngine.on(TUICallEvent.NO_RESP, this.handleNoResponse, this);
```

参数如下表所示：

参数	类型	含义
groupId	String	群ID, 唯一标识
sponsor	String	发起方的用户 ID
userIDList	Array<String>	触发超时无响应的用户列表
timeoutUserList	Array<String>	触发超时无响应的用户列表（后续将废弃）
inviteID	String	邀请 ID, 标识一次邀请

LINE_BUSY

用户忙线。

```
let handleLineBusy = function(event) {  
    console.log('用户忙线')  
}  
tuiCallEngine.on(TUICallEvent.LINE_BUSY, this.handleLineBusy, this);
```

参数如下表所示：

参数	类型	含义
userID	String	忙线用户 ID

invitee	String	忙线用户 ID（后续将废弃）
inviteID	String	邀请 ID, 标识一次邀请
reason	string	line busy 表示忙线

ON_CALL_CANCELED

如果通话未建立，都会抛出该事件。您可以通过监听该事件来实现类似未接来电、重置 UI 状态等显示逻辑。通话未建立的场景如下：

- 主叫取消：主叫抛出该事件，userID 为主叫；被叫也会抛出该事件，userID 为被叫；
- 被叫超时：主叫会同时抛出 **NO_RESP** 和 **CALLING_CANCEL** 事件，userID 为主叫；被叫抛出 **CALLING_CANCEL** 事件，userID 为被叫；
- 被叫拒接：主叫会同时抛出 **REJECT** 和 **CALLING_CANCEL** 事件，userID 为主叫；被叫抛出 **CALLING_CANCEL** 事件，userID 为被叫；
- 被叫忙线：主叫会同时抛出 **LINE_BUSY** 和 **CALLING_CANCEL** 事件，userID 为主叫；被叫抛出 **CALLING_CANCEL** 事件，userID 为被叫；

```
let handleOnCallCanceled = function(event) {
  console.log(event.userID);
};
tuiCallEngine.on(TUICallEvent.ON_CALL_CANCELED, handleOnCallCanceled,
this);
```

参数如下表所示：

参数	类型	含义
userID	String	通话取消的用户 ID
callId	String	本次通话的唯一 ID
roomId	Number	此次通话的音视频房间 ID
callMediaType	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，枚举类型：主叫、被叫

ON_CALL_BEGIN

表示通话接通，主叫和被叫都可以收到，您可以通过监听这个事件来开启云端录制、内容审核等流程。

```
let handleOnCallBegin = function(event) {
    console.log(event)
};
tuiCallEngine.on(TUICallEvent.ON_CALL_BEGIN, handleOnCallBegin, this);
```

参数如下表所示：

参数	类型	含义
callId	String	本次通话的唯一 ID
roomId	Number	此次通话的音视频房间 ID
callMedia aType	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，类型：主叫、被叫

CALL_END

表示通话挂断，主叫和被叫都可以收到，您可以通过监听这个事件来显示通话时长、通话类型等信息，或者来停止云端的录制流程。

```
let handleCallingEnd = function(event) {
    console.log('通话结束')
}
tuiCallEngine.on(TUICallEvent.CALL_END, this.handleCallingEnd, this);
```

参数如下表所示：

参数	类型	含义
roomId	Number	此次通话的音视频房间 ID，目前仅支持数字房间号，后续版本会支持字符串房间号
callMedia Type	Number	通话的媒体类型，视频通话、语音通话
callRole	String	角色，枚举类型：主叫 ('inviter')、被叫 ('invitee')、未知 ('')
totalTime	Number	此次通话的时长，单位: 秒
userID	String	表示引起通话结束 userID

callId	String	本次通话的唯一 ID。
--------	--------	-------------

SDK_READY

SDK Ready 回调。

```
let handleSDKReady = function(event) {  
    console.log('SDK Ready 回调')  
}  
tuiCallEngine.on(TUICallEvent.SDK_READY, this.handleSDKReady, this);
```

KICKED_OUT

被踢下线。

```
let handleKickedOut = function(event) {  
    console.log('被踢下线')  
}  
tuiCallEngine.on(TUICallEvent.KICKED_OUT, this.handleKickedOut, this);
```

CALL_MODE

切换通话模式。

```
let handleCallMode = function(event) {  
    console.log('切换通话模式')  
}  
tuiCallEngine.on(TUICallEvent.CALL_MODE, this.handleCallMode, this);
```

ERROR

监听 SDK 的错误信息。

```
let handleError = function(event) {  
    console.log('错误信息')  
}  
tuiCallEngine.on(TUICallEvent.ERROR, this.handleError, this);
```

ON_USER_NETWORK_QUALITY_CHANGED

所有用户网络质量的事件。

```
let handleOnUserNetworkQualityChange = function(event) {
  console.log(event.networkQualityList);
};
tuiCallEngine.on(TUICallEvent.ON_USER_NETWORK_QUALITY_CHANGED,
this.handleOnUserNetworkQualityChange, this);
```

参数如下表所示：

参数	类型	含义
networkQualityList	Array<Object>	网络状态，根据 userID 可以获取对应用户当前的网络质量。例如：networkQualityList: [{ userId: quality }]。 网络质量说明： - quality = 0，网络状况未知 - quality = 1，网络状况极佳 - quality = 2，网络状况较好 - quality = 3，网络状况一般 - quality = 4，网络状况差 - quality = 5，网络状况极差 - quality = 6，网络链接已断开

废弃事件

INVITED

收到一个新的来电请求事件，被叫会收到，您可以通过监听这个事件，来决定是否显示通话接听界面。

```
let handleNewInvitationReceived = function(event){
  console.log('被邀请通话')
}
tuiCallEngine.on(EVENT.INVITED, this.handleNewInvitationReceived, this);
```

参数如下表所示：

参数	类型	含义
sponsor	String	邀请者
inviteeList	Array<String>	同时还被邀请的人（后续将废弃）

userIDList	Array<String>	同时还被邀请的人
isFromGroup	Boolean	是否是群组通话
inviteData	Object	邀请数据
inviteID	String	邀请 ID，标识一次邀请
userData	String	扩展字段：用于在邀请信令中增加扩展信息
callId	String	本次通话的唯一 ID。 v1.5.1+ 版本支持
roomId	Number	此次通话的音视频房间 ID。 v1.5.1+ 版本支持
callMediaType	Number	通话的媒体类型，视频通话、语音通话。 v1.5.1+ 版本支持
callRole	String	角色，枚举类型：主叫、被叫。 v1.5.1+ 版本支持

CALLING_CANCEL

表示此次通话被主叫取消（取消原因有可能是主叫主动取消、也有可能是来自于通话超时取消），被叫会收到，您可以通过监听这个事件来实现类似未接来电等显示逻辑。

1. A 呼叫 B，A 挂断，A 会抛出该事件，userID 就是 A。
2. A 呼叫 B，B 一直接通超时，B 会抛出该事件，userID 是 B。
3. A 呼叫 B、C、D，B 接通后，A 和 B 都挂断，此时 C 拒绝，此时 B 也会抛出该事件，事件里的 userID 为 C。

