

音视频通话 SDK

常见问题



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

常见问题

产品概念与购买使用

Android

iOS

Web

小程序

Flutter

常见问题

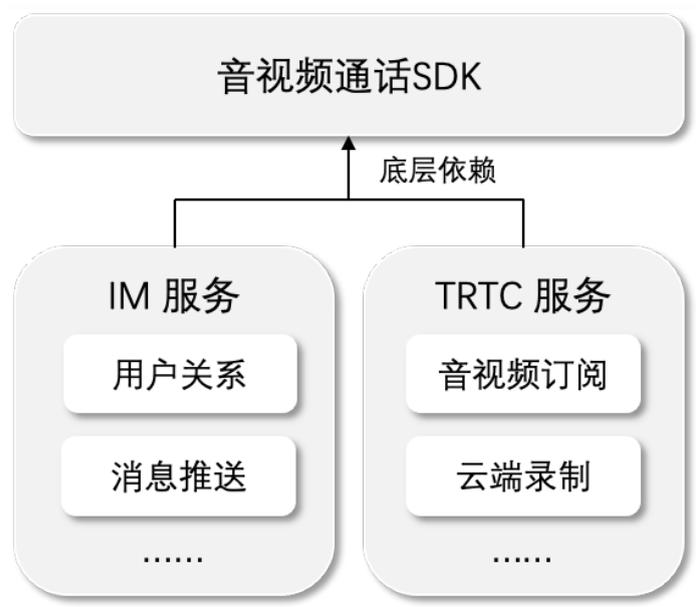
产品概念与购买使用

最近更新时间：2024-11-25 15:41:12

产品概念相关

音视频通话 SDK 与 即时通信 IM 和实时音视频 TRTC 的关系是什么？

音视频通话 SDK 基于即时通信 IM 和实时音视频 TRTC 底层技术封装实现，从而实现音视频通话场景下的各项功能。因此使用音视频通话 SDK 依赖于即时通信 IM 与实时音视频 TRTC 服务，您需要开通并正常使用即时通信 IM 及实时音视频 TRTC 服务才可使用音视频通话 SDK。



音视频通话 SDK 与实时音视频 TRTC 有什么区别？

实时音视频（TRTC）是支撑音视频通话相关功能的基础服务，按基础通话分钟数资源消耗付费，仅提供基础通信底层能力；音视频通话 SDK 封装了即时通信 IM 与实时音视频 TRTC 底层技术，针对通话场景进行深度定制，提供了1v1语音和视频通话、群组通话、中途呼叫等功能，提供微信视频聊天同款 UI 快速集成方案。能够大大降低客户的开发集成难度及开发量，赠送通话时长也更加优惠。

使用音视频通话 SDK 实现视频聊天功能，与使用 TRTC 自行实现视频聊天功能有区别吗？

音视频通话 SDK 基于即时通信 IM 与实时音视频 TRTC 两个PaaS服务实现，提供针对音视频通话场景下双人通话、群组通话、中途呼叫等多个功能，更提供微信同款 UI 方案，仅需3步集成，最快一天即可上线。除了能大大降低客户的开发集成难度及开发量之外，可专属享受通话 AI 降噪、通话卡顿优化等高级功能；通过 TRTC 来实现视频聊天功能，您需要自行集成 TRTC SDK 与 IM SDK，自行处理通话场景下呼叫/接听/挂断等复杂的信令交互及状态判断逻辑，还要考虑到各种异常情况以及通话UI界面的设计。预计需要投入2-3个月的时

间才能实现一个基本的通话场景。考虑到平台的多样性，研发资源的投入还会更多。除此之外，音视频通话 SDK 独有的通话 AI 降噪、户外通话卡顿优化等高级能力无法实现。

音视频通话 SDK 与音视频终端 SDK（腾讯云视立方 SDK）的关系是什么？

腾讯云视立方 SDK 为音视频终端 SDK 产品家族，包含多个子 SDK。基础 SDK 仅提供音视频终端基础能力，含 UI 低代码组件提供包含后台服务及终端 SDK 一整套解决方案，支持含 UI 资源快速集成。