```
let handleCallingCancel = function(event) {
  console.log('通话被取消')
}
tuiCallEngine.on(EVENT.CALLING_CANCEL, this.handleCallingCancel, this);
```

参数如下表所示：

参数	类型	含义
userID	String	通话取消的用户 ID
invitee	String	通话取消的用户 ID（ 后续将废弃 ）
inviteID	String	邀请 ID，标识一次邀请

callId	String	本次通话的唯一 ID。 v1.5.1+ 版本支持
roomId	Number	此次通话的音视频房间 ID。 v1.5.1+ 版本支持
callMediaType	Number	通话的媒体类型，视频通话、语音通话。 v1.5.1+ 版本支持
callRole	String	角色，枚举类型：主叫、被叫。 v1.5.1+ 版本支持

uni-app 客户端

API 概览

最近更新时间：2025-05-14 10:09:22

TUICallKit (含 UI 接口)

TUICallKit API 是音视频通话组件的含 UI 接口，使用TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景。

API	描述
login	登录
logout	登出
setSelfInfo	设置用户的头像、昵称
calls	发起 1v1 通话
join	发起群组通话
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
enableIncomingBanner	开启/关闭来电横幅显示
setScreenOrientation	设置通话界面显示方向

TUICallEvent

TUICallEvent 是 TUICallEngine 对应的事件类，您可以通过监听如下事件，来监听自己感兴趣的事件。

API	描述
onError	通话过程中错误回调
onCallReceived	通话请求的回调
onCallCancelled	通话取消的回调
onCallBegin	通话接通的回调

onCallEnd	通话结束的回调
onCallMediaTypeChanged	通话媒体类型发生改变的回调
onUserReject	xxxx 用户拒绝通话的回调
onUserNoResponse	xxxx 用户不响应的回调
onUserLineBusy	xxxx 用户忙线的回调
onUserJoin	xxxx 用户加入通话的回调
onUserLeave	xxxx 用户离开通话的回调
onUserVideoAvailable	xxx 用户是否有视频流的回调
onUserAudioAvailable	xxx 用户是否有音频流的回调
onUserVoiceVolumeChanged	所有用户音量大小的反馈回调
onUserNetworkQualityChanged	所有用户网络质量的反馈回调

TUICallEngine (无 UI 接口)

TUICallEngine API 是音视频通话组件的**无 UI 接口**，使用 TUICallEngine API，您可以做一些额外的操作（目前仅支持以下接口）。

API	描述
hangup	结束通话
accept	接听通话
setVideoRenderParams	设置用户视频画面的渲染模式
setVideoEncoderParams	设置视频编码的编码参数

TUICallKit

最近更新时间：2025-05-14 10:09:22

TUICallKit API 是音视频通话组件的含 UI 接口，使用TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景，更详细的接入步骤，详情请参见 [快速接入 TUICallKit](#)。

API 概览

API	描述
login	登录
logout	登出
setSelfInfo	设置用户的昵称、头像
calls	发起单人或多人通话
join	主动加入通话
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
enableIncomingBanner	开启/关闭来电横幅显示
setScreenOrientation	设置通话界面显示方向

API 详情

login

登录

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  SDKAppID: 0,
  userID: 'mike',
  userSig: '',
};
TUICallKit.login(options, (res) => {
  if (res.code === 0) {
```

```
console.log('login success');
} else {
  console.log(`login failed, error message = ${res.msg}`);
}
});
```

参数	类型	含义
options	Object	初始化参数
options.SDKAppID	Number	用户 SDKAppID
options.userID	String	用户 ID
options.userSig	String	用户签名 userSig
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

logout

登出

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
TUICallKit.logout((res) => {
  if (res.code === 0) {
    console.log('logout success');
  } else {
    console.log(`logout failed, error message = ${res.msg}`);
  }
});
```

参数	类型	含义
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

setSelfInfo

设置用户昵称、头像。用户昵称不能超过500字节，用户头像必须是 URL 格式。

⚠ 注意:

通话中使用该接口修改用户信息，UI 不会立即更新，需要等到下次通话才能看到变化。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  nickName: 'jack',
  avatar: 'https://****/user_avatar.png'
}
TUICallKit.setSelfInfo(options, (res) => {
  if (res.code === 0) {
    console.log('setSelfInfo success');
  } else {
    console.log(`setSelfInfo failed, error message = ${res.msg}`);
  }
});
```

参数	类型	含义
options	Object	初始化参数
options.nickName	String	目标用户的昵称，非必填
options.avatar	String	目标用户的头像，非必填
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

calls

拨打电话（1v1通话）。

💡 说明:

v2.9.0 支持。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  userIDList: ['user1'],
```

```
callMediaType: 1, // 语音通话 (callMediaType = 1)、视频通话 (callMediaType = 2)
};
TUICallKit.calls(options, (res) => {
  if (res.code === 0) {
    console.log('call success');
  } else {
    console.log(`call failed, error message = ${res.msg}`);
  }
});
```

参数如下表所示：

参数	类型	含义
options	Object	初始化参数
options.userIDList	String[]	被呼叫的用户列表
options.callMediaType	Number	通话的媒体类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)
options.callParams	Object	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等。
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

join

主动加入通话。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  callId: xxx
};
TUICallKit.join(options, (res) => {
  if (res.code === 0) {
    console.log('joinInGroupCall success');
  } else {
    console.log(`joinInGroupCall failed, error message = ${res.msg}`);
  }
});
```

```
});
```

参数	类型	含义
options	Object	初始化参数
options.callId	String	此次通话的唯一 ID。
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

setCallingBell

设置自定义来电铃声，这里仅限传入本地文件地址，需要确保该文件目录是应用可以访问的。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');

// 【1】通过 uni.saveFile 保存音频文件到本地，具体参考 saveFile 接口：
https://zh.uniapp.dcloud.io/api/file/file.html#savefile
const tempFilePath = './static/rain.mp3'; // 本地存放的音频文件
let musicFilePath = '';
uni.saveFile({
  tempFilePath: tempFilePath,
  success: (res) => {
    console.warn('保存文件成功 = ', JSON.stringify(res)); // 获取的是相对路径
    musicFilePath = res.savedFilePath;

    // 【2】相对路径转绝对路径，否则访问不到
    musicFilePath =
      plus.io.convertLocalFileSystemURL(musicFilePath); // 转绝对路径

    // 【3】设置铃声
    TUICallKit.setCallingBell(musicFilePath, (res) => {
      if (res.code === 0) {
        console.log('setCallingBell success');
      } else {
        console.log(`setCallingBell failed, error message = ${res.msg}`);
      }
    });
  },
},
```

```
fail: (err) => {  
    console.error('保存文件失败');  
},  
});
```

参数	类型	含义
filePath	String	来电铃音本地文件地址
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

enableMuteMode

开启/关闭静音模式。

开启后，收到通话请求，不会播放来电铃声。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');  
const enable = true;  
TUICallKit.enableMuteMode(enable);
```

参数	类型	含义
enable	Boolean	开启、关闭静音；true 表示开启静音

enableFloatWindow

开启/关闭悬浮窗功能，设置为 false 后，通话界面左上角的悬浮窗按钮会隐藏。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');  
const enable = true;  
TUICallKit.enableFloatWindow(enable);
```

参数	类型	含义
enable	Boolean	开启、关闭悬浮窗功能；true 表示开启浮窗

enableIncomingBanner

开启/关闭来电横幅显示。

默认为 `false`，被叫端收到邀请后默认弹出全屏通话等待界面，开启后先展示一个横幅，然后根据需要拉起全屏通话界面。

注意：

v2.3.1+ 支持

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const enable = true;
TUICallKit.enableIncomingBanner(enable);
```

setScreenOrientation

设置通话界面显示方向。

默认为竖屏，orientation：0-Portrait、1-LandScape、2-Auto。

注意：

v2.6.0+ 支持，仅支持 Android 设备。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
TUICallKit.setScreenOrientation(1);
```

废弃接口

TUICallKit 类型定义

CallParams

通话参数

注意：

- roomID 与 strRoomID 是互斥的，若您选用 strRoomID，则 roomID 需要填写为 0。若两者都填，SDK 将优先选用 roomID。
- 不要混用 roomID 和 strRoomID，因为它们之间是不互通的，比如数字 123 和字符串 123 是两个完全不同的房间。

参数	类型	描述
roomID	Number	数字房间号

		取值范围 1 – 2147483647(2^31-1)
strRoomID	String	字符串房间号 推荐取值 限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： - 大小写英文字母（a-zA-Z） - 数字（0-9） - 空格、！、#、\$、%、&、（、）、+、-、:、;、<、=、.、>、?、@、[、]、^、_、{、}、 、~、,
chatGroupID	String	与 chat 结合使用时，chat 群组的群ID
offlinePushInfo	OfflinePushInfo	厂商离线推送配置信息
timeout	Number	通话超时时间，默认：30s，单位：秒。设置范围：10s ~ 600s。如果传入0，默认30s。
userData	String	发起通话时自定义扩展字段，被叫端在 onCallReceived 中回调

OfflinePushInfo

 **注意：**

- title、description 字段值不能传入空字符串，必须传入有效值。
- v2.3.2+ 支持。

参数	类型	描述
title	String	离线推送展示通知栏标题
description	String	离线推送展示通知栏内容
iOSSound	String	离线推送声音设置（仅对 iOS 生效）。当 sound = IOS_OFFLINE_PUSH_NO_SOUND，表示接收时不会播放声音。当 sound = IOS_OFFLINE_PUSH_DEFAULT_SOUND，表示接收时播放系统声音。如果要自定义 iOSSound，需要先把语音文件链接进 Xcode 工程，然后把语音文件名（带后缀）设置给 iOSSound
androidSound	String	离线推送声音设置（仅对 Android 生效, IMSDK 6.1 及以上版本支持）。只有华为和谷歌手机支持设置声音提示，小米手机设置声音提示，请您参照：小米自定义铃声。另外，谷歌手机 FCM 推送在 Android 8.0 及以

		上系统设置声音提示，必须调用 <code>setAndroidFCMChannelID</code> 设置好 <code>channelID</code> ，才能生效
<code>androidOPPOChannelID</code>	String	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID
<code>androidXiaomiChannelID</code>	String	小米手机 8.0 系统及以上的渠道 ID
<code>androidFCMChannelID</code>	String	FCM 通道手机 8.0 系统及以上的渠道 ID
<code>ignoreIOSBadge</code>	boolean	离线推送忽略 badge 计数（仅对 iOS 生效），如果设置为 <code>true</code> ，在 iOS 接收端，这条消息不会使 APP 的应用图标未读计数增加。
<code>isDisablePush</code>	boolean	是否关闭推送（默认开启推送）

废弃接口

call

拨打电话（1v1通话）

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  userID: 'mike',
  callMediaType: 1, // 语音通话 (callMediaType = 1)、视频通话 (callMediaType = 2)
};
TUICallKit.call(options, (res) => {
  if (res.code === 0) {
    console.log('call success');
  } else {
    console.log(`call failed, error message = ${res.msg}`);
  }
});
```

参数如下表所示：

参数	类型	含义
<code>options</code>	Object	初始化参数

options.userId	String	目标用户的 userId
options.callMediaType	Number	通话的媒体类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)
options.callParams	Object	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等。
callback	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

groupCall

发起群组通话，注意：使用群组通话前需要创建 IM 群组，如果已经创建，请忽略。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  groupId: 'myGroup',
  userIdList: ['mike', 'tom'],
  callMediaType: 1, // 语音通话 (callMediaType = 1)、视频通话 (callMediaType = 2)
};
TUICallKit.groupCall(options, (res) => {
  if (res.code === 0) {
    console.log('call success');
  } else {
    console.log(`call failed, error message = ${res.msg}`);
  }
});
```

参数	类型	含义
options	Object	初始化参数
options.groupID	String	此次群组通话的群 ID
options.userIdList	Array<string>	目标用户的 userId 列表
options.callMediaType	Number	通话的媒体类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)

<code>options.callParams</code>	Object	通话扩展参数，例如：房间号、通话邀请超时时间，离线推送自定义内容等。
<code>callback</code>	Function	回调函数，code = 0 表示调用成功；code != 0 表示调用失败，失败原因见 msg

joinInGroupCall

加入群组中已有的音视频通话。

```
const TUICallKit = uni.requireNativePlugin('TencentCloud-TUICallKit');
const options = {
  roomId: 9898,
  groupId: 'myGroup',
  callMediaType: 1, // 语音通话 (callMediaType = 1)、视频通话 (callMediaType = 2)
};
TUICallKit.joinInGroupCall(options, (res) => {
  if (res.code === 0) {
    console.log('joinInGroupCall success');
  } else {
    console.log(`joinInGroupCall failed, error message = ${res.msg}`);
  }
});
```

⚠ 注意：

- roomId 与 strRoomID 是互斥的，若您选用 strRoomID，则 roomId 需要填写为 0。若两者都填，SDK 将优先选用 roomId。
- 不要混用 roomId 和 strRoomID，因为它们之间是不互通的，比如数字 123 和字符串 123 是两个完全不同的房间。

参数	类型	含义
options	Object	初始化参数
options.roomID	Number	数字房间号 取值范围 1 – 2147483647(2 ³¹ -1)
options.strRoomID	String	字符串房间号 推荐取值 限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）：

		• 大小写英文字母（a-zA-Z） • 数字（0-9） • 空格、<code>!</code>、<code>#</code>、<code>\$</code>、<code>%</code>、<code>&</code>、<code>(</code>、<code>)</code>、<code>+</code>、<code>-</code>、<code>:</code>、<code>;</code>、<code><</code>、<code>=</code>、<code>.</code>、<code>></code>、<code>?</code>、<code>@</code>、<code>[</code>、<code>]</code>、<code>^</code>、<code>_</code>、<code>{</code>、<code>}</code>、<code> </code>、<code>~</code>、<code>,</code>
<code>options.groupID</code>	String	此次群组通话的群 ID
<code>options.callMediaType</code>	Number	通话的媒体类型，比如：语音通话(<code>callMediaType = 1</code>)、视频通话(<code>callMediaType = 2</code>)
<code>callback</code>	Function	回调函数， <code>code = 0</code> 表示调用成功； <code>code != 0</code> 表示调用失败，失败原因见 <code>msg</code>

TUICallEngine

最近更新时间：2025-05-14 10:09:22

TUICallEngine API 简介

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套接口实现交互。目前仅提供如下接口，如有需求可到这里 [反馈](#)。

⚠ 注意：

需要结合 [腾讯云音视频通话插件 TencentCloud-TUICallKit](#) 插件使用，不能单独使用。下述接口也是结合客户有自动挂断、设置渲染模式的需求下，才支持的。

API 概览

API	描述
hangup	结束通话
accept	接听通话
setVideoRenderParams	设置用户视频画面的渲染模式
setVideoEncoderParams	设置视频编码的编码参数

API 详情

hangup

挂断当前通话，当您处于通话中，可以调用该函数结束通话。

```
const TUICallEngine = uni.requireNativePlugin('TencentCloud-TUICallKit-TUICallEngine');
TUICallEngine.hangup();
```

accept

接受当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数接听来电。

```
const TUICallEngine = uni.requireNativePlugin('TencentCloud-TUICallKit-TUICallEngine');
```

```
TUICallEngine.accept();
```

setVideoRenderParams

设置用户视频画面的渲染模式。

```
const TUICallEngine = uni.requireNativePlugin('TencentCloud-TUICallKit-TUICallEngine');
const params = {
  userID: '234',
  fillMode: 0, // 0-填充模式, 1-适应模式
  rotation: 1, // 0~3分别对应值: Rotation_0, Rotation_90, Rotation_180, Rotation_270;
};
TUICallEngine.setVideoRenderParams(params, (res) => {
  console.warn('res = ', JSON.stringify(res));
});
```

参数如下表所示：

参数	类型	含义
userID	String	目标用户的 userID
params	Object	视频画面渲染参数，例如：画面旋转角度、填充模式

setVideoEncoderParams

设置视频编码的编码参数。

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

```
const TUICallEngine = uni.requireNativePlugin('TencentCloud-TUICallKit-TUICallEngine');
const params = {
  resolution: 108,
  resolutionMode: 0, // 0--横屏, 1--竖屏
};
TUICallEngine.setVideoEncoderParams(params, (res) => {
  console.warn('res = ', JSON.stringify(res));
});
```

参数如下表所示：

参数	类型	含义
resolution	Number	视频分辨率 62 -- 宽高比 16:9；分辨率 640x360； 64 -- 宽高比 4:3；分辨率 960x720； 108 -- 宽高比 16:9；分辨率 640x360； 110 -- 宽高比 16:9；分辨率 960x540； 112 -- 宽高比 16:9；分辨率 1280x720； 114 -- 宽高比 16:9；分辨率 1920x1080；
resolutionMode	Number	分辨率模式（横屏分辨率 or 竖屏分辨率） 0-Landscape（横屏） 1-Portrait（竖屏）

TUICallEvent

最近更新时间：2025-05-14 10:09:22

事件简介

可以监听 TUICallKit 对应的事件，进行提示或其它交互。

回调事件概览

API	描述
onError	通话过程中错误回调
onCallReceived	通话请求的回调
onCallCancelled	通话取消的回调
onCallBegin	通话接通的回调
onCallEnd	通话结束的回调
onCallMediaTypeChanged	通话媒体类型发生改变的回调
onUserReject	xxxx 用户拒绝通话的回调
onUserNoResponse	xxxx 用户不响应的回调
onUserLineBusy	xxxx 用户忙线的回调
onUserJoin	xxxx 用户加入通话的回调
onUserLeave	xxxx 用户离开通话的回调
onUserVideoAvailable	xxx 用户是否有视频流的回调
onUserAudioAvailable	xxx 用户是否有音频流的回调
onUserVoiceVolumeChanged	所有用户音量大小的反馈回调
onUserNetworkQualityChanged	所有用户网络质量的反馈回调
onKickedOffline	当前用户被踢下线
onUserSigExpired	在线时票据过期
CustomViewClickEvent	点击新增自定义按钮时会触发该事件。v2.7.0+

回调事件详情

通过 globalEvent 监听原生插件抛出的事件。