音视频通话 SDK 为其中的子产品之一，如果您有别的终端需求（如直播、短视频剪辑、美颜特效等等），可参考产品家族中的其他子 SDK。



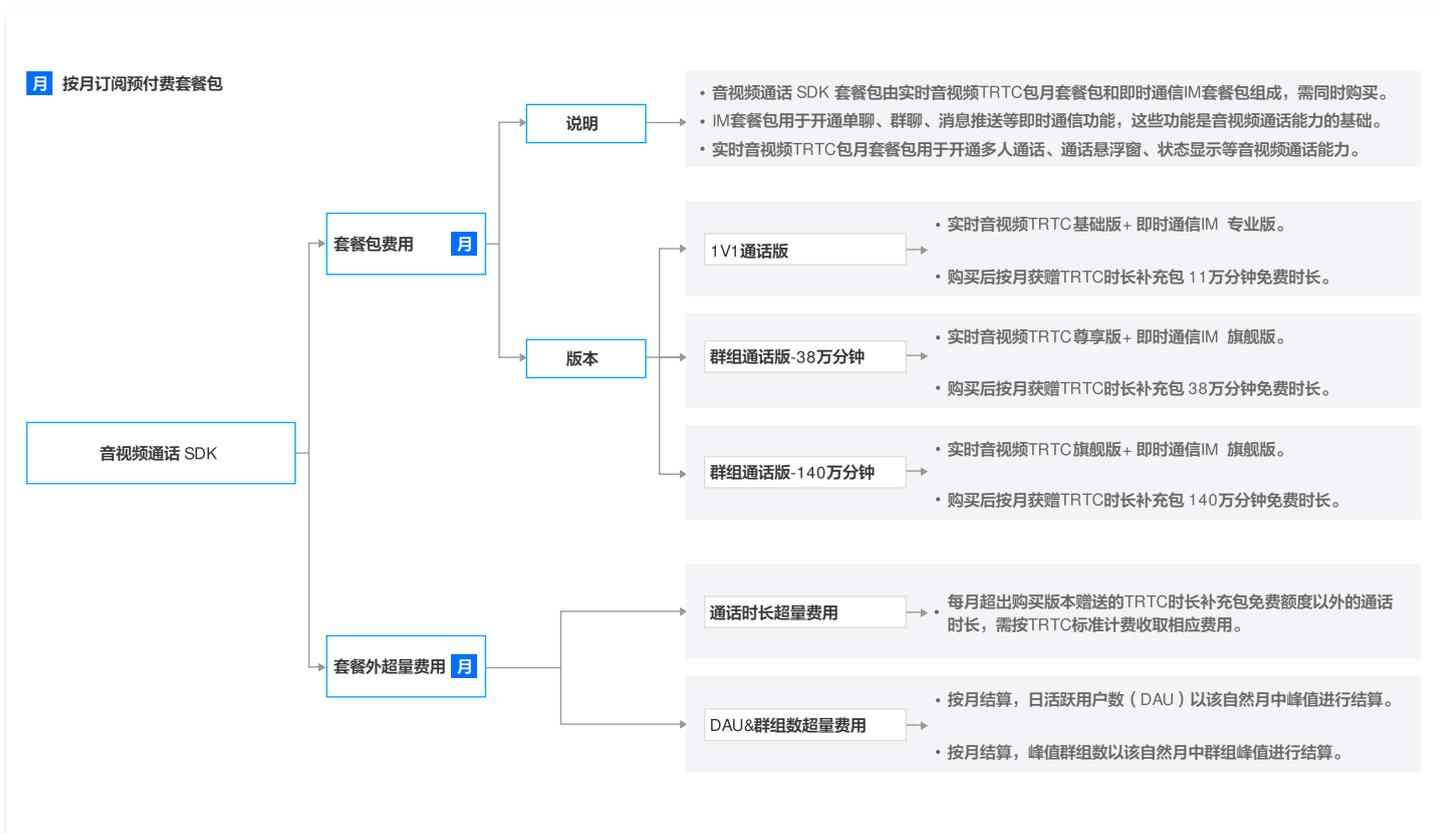
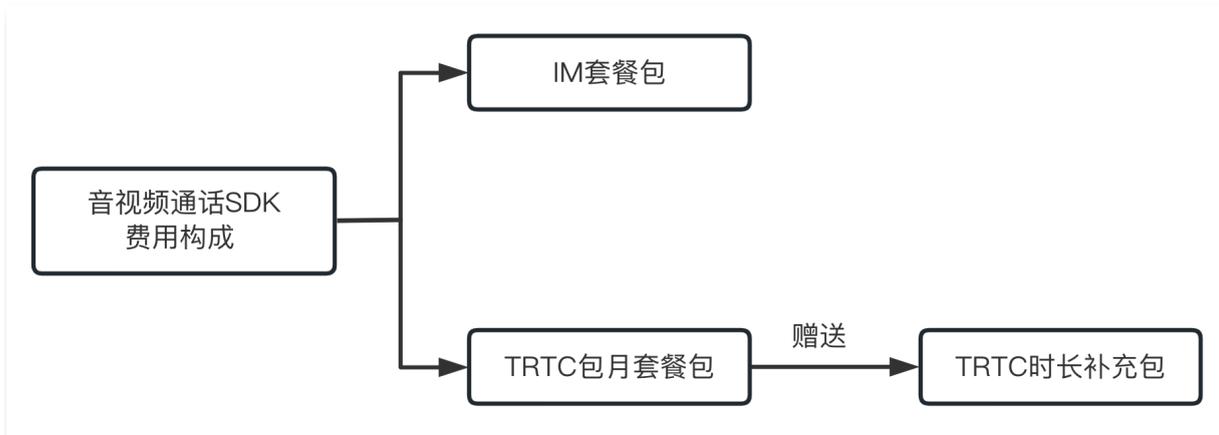
购买音视频通话 SDK 必须要买即时通信 IM 吗？