```
const TUICallKitEvent = uni.requireNativePlugin('globalEvent');
```

onError

错误回调。

❗ 说明

SDK 不可恢复的错误，一定要监听，并分情况给用户适当的界面提示。

```
TUICallKitEvent.addEventListener('onError', (res) => {  
  console.log('onError', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	错误回调参数
res.code	Number	错误码
res.msg	String	错误信息

onCallReceived

收到一个新的来电请求回调，被叫会收到，您可以通过监听这个事件，来决定是否显示通话接听界面。

```
TUICallKitEvent.addEventListener('onCallReceived', (res) => {  
  console.log('onCallReceived', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.callerId	String	主叫 ID（邀请方）

res.calleeIdList	Array<String>	被叫 ID 列表（被邀请方）
res.groupId	String	群通话 ID。在群组通话时返回，1v1 返回 groupId 为空字符串
res.callMediaType	Number	通话的媒体类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)
res.userData	String	用户增加的扩展字段

onCallCancelled

表示此次通话主叫取消、被叫超时、拒接等，涉及多个场景，您可以通过监听这个事件来实现类似未接来电、重置 UI 状态等显示逻辑。

- 主叫取消：主叫收到该回调（userId 为自己）；被叫收到该回调（userId 为主叫的 ID）。
- 被叫超时：主叫会同时收到 [onUserNoResponse](#) 和 onCallCancelled 回调（userId 是自己的 ID）；被叫收到 onCallCancelled 回调（userId 是自己的 ID）。
- 被叫拒接：主叫会同时收到 [onUserReject](#) 和 onCallCancelled 回调（userId 是自己的 ID）；被叫收到 onCallCancelled 回调（userId 是自己的 ID）。
- 被叫忙线：主叫会同时收到 [onUserLineBusy](#) 和 onCallCancelled 回调（userId 是自己的 ID）。
- 异常中断：被叫接收通话失败，收到该回调（userId 是自己的 ID）。

```
TUICallKitEvent.addEventListener('onCallCancelled', (res) => {
  console.log('onCallCancelled', res);
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	用户的 ID

onCallBegin

表示通话接通，主叫和被叫都可以收到，您可以通过监听这个事件来开启云端录制、内容审核等流程。

```
TUICallKitEvent.addEventListener('onCallBegin', (res) => {
  console.log('onCallBegin', JSON.stringify(res));
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.roomID	Number	此次通话的音视频房间 ID，目前仅支持数字房间号，后续版本会支持字符串房间号
res.callMediaType	Number	通话的媒体类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)
res.callRole	Number	角色，枚举类型：主叫(callRole = 1)、被叫(callRole = 2)

onCallEnd

表示通话结束，主叫和被叫都可以收到，您可以通过监听这个事件来显示通话时长、通话类型等信息，或者来停止云端的录制流程。

```
TUICallKitEvent.addEventListener('onCallEnd', (res) => {  
  console.log('onCallEnd', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.roomID	Number	此次通话的音视频房间 ID，目前仅支持数字房间号，后续版本会支持字符串房间号
res.callMediaType	Number	通话的媒体类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)
res.callRole	Number	角色，枚举类型：主叫(callRole = 1)、被叫(callRole = 2)
res.totalTime	Number	此次通话的时长，单位：秒

 注意

客户端的事件一般都会随着杀进程等异常事件丢失掉，如果您需要通过监听通话时长来完成计费逻辑，建议使用 REST API 来完成这类流程。

onCallMediaTypeChanged

表示通话的媒体类型发生变化。

```
TUICallKitEvent.addEventListener('onCallMediaTypeChanged', (res) => {  
  console.log('onCallMediaTypeChanged', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.oldCallMediaType	Number	旧的通话类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)
res.newCallMediaType	Number	新的通话类型，比如：语音通话(callMediaType = 1)、视频通话(callMediaType = 2)

onUserReject

通话被拒绝的回调，在1v1 通话中，只有主叫方会收到拒绝回调，在群组通话中，所有被邀请者都可以收到该回调。

```
TUICallKitEvent.addEventListener('onUserReject', (res) => {  
  console.log('onUserReject', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	拒绝用户的 ID

onUserNoResponse

对方无回应的回调。

```
TUICallKitEvent.addListener('onUserNoResponse', (res) => {  
  console.log('onUserNoResponse', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	无响应用户的 ID

onUserLineBusy

通话忙线回调。

```
TUICallEngine.addListener('onUserLineBusy', (res) => {  
  console.log('onUserLineBusy', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	忙线用户的 ID

onUserJoin

有用户进入此次通话的回调。

```
TUICallEngine.addListener('onUserJoin', (res) => {  
  console.log('onUserJoin', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数

res.userId	String	加入当前通话的用户 ID
------------	--------	--------------

onUserLeave

有用户离开此次通话的回调。

```
TUICallEngine.addEventListener('onUserLeave', (res) => {  
  console.log('onUserLeave', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	离开当前通话的用户 ID

onUserVideoAvailable

用户是否开启视频上行回调。

```
TUICallEngine.addEventListener('onUserVideoAvailable', (res) => {  
  console.log('onUserVideoAvailable', JSON.stringify(res));  
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	通话用户 ID
res.isVideoAvailable	boolean	用户视频是否可用

onUserAudioAvailable

用户是否开启音频上行回调。

```
TUICallEngine.addEventListener('onUserAudioAvailable', (res) => {
```

```
console.log('onUserAudioAvailable', JSON.stringify(res));
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.userId	String	用户 ID
res.isAudioAvailable	boolean	用户音频是否可用

onUserVoiceVolumeChanged

用户通话音量的回调。

```
TUICallEngine.addListener('onUserVoiceVolumeChanged', (res) => {
  console.log('onUserVoiceVolumeChanged', JSON.stringify(res));
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数
res.volumeMap	Map<String, Integer>	音量表，根据每个 userId 可以获取对应用户的音量大小，音量最小值为0，音量最大值为100

onUserNetworkQualityChanged

用户网络质量的回调。

```
TUICallEngine.addListener('onUserNetworkQualityChanged', (res) => {
  console.log('onUserNetworkQualityChanged', JSON.stringify(res));
});
```

参数如下表所示：

参数	类型	含义
res	Object	回调参数

res.networkQualityList	List	网络状态，根据每个 userId 可以获取对应用户当前的网络质量
------------------------	------	----------------------------------

onKickedOffline

当前用户被踢下线：此时可以 UI 提示用户，并再次重新调用初始化。

```
TUICallingEvent.addEventListener('onKickedOffline', (res) => {
  console.log('onKickedOffline', JSON.stringify(res));
});
```

onUserSigExpired

在线时票据过期：此时您需要生成新的 userSig，并再次重新调用初始化。

```
TUICallEngine.addEventListener('onUserSigExpired', (res) => {
  console.log('onUserSigExpired', JSON.stringify(res));
});
```

CustomViewClickEvent

点击新增自定义按钮时会触发该事件。