是的，音视频通话 SDK 依赖于即时通信 IM 与实时音视频 TRTC 服务，您需要开通并正常使用即时通信 IM 及实时音视频 TRTC 服务才可使用音视频通话 SDK。

计费与购买相关

音视频通话 SDK 费用包含哪些？

音视频通话 SDK 费用包含购买即时通信 IM 的费用和实时音视频包月套餐费用两部分，其中购买音视频通话能力包会赠送等价格实时音视频 TRTC 时长补充包，赠送时长包按月赠送，仅在当月生效。具体的功能及对应价格可参见 [功能说明](#)。部分资源如果使用超量，可能会产生后付费账单，详细规则可参见 [计费说明](#)。



购买了实时音视频 TRTC 时长补充包，为什么无法使用音视频通话 SDK ？

TRTC 时长补充包仅包含一定的时长资源，仅可用于抵扣通话时长，无法开通音视频通话 SDK 的使用权限；使用音视频通话 SDK 需要购买即时通信 IM 及实时音视频 TRTC 包月套餐包，购买 TRTC 包月套餐包后会赠送一定量的实时音视频 TRTC 时长补充包。

音视频通话 SDK 赠送的时长补充包与实时音视频 TRTC 购买时长补充包有什么区别？

购买音视频通话 SDK 赠送的时长包抵扣逻辑与实时音视频 TRTC 套餐包一致，但赠送时长包按月赠送，当月生效。若客户账号下有多个时长包，则优先按照到期时间进行抵扣，即最早到期的优先抵扣。