```
uni.$TUICallKitEvent.addEventListener('CustomViewClickEvent', (res) => {
  console.log(res.viewId);
});
```

参数	类型	含义
res	Object	回调参数
res.viewId	String	触发的点击事件的自定义按钮 viewId

Flutter

API 概览

最近更新时间：2024-10-14 17:24:31

TUICallKit (含 UI 接口)

TUICallKit API 是音视频通话组件的**含 UI 接口**，使用TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景。

API	描述
login	登录
logout	登出
setSelfInfo	设置用户的昵称、头像
call	发起 1v1 通话
groupCall	发起群组通话
joinInGroupCall	主动加入当前的群组通话中
setCallingBell	设置自定义来电铃声
enableMuteMode	开启/关闭静音模式
enableVirtualBackground	开启/关闭虚拟背景功能

TUICallEngine (无 UI 接口)

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套 API 根据您的业务需求自定义封装。

API	描述
init	初始化 SDK 函数，完成包含通话服务鉴权在内初始化动作
unInit	反初始化函数，释放 TUICallEngine 使用到的相关资源
addObserver	增加事件回调
removeObserver	移除回调接口

<code>call</code>	发起 1v1 通话
<code>groupCall</code>	发起群组通话
<code>accept</code>	接听通话
<code>reject</code>	拒绝通话
<code>hangup</code>	结束通话
<code>ignore</code>	忽略通话
<code>inviteUser</code>	在群组通话中，邀请其他人加入
<code>joinInGroupCall</code>	主动加入当前的群组通话中
<code>switchCallMediaType</code>	切换通话媒体类型，例如视频通话切音频通话
<code>startRemoteView</code>	开始订阅远端用户视频流
<code>stopRemoteView</code>	停止订阅远端用户视频流
<code>openCamera</code>	开启摄像头
<code>closeCamera</code>	关闭摄像头
<code>switchCamera</code>	切换前后摄像头
<code>openMicrophone</code>	打开麦克风
<code>closeMicrophone</code>	关闭麦克风
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/免提）
<code>setSelfInfo</code>	设置用户的头像、昵称
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式（尊享版套餐支持）
<code>setVideoRenderParams</code>	设置用户视频画面的渲染模式
<code>setVideoEncoderParams</code>	设置视频编码的编码参数
<code>queryRecentCalls</code>	查询通话记录
<code>deleteRecordCalls</code>	删除通话记录
<code>setBlurBackground</code>	设置视频模糊效果
<code>setVirtualBackground</code>	设置虚拟背景图片

setBeautyLevel

设置美颜级别

TUICallObserver

TUICallObserver 是回调事件类，您可以通过此回调，来监听自己感兴趣的回调事件。

API	描述
onError	通话过程中错误回调
onCallReceived	通话请求的回调
onCallCancelled	通话取消的回调
onCallBegin	通话接通的回调
onCallEnd	通话结束的回调
onCallMediaTypeChanged	通话媒体类型发生改变的回调
onUserReject	xxxx 用户拒绝通话的回调
onUserNoResponse	xxxx 用户不响应的回调
onUserLineBusy	xxxx 用户忙线的回调
onUserJoin	xxxx 用户加入通话的回调
onUserLeave	xxxx 用户离开通话的回调
onUserVideoAvailable	xxx 用户是否有视频流的回调
onUserAudioAvailable	xxx 用户是否有音频流的回调
onUserVoiceVolumeChanged	所有用户音量大小的反馈回调
onUserNetworkQualityChanged	所有用户网络质量的反馈回调
onKickedOffline	当前用户被移下线
onUserSigExpired	在线时票据过期

TUICallKit

最近更新时间：2024-12-11 16:36:12

TUICallKit API 简介

TUICallKit API 是音视频通话组件的含 UI 接口，使用TUICallKit API，您可以通过简单接口快速实现一个类微信的音视频通话场景，更详细的接入步骤，详情请参见 [快速接入（TUICallKit）](#)。

API 概览

API	描述
login	登录
logout	登出
setSelfInfo	设置用户的昵称、头像
call	发起 1v1 通话
groupCall	发起群组通话
joinInGroupCall	主动加入当前的群组通话中
enableMuteMode	开启/关闭静音模式
enableFloatWindow	开启/关闭悬浮窗功能
setCallingBell	设置自定义来电铃声
enableVirtualBackground	开启/关闭虚拟背景功能

API 详情

login

登录。

```
Future<TUIResult> login(int sdkAppId, String userId, String userSig);
```

参数	类型	含义
sdkAppId	int	用户 SDKAppID

userId	String	用户 ID
userSig	String	用户签名 userSig (计算方法详见)
返回值	TUIResult	包含 code 和 message 信息： code 为空 ("") 表示调用成功 code 不为空 ("") 表示调用失败，失败原因见 message

logout

登出。

```
Future<void> logout ()
```

setSelfInfo

设置用户昵称、头像。用户昵称不能超过500字节，用户头像必须是URL格式。

```
Future<TUIResult> setSelfInfo(String nickname, String avatar)
```

参数	类型	含义
nickName	String	目标用户的昵称，非必填
avatar	String	目标用户的头像，非必填
返回值	TUIResult	包含 code 和 message 信息： code 为空 ("") 表示调用成功 code 不为空 ("") 表示调用失败，失败原因见 message

call

拨打电话（1v1通话）。

```
Future<void> call(String userId, TUICallMediaType callMediaType,  
[TUICallParams? params])
```

参数	类型	含义
userId	String	目标用户的 userId
callMediaType	TUICallMediaType	通话的媒体类型，比如： <code>TUICallMediaType.video</code> 或

```
TUICallMediaType.audio
```

groupCall

发起群组通话，注意：使用群组通话前需要创建IM 群组，如果已经创建，请忽略。

```
Future<void> groupCall(String groupId, List<String> userIdList,
    TUICallMediaType callMediaType, [TUICallParams? params])
```

参数	类型	含义
groupId	String	此次群组通话的群 ID
userIdList	List<String> >	目标用户的userId 列表
callMediaType	TUICallMe diaType	通话的媒体类型，比如： TUICallMediaType.video 或 TUICallMediaType.audio

joinInGroupCall

加入群组中已有的音视频通话。

```
Future<void> joinInGroupCall(TUIRoomId roomId, String groupId,
    TUICallMediaType callMediaType)
```

参数	类型	含义
roomId	TUIRoomI D	此次通话的音视频房间 ID
groupId	String	此次群组通话的群 ID
callMediaType	TUICallMe diaType	通话的媒体类型，比如： TUICallMediaType.video 或 TUICallMediaType.audio

enableMuteMode

开启后，收到通话请求，不会播放来电铃声。

```
Future<void> enableMuteMode(bool enable)
```

参数	类型	含义
enable	bool	开启、关闭静音；true 表示开启静音

enableFloatWindow

开启/关闭悬浮窗功能，设置为 false 后，通话界面左上角的悬浮窗按钮会隐藏。

```
Future<void> enableFloatWindow(bool enable)
```

参数	类型	含义
enable	bool	开启、关闭悬浮窗功能；true 表示开启浮窗

setCallingBell

自定义来电铃声。

```
Future<void> setCallingBell(String assetName)
```

参数	类型	含义
assetName	String	铃声的路径，需要将铃声文件添加至主工程的 assets 资源中。

enableVirtualBackground

开启/关闭虚拟背景功能，开启虚拟背景功能后，您可以在 UI 上显示模糊背景的功能按钮，点击按钮可直接启用模糊背景功能。

```
Future<void> enableVirtualBackground(bool enable)
```

参数	类型	含义
enable	bool	开启、关闭静音；true 表示开启静音

TUICallEngine

最近更新时间：2024-04-30 09:47:21

TUICallEngine API 简介

TUICallEngine API 是音视频通话组件的**无 UI 接口**，如果 TUICallKit 的交互并不满足您的需求，您可以使用这套接口自己封装交互。

API 概览

API	描述
init	初始化 SDK 函数，完成包含通话服务鉴权在内初始化动作
unInit	反初始化函数，释放 TUICallEngine 使用到的相关资源。
addObserver	增加事件回调
removeObserver	移除回调接口
call	发起 1v1 通话
groupCall	发起群组通话
accept	接听通话
reject	拒绝通话
hangup	结束通话
ignore	忽略通话
inviteUser	在群组通话中，邀请其他人加入
joinInGroupCall	主动加入当前的群组通话中
switchCallMediaType	切换通话媒体类型，例如视频通话切音频通话
startRemoteView	开始订阅远端用户视频流
stopRemoteView	停止订阅远端用户视频流
openCamera	开启摄像头
closeCamera	关闭摄像头
switchCamera	切换前后摄像头

<code>openMicrophone</code>	打开麦克风
<code>closeMicrophone</code>	关闭麦克风
<code>selectAudioPlaybackDevice</code>	选择音频播放设备（听筒/免提）
<code>setSelfInfo</code>	设置用户的头像、昵称
<code>enableMultiDeviceAbility</code>	开启/关闭 TUICallEngine 的多设备登录模式（尊享版套餐支持）
<code>setVideoRenderParams</code>	设置用户视频画面的渲染模式
<code>setVideoEncoderParams</code>	设置视频编码的编码参数
<code>queryRecentCalls</code>	查询通话记录
<code>deleteRecordCalls</code>	删除通话记录
<code>setBlurBackground</code>	设置视频模糊效果
<code>setVirtualBackground</code>	设置虚拟背景图片
<code>setBeautyLevel</code>	设置美颜级别

API 详情

init

初始化函数，请在使用所有功能之前先调用该函数，以便完成包含通话服务鉴权在内初始化动作。