如果我已购买了即时通信 IM，怎么使用音视频通话 SDK？

若您已是即时通信 IM 付费用户，您仅需加购实时音视频包月套餐包即可，其中购买实时音视频包月套餐包会赠送等价格实时音视频 TRTC 时长补充包，赠送时长包按月赠送，当月生效。若您在音视频通话 SDK 购买页购买，可选择仅加购实时音视频包月套餐包或同时为您的即时通信 IM 套餐进行续期。

即时通信 IM 与音视频通话能力包组合搭配是否有限制？如果我是 IM 的专业版能否购买 TRTC 尊享版或旗舰版？

为了保证更好的使用体验，目前即时通信 IM 专业版客户可加购实时音视频 TRTC 基础版；即时通信 IM 旗舰版客户可加购实时音视频 TRTC 尊享版或旗舰版；可参见 [音视频通话 SDK 费用包含](#)。

音视频通话 SDK 如何开启自动续期？

如前所述，音视频通话 SDK 套餐包由即时通信 IM 套餐包和实时音视频 TRTC 包月套餐包组成，您需要为两种资源分别开启自动续期。具体步骤如下：

1. 登录腾讯云控制台 > 费用中心，在左侧导航中，选择 [续费管理](#)；
2. 右侧搜索框中搜索“IM + SDKAppID”，在操作栏中选择 [设为自动续期](#)；
3. 右侧搜索框中搜索“TRTC + SDKAppID”，在操作栏中选择 [设为自动续期](#)。

注意：若您通过音视频通话 SDK 购买页下单，我们会默认为您开启自动续期，您可以前往 [续费管理](#) 取消该功能。

音视频通话 SDK 如何退订？

如果您购买的音视频通话 SDK 预付费套餐包未被使用，腾讯云支持5天内无理由退货退款。对于单个腾讯云主体而言，在为您的 SDKAppID 购买了任意套餐包之日起五天内（含五天），可支持一次一个套餐包或功能包五天无理由退还。详细退款规则参见 [退费说明](#)。

音视频通话 SDK（TUICallKit）使用免费试用版本的 SDK 调试通过后，上线购买正式版套餐需要重新更新 SDK 吗？

不需要。您只需要购买指定正式版套餐即可，SDK 相关配置均无需变更。

产品使用相关

音视频通话 SDK 都有哪些核心功能？

音视频通话 SDK 基于通话场景深度定制，提供了1v1语音和视频通话、群组通话、中途呼叫第三方、AI 通话降噪等功能，支持标清、高清、超高清等多种画质，提供微信视频聊天同款 UI 快速集成方案，配合低延时高质量的音视频服务，帮您快速搭建稳定可靠的音视频通话工具应用。

音视频通话 SDK 下载的 SDK 包含哪些内容，与 TUICallKit 的关系是什么？含 UI 吗？

音视频通话 SDK 下载的 SDK 即为 TUICallKit，包含 UI 层以及核心功能 SDK 两部分，提供微信同款 UI，核心功能 SDK 又称为 TUICallEngine，即音视频通话 SDK 就是 TUICallKit，包含了 TUICallEngine 与 UI。

音视频通话 SDK (TUICallKit)

UI层

TUICallEngine

音视频通话 SDK 是否包含了实时音视频 TRTC SDK 中全部的 API？

实时音视频（TRTC）是支撑音视频通话相关功能的腾讯云基础服务之一，主要负责提供基础的音视频通信能力，其包含诸如音视频采集、编码、渲染等等，音视频通话 SDK (1.2.0 及其以后版本) 提供有专门的 API 用来获取底层 TRTC 的关键能力。

如果我有自己的 UI，不需要音视频通话 SDK 的默认 UI 怎么办？

您仅需集成核心功能 SDK（TUICallEngine），详情请参见[界面定制](#)。

Android

最近更新时间：2024-11-29 09:13:12

TUICallKit 是否可以不引入 IM SDK，只使用 TRTC？

不可以，TUICallKit 全系组件都使用了腾讯云 IM SDK 作为通信的基础服务，例如通话拨打信令、通话忙线信令等核心逻辑，如果您已经购买有其他 IM 产品，也可以参照 TUICallKit 逻辑进行适配。