```
Future<TUIResult> init(int sdkAppID, String userId, String userSig)
```

unInit

反初始化函数，释放 TUICallEngine 使用到的相关资源。

```
Future<TUIResult> unInit()
```

addObserver

添加回调接口，您可以通过这个接口，监听 TUICallObserver 相关的事件回调。

```
Future<void> addObserver(TUICallObserver observer)
```

removeObserver

移除回调接口。

```
Future<void> removeObserver(TUICallObserver observer)
```

call

拨打电话（1v1通话）。

```
Future<TUIResult> call(String userId, TUICallMediaType mediaType,
    TUICallParams params)
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId
mediaType	TUICallMediaType	通话的媒体类型，例如视频通话、语音通话
params	TUICallParams	通话参数扩展字段，例如：房间号、通话邀请超时时间，离线推送自定义内容等

groupCall

发起群组通话。

⚠ 注意：
使用群组通话前需要创建 IM 群组，如果已经创建，请忽略。

```
Future<TUIResult> groupCall(String groupId, List<String> userIdList,
    TUICallMediaType mediaType, TUICallParams params)
```

参数	类型	含义
groupId	String	此次群组通话的群 ID
userIdList	List<String>	目标用户的 userId 列表
mediaType	TUICallMediaType	通话的媒体类型，例如视频通话、语音通话

e		
params	TUICallParams	通话参数扩展字段，例如：房间号、通话邀请超时时间，离线推送自定义内容等

accept

接收当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数接听来电。

```
Future<TUIResult> accept()
```

reject

拒绝当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数拒绝来电。

```
Future<TUIResult> reject()
```

ignore

忽略当前通话，当您作为被叫收到 `onCallReceived()` 的回调时，可以调用该函数忽略来电，此时主叫会收到 `onUserLineBusy` 的回调。

备注：如果您的业务中存在直播、会议等场景，在直播/会议中的情况时，也可以调用这个函数来忽略此次来电。

```
Future<TUIResult> ignore()
```

hangup

挂断当前通话，当您处于通话中，可以调用该函数结束通话。

```
Future<TUIResult> hangup()
```

inviteUser

邀请用户加入此次群组通话。

使用场景：一个群组通话中的用户主动邀请其他人时使用。

```
Future<void> ininviteUser(List<String> userIdList, TUICallParams params,
TUIValueCallback callback)
```

参数	类型	含义
----	----	----

userIdList	List<String>	目标用户的 userId 列表
params	TUICallParams	通话参数扩展字段，例如：离线推送自定义内容

joinInGroupCall

主动加入此次群组通话。

使用场景：群组内用户主动加入此次群组通话使用。

```
Future<TUIResult> joinInGroupCall(TUIRoomId roomId, String groupId,
TUICallMediaType mediaType)
```

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间 ID
groupId	String	此次群组通话的群 ID
mediaType	TUICallMediaType	通话的媒体类型，例如视频通话、语音通话

switchCallMediaType

切换视频通话到语音通话。

```
Future<void> switchCallMediaType(TUICallMediaType mediaType)
```

参数	类型	含义
mediaType	TUICallMediaType	通话的媒体类型，例如视频通话、语音通话

startRemoteView

设置显示视频画面的 View 对象。

```
Future<void> startRemoteView(String userId, intviewId)
```

参数	类型	含义
userId	String	目标用户的 userId

intviewId	int	显示视频画面的widget对应的Id
-----------	-----	--------------------

stopRemoteview

停止订阅远端用户的视频数据。

```
Future<void> stopRemoteView(String userId)
```

参数	类型	含义
userId	String	目标用户的 userId

openCamera

开启摄像头。

```
Future<TUIResult> openCamera(TUICamera camera, int? viewId)
```

参数	类型	含义
camera	TUICamera	前置/后置 摄像头
viewId	int	显示视频画面的 widget 对应的 Id

closeCamera

关闭摄像头。

```
Future<void> closeCamera()
```

switchCamera

切换前后摄像头。

```
Future<void> switchCamera(TUICamera camera)
```

参数	类型	含义
camera	TUICamera	前置/后置 摄像头

openMicrophone

打开麦克风。

```
Future<TUIResult> openMicrophone()
```

closeMicrophone

关闭麦克风。

```
Future<void> closeMicrophone()
```

selectAudioPlaybackDevice

选择音频播放设备。

目前支持听筒、扬声器，在通话场景中，可以使用这个接口来开启/关闭免提模式。

```
Future<void> selectAudioPlaybackDevice(TUIAudioPlaybackDevice device)
```

参数	类型	含义
device	TUIAudioPlaybackDevice	听筒/扬声器

setSelfInfo

设置用户昵称、头像。

用户昵称不能超过500字节，用户头像必须是 URL 格式。

```
Future<TUIResult> setSelfInfo(String nickname, String avatar)
```

参数	类型	含义
nickname	String	用户昵称
avatar	String	用户头像（格式为 URL）

enableMultiDeviceAbility

开启/关闭 TUICallEngine 的多设备登录模式（尊享版套餐支持）。

```
Future<TUIResult> enableMultiDeviceAbility(bool enable)
```

setVideoRenderParams

设置用户视频画面的渲染模式。

```
Future<TUIResult> setVideoRenderParams(String userId, VideoRenderParams params)
```

参数如下表所示：

参数	类型	含义
userId	String	目标用户的 userId
params	VideoRenderParams	视频画面渲染参数，例如：画面旋转角度、填充模式

setVideoEncoderParams

设置视频编码的编码参数。

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

```
Future<TUIResult> setVideoEncoderParams(VideoEncoderParams params)
```

参数如下表所示：

参数	类型	含义
params	VideoEncoderParams	视频编码的参数，例如：分辨率、视频宽高比

queryRecentCalls

查询通话记录。

```
Future<void> queryRecentCalls(TUICallRecentCallsFilter filter, TUIValueCallback callback)
```

参数如下表所示：

参数	类型	含义
filter	TUICallRecentCallsFilter	查询过滤条件

deleteRecordCalls

删除通话记录。

```
Future<void> deleteRecordCalls(List<String> callIdList, TUIValueCallback callback)
```

参数如下表所示：

参数	类型	含义
callIdList	List<String>	待删除记录的Id列表

setBeautyLevel

设置美颜级别。

```
Future<TUIResult> setBeautyLevel(double level)
```

参数如下表所示：

参数	类型	含义
level	double	美颜等级，使用范围0.0~9.0

setBlurBackground

设置视频模糊效果。

```
void setBlurBackground(int level, Function<int code, String message>? errorCallback)
```

参数如下表所示：

参数	类型	含义
level	int	0: 关闭 1: 低 2: 中 3: 高

setVirtualBackground

设置虚拟背景图片。

```
void setVirtualBackground(String imagePath, Function<int, String> errorCallback)
```

参数	类型	含义
imagePath	String	图片文件名，需要将文件添加至主工程的 assets 资源中。

TUICallObserver

最近更新时间：2024-11-29 09:13:12

TUICallObserver API 简介

TUICallObserver 是 TUICallKit 对应的回调事件类，您可以通过此回调，来监听自己感兴趣的回调事件。

回调事件概览

API	描述
onError	通话过程中错误回调
onCallReceived	通话请求的回调
onCallCancelled	通话取消的回调
onCallBegin	通话接通的回调
onCallEnd	通话结束的回调
onCallMediaTypeChanged	通话媒体类型发生改变的回调
onUserReject	xxxx 用户拒绝通话的回调
onUserNoResponse	xxxx 用户不响应的回调
onUserLineBusy	xxxx 用户忙线的回调
onUserJoin	xxxx 用户加入通话的回调
onUserLeave	xxxx 用户离开通话的回调
onUserVideoAvailable	xxxx 用户是否有视频流的回调
onUserAudioAvailable	xxxx 用户是否有音频流的回调
onUserVoiceVolumeChanged	所有用户音量大小的反馈回调
onUserNetworkQualityChanged	所有用户网络质量的反馈回调
onKickedOffline	当前用户被移下线
onUserSigExpired	在线时票据过期

回调事件详情

通过 `addObserver` 监听 Flutter 插件抛出的事件。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
  onError: (int code, String message) {  
    //您的回调处理逻辑  
  }, onCallCancelled: (String callerId) {  
    //您的回调处理逻辑  
  }, onCallBegin: (TUIRoomId roomId, TUICallMediaType callMediaType,  
TUICallRole callRole) {  
    //您的回调处理逻辑  
  }, onCallEnd: (TUIRoomId roomId, TUICallMediaType callMediaType,  
TUICallRole callRole, double totalTime) {  
    //您的回调处理逻辑  
  }, onCallMediaTypeChanged: (TUICallMediaType oldCallMediaType,  
TUICallMediaType newCallMediaType) {  
    //您的回调处理逻辑  
  }, onUserReject: (String userId) {  
    //您的回调处理逻辑  
  }, onUserNoResponse: (String userId) {  
    //您的回调处理逻辑  
  }, onUserLineBusy: (String onUserLineBusy) {  
    //您的回调处理逻辑  
  }, onUserJoin: (String userId) {  
    //您的回调处理逻辑  
  }, onUserLeave: (String userId) {  
    //您的回调处理逻辑  
  }, onUserVideoAvailable: (String userId, bool isVideoAvailable) {  
    //您的回调处理逻辑  
  }, onUserAudioAvailable: (String userId, bool isAudioAvailable) {  
    //您的回调处理逻辑  
  }, onUserNetworkQualityChanged: (List<TUINetworkQualityInfo>  
networkQualityList) {  
    //您的回调处理逻辑  
  }, onCallReceived: (String callerId, List<String> calleeIdList,  
String groupId, TUICallMediaType callMediaType) {  
    //您的回调处理逻辑  
  }, onUserVoiceVolumeChanged: (Map<String, int> volumeMap) {  
    //您的回调处理逻辑  
  }, onKickedOffline: () {  
    //您的回调处理逻辑  
  }, onUserSigExpired: () {
```

```
//您的回调处理逻辑
}
));
```

onError

错误事件回调。

❗ 说明

SDK 不可恢复的错误，一定要监听，并分情况给用户适当的界面提示。

```
TUICallEngine.instance.addObserver(TUICallObserver(
    onError: (int code, String message) {
        //您的回调处理逻辑
    }
));
```

参数如下表所示：

参数	类型	含义
code	int	错误码
message	String	错误信息

onCallReceived

收到一个新的来电请求回调，被叫会收到，您可以通过监听这个事件，来决定是否显示通话接听界面。

```
TUICallEngine.instance.addObserver(TUICallObserver(
    onCallReceived: (String callerId, List<String> calleeIdList, String
groupId, TUICallMediaType callMediaType) {
        //您的回调处理逻辑
    }
));
```

参数如下表所示：

参数	类型	含义
callerId	String	主叫 ID（邀请方）

calleeldList	List<String>	被叫 ID 列表（被邀请方）
groupId	String	群组 ID
callMediaType	TUICallMediaType	通话的媒体类型，示例： <code>TUICallMediaType.video</code> 或 <code>TUICallMediaType.audio</code>

onCallCancelled

表示此次通话主叫取消、被叫超时、拒接等，涉及多个场景，您可以通过监听这个事件来实现类似未接来电、重置 UI 状态等显示逻辑。

- 主叫取消：主叫收到该回调（userId 为自己）；被叫收到该回调（userId 为主叫的 ID）
- 被叫超时：主叫会同时收到 `onUserNoResponse` 和 `onCallCancelled` 回调（userId 是自己的 ID）；被叫收到 `onCallCancelled` 回调（userId 是自己的 ID）
- 被叫拒接：主叫会同时收到 `onUserReject` 和 `onCallCancelled` 回调（userId 是自己的 ID）；被叫收到 `onCallCancelled` 回调（userId 是自己的 ID）
- 被叫忙线：主叫会同时收到 `onUserLineBusy` 和 `onCallCancelled` 回调（userId 是自己的 ID）
- 异常中断：被叫接收通话失败，收到该回调（userId 是自己的 ID）

```
TUICallEngine.instance.addObserver(TUICallObserver(
    onCallCancelled: (String userId) {
        //您的回调处理逻辑
    }
));
```

参数如下表所示：

参数	类型	含义
userId	String	用户的 ID

onCallBegin

表示通话接通，主叫和被叫都可以收到，您可以通过监听这个事件来开启云端录制、内容审核等流程。

```
TUICallEngine.instance.addObserver(TUICallObserver(
    onCallBegin: (TUIRoomId roomId, TUICallMediaType callMediaType,
    TUICallRole callRole) {
        //您的回调处理逻辑
    }
});
```

```
));
```

参数如下表所示：

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间
callMediaType	TUICallMediaType	通话的媒体类型，示例： TUICallMediaType.video 或 TUICallMediaType.audio
callRole	TUICallRole	角色，枚举类型：主叫(TUICallRole.caller)、被叫(TUICallRole.called)

onCallEnd

表示通话接通，主叫和被叫都可以收到，您可以通过监听这个事件来显示通话时长、通话类型等信息，或者来停止云端的录制流程。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onCallEnd: (TUIRoomId roomId, TUICallMediaType callMediaType,  
    TUICallRole callRole, double totalTime) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
roomId	TUIRoomId	此次通话的音视频房间 Id
callMediaType	TUICallMediaType	通话的媒体类型，示例： TUICallMediaType.video 或 TUICallMediaType.audio
callRole	Number	角色，枚举类型：主叫(callRole = 1)、被叫(callRole = 2)
totalTime	double	此次通话的时长，单位 ms

⚠ 注意：

客户端的事件一般都会随着杀进程等异常事件丢失掉，如果您需要通过监听通话时长来完成计费逻辑，建议使用 REST API 来完成这类流程。

onCallMediaTypeChanged

表示通话的媒体类型发生变化。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onCallMediaTypeChanged: (TUICallMediaType oldCallMediaType,  
    TUICallMediaType newCallMediaType) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
oldCallMediaType	TUICallMediaType	通话的媒体类型，示例： <code>TUICallMediaType.video</code> 或 <code>TUICallMediaType.audio</code>
newCallMediaType	TUICallMediaType	通话的媒体类型，示例： <code>TUICallMediaType.video</code> 或 <code>TUICallMediaType.audio</code>

onUserReject

通话被拒绝的回调，在1v1 通话中，只有主叫方会收到拒绝回调，在群组通话中，所有被邀请者都可以收到该回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserReject: (String userId) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
res.userId	String	拒绝用户的 ID

onUserNoResponse

对方无回应的回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserNoResponse: (String userId) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
userId	String	无响应用户的 ID

onUserLineBusy

通话忙线回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserLineBusy: (String onUserLineBusy) {  
        //您的回调处理逻辑  
    },  
));
```

参数如下表所示：

参数	类型	含义
userId	String	忙线用户的 ID

onUserJoin

有用户进入此次通话的回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserJoin: (String userId) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
userId	String	加入当前通话的用户 ID

onUserLeave

有用户离开此次通话的回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserLeave: (String userId) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
userId	String	离开当前通话的用户 ID

onUserVideoAvailable

用户是否开启视频上行回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserVideoAvailable: (String userId, bool isVideoAvailable) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
userId	String	通话用户 ID
isVideoAvailable	bool	用户视频是否可用

onUserAudioAvailable

用户是否开启音频上行回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserAudioAvailable: (String userId, bool isAudioAvailable) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
userId	String	用户 ID
isAudioAvailable	bool	用户音频是否可用

onUserVoiceVolumeChanged

用户通话音量的回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserVoiceVolumeChanged: (Map<String, int> volumeMap) {  
        //您的回调处理逻辑  
    }  
));
```

参数如下表所示：

参数	类型	含义
volumeMap	Map<String, int>	音量表，根据每个 userId 可以获取对应用户的音量大小，音量最小值为 0，音量最大值为 100

onUserNetworkQualityChanged

用户网络质量的回调。

```
TUICallEngine.instance.addObserver(TUICallObserver(  
    onUserNetworkQualityChanged: (List<TUINetworkQualityInfo>  
networkQualityList) {  
        //您的回调处理逻辑  
    }  
));
```

```
//TUINetworkQualityInfo的定义如下:
class TUINetworkQualityInfo {
    String userId;
    TUINetworkQuality quality;
    TUINetworkQualityInfo({required this.userId, required
this.quality});
}
// TUINetworkQuality的定义如下:
enum TUINetworkQuality {
    unknown,
    excellent,
    good,
    poor,
    bad,
    vBad,
    down
}
```

参数如下表所示:

参数	类型	含义
networkQualityList	List< TUINetworkQualityInfo >	网络状态，根据每个 userId 可以获取对应用户当前的网络质量

onKickedOffline

当前用户被移下线：此时可以 UI 提示用户，并再次重新调用初始化。

```
TUICallEngine.instance.addObserver(TUICallObserver(
    onKickedOffline: () {
        //您的回调处理逻辑
    }
));
```

onUserSigExpired

在线时票据过期：此时您需要生成新的 userSig，并再次重新调用初始化。

```
TUICallEngine.instance.addObserver(TUICallObserver(
    onUserSigExpired: () {
        //您的回调处理逻辑
    }
});
```

```
));
```

类型定义

最近更新时间：2024-11-25 15:41:12

常用结构

TUIResult

调用 api 的返回值

类型	类型	描述
code	String	code 为空 "" 表示调用成功，code 不为空 "" 表示调用失败
message	String	失败原因

TUIRoomId

房间 ID

类型	类型	描述
intRoomId	int	数字房间号
strRoomId	String	字符串房间号

VideoRenderParams

视频画面的渲染参数

类型	类型	描述
fillMode	FillMode	视频画面填充模式
rotation	Rotation	视频画面旋转方向

VideoEncoderParams

视频编码参数

类型	类型	描述
resolution	Resolution	视频分辨率

resolutionMode	ResolutionMode	视频宽高比模式
----------------	----------------	---------

TUICallParams

通话参数

类型	类型	描述
roomId	TUIRoomId	房间 ID
offlinePushInfo	TUIOfflinePushInfo	厂商离线推送配置信息
timeout	String	超时时常
userData	String	用户自定义数据

TUIOfflinePushInfo

厂商离线推送配置信息，详情配置步骤请参考：[离线唤醒](#)。

类型	类型	描述
title	String	离线推送展示通知栏标题
desc	String	离线推送展示通知栏内容
ignoreIOSBadge	bool	离线推送忽略 badge 计数（仅对 iOS 生效），如果设置为 true，在 iOS 接收端，这条消息不会使 APP 的应用图标未读计数增加。
iOSSound	String	离线推送声音设置（仅对 iOS 生效）。当 sound = IOS_OFFLINE_PUSH_NO_SOUND，表示接收时不会播放声音。当 sound = IOS_OFFLINE_PUSH_DEFAULT_SOUND，表示接收时播放系统声音。如果要自定义 iOSSound，需要先把语音文件链接进 Xcode 工程，然后把语音文件名（带后缀）设置给 iOSSound
androidSound	String	离线推送声音设置（仅对 Android 生效, IMSDK 6.1 及以上版本支持）。只有华为和谷歌手机支持设置声音提示，小米手机设置声音提示，请您参照： 小米自定义铃声 。另外，谷歌手机 FCM 推送在 Android 8.0 及以上系统设置声音提示，必须调用

		setAndroidFCMChannelID 设置好 channelId，才能生效
androidOPPOChannelID	String	离线推送设置 OPPO 手机 8.0 系统及以上的渠道 ID
androidVIVOClassification	int	VIVO 推送消息分类 (待废弃接口，VIVO 推送服务于 2023年4月3日优化消息分类规则，推荐使用 setAndroidVIVOCategory 设置消息类别)。0：运营消息 1：系统消息，默认取值为 1
androidXiaoMiChannelID	String	小米手机 8.0 系统及以上的渠道 ID
androidFCMChannelID	String	FCM 通道手机 8.0 系统及以上的渠道 ID
androidHuaWeiCategory	String	华为推送消息分类，详见： 华为消息分类标准
isDisablePush	bool	是否关闭推送（默认开启推送）
iOSPushType	TUICallIOSOfflinePushType	iOS 离线推送类型，默认：APNs

TUICallRecords

通话记录信息

类型	类型	描述
callId	String	通话记录 ID
inviter	String	邀请者 ID
inviteList	List<String>	被邀请用户 ID 列表
scene	TUICallScene	通话场景
mediaType	TUICallMediaType	通话媒体类型
groupId	String	群组 ID
role	TUICallRole	通话角色
result	TUICallResultType	通话结果类型

beginTime	int	开始时间
totalTime	int	总时间

TUICallRecentCallsFilter

通话记录过滤条件

类型	类型	描述
begin	double	开始时间
end	double	结束时间
resultType	TUICallResult Type	通话记录类型

枚举定义

TUICallMediaType

通话类型

类型	取值	描述
none	0	未知类型
audio	1	语音通话
video	2	视频通话

TUICallRole

通话角色

类型	取值	描述
none	0	未知类型
caller	1	主叫（邀请方）
called	2	被叫（被邀请方）

TUICallStatus

通话状态

类型	取值	描述
none	0	未知类型
waiting	1	通话等待中
accept	2	通话已接听

TUICallScene

通话场景

类型	取值	描述
groupCall	0	群组通话
multiCall	1	匿名群组通话（暂不支持，敬请期待）
singleCall	2	单人通话

TUINetworkQuality

网络质量

类型	取值	描述
unknown	0	未定义
excellent	1	当前网络非常好
good	2	当前网络比较好
poor	3	当前网络一般
bad	4	当前网络较差
vBad	5	当前网络很差
down	6	当前网络不满足通话的最低要求

FillMode

视频画面填充模式

类型	取值	描述
fill	0	填充模式：即将画面内容居中等比缩放以充满整个显示区域，超出显示区域的部分将会被裁剪掉，此模式下画面可能

		不完整。
fit	1	适应模式：即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边。

Rotation

视频画面旋转方向

类型	取值	描述
rotation_0	0	不旋转
rotation_90	1	顺时针旋转90度
rotation_180	2	顺时针旋转180度
rotation_270	3	顺时针旋转270度

ResolutionMode

视频宽高比模式

类型	取值	描述
landscape	0	横屏分辨率，例如：Resolution.Resolution_640_360 + ResolutionMode.Landscape = 640 × 360
portrait	1	竖屏分辨率，例如：Resolution.Resolution_640_360 + ResolutionMode.Portrait = 360 × 640

Resolution

视频分辨率

类型	取值	描述
resolution_640_360	0	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps
resolution_640_480	1	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps
resolution_960_540	2	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps

resolution_960_720	3	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps
resolution_1280_720	4	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps
resolution_1920_1080	5	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps

TUICallIOSOfflinePushType

iOS 离线推送类型

类型	取值	描述
APNs	0	普通推送
VoIP	1	VoIP推送

TUICamera

摄像头类型类型

类型	取值	描述
front	0	前置摄像头
back	1	后置摄像头

TUIAudioPlaybackDevice

音频播放路由

类型	取值	描述
speakerphone	0	扬声器
earpiece	1	耳麦

TUICallResultType

通话记录类型

类型	取值	描述
unknown	0	未知

missed	1	未接
incoming	2	接入
outgoing	3	拨出