错误提示 “The package you purchased does not support this ability” ？

如遇以上错误提示，是由于您当前应用的音视频通话能力包过期或未开通，请参见 [开通服务](#)，领取或者开通音视频通话能力，进而继续使用 TUICallKit 组件。

如何购买音视频通话套餐？

请参考购买链接 [音视频通话 SDK 价格总览](#)，如有其他问题，请点击页面右侧，进行售前套餐咨询；也可以加入我们的 TUICallKit 技术交流平台 [zhiliao](#)，进行咨询和反馈。

TUICallKit 崩溃，崩溃日志：“No implementation found for xxxx”？

堆栈信息如下：

```
java.lang.UnsatisfiedLinkError: No implementation found for void
com.tencent.liteav.base.Log.nativeWriteLogToNative(int,
java.lang.String, java.lang.String) (tried
Java_com_tencent_liteav_base_Log_nativeWriteLogToNative and
Java_com_tencent_liteav_base_Log_nativeWriteLogToNative__ILjava_lang_Str
ing_2Ljava_lang_String_2)
    at com.tencent.liteav.base.Log.nativeWriteLogToNative (Native
Method)
    at com.tencent.liteav.base.Log.i (SourceFile:177)
    at com.tencent.liteav.basic.log.TXCLog.i (SourceFile:36)
    at
com.tencent.liteav.trtccalling.model.impl.base.TRTCLogger.i (TRTCLogger.j
ava:15)
    at
com.tencent.liteav.trtccalling.model.impl.ServiceInitializer.init (Servic
eInitializer.java:36)
    at
com.tencent.liteav.trtccalling.model.impl.ServiceInitializer.onCreate (Se
rviceInitializer.java:101)
    at
android.content.ContentProvider.attachInfo (ContentProvider.java:2097)
```

```
at
android.content.ContentProvider.attachInfo(ContentProvider.java:2070)
at
android.app.ActivityThread.installProvider(ActivityThread.java:8168)
at
android.app.ActivityThread.installContentProviders(ActivityThread.java:7709)
at
android.app.ActivityThread.handleBindApplication(ActivityThread.java:7573)
at
android.app.ActivityThread.access$2600(ActivityThread.java:260)
at
android.app.ActivityThread$H.handleMessage(ActivityThread.java:2435)
at android.os.Handler.dispatchMessage(Handler.java:110)
at android.os.Looper.loop(Looper.java:219)
at android.app.ActivityThread.main(ActivityThread.java:8668)
at java.lang.reflect.Method.invoke(Native Method)
at
com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:513)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1109)
```

⚠ 注意:

上述异常,是因为 TUICallKit 依赖的 TRTC 等 SDK 部分接口使用了 JNI 实现,Android Studio 在某些条件进行编译 APK 的时候可能并不会把 Native 的 .so 库打包进去,导致出现这类报错,重新 Clean 一下工程,卸载重装即可。

allowBackup 异常?

● 报错信息:

```
Manifest merger failed : Attribute application@allowBackup value=(true) from
AndroidManifest.xml
```

- **问题原因:** TUICallKit 依赖的 IM SDK 中默认 allowBackup 的值为 false,表示关闭应用的备份和恢复功能。
- **解决方法:** 您可以在您工程的 AndroidManifest.xml 文件中删除 allowBackup 属性或将该属性改为 false,表示关闭备份和恢复功能;也可以在 AndroidManifest.xml 文件的 application 节点中添加 tools:replace="android:allowBackup"; 表示覆盖 IM SDK 的设置,使用您自己的设置。

TUICallKit 是否支持离线推送呢?

支持,Android 离线推送采用腾讯云的TIMPush推送插件以及自集成推送,接入方式详见: [离线唤醒](#)。

在通话邀请超时时间内，被邀请者如果离线再上线，能否弹出通话界面？

单人通话时，如果在超时时间内上线，会触发来电邀请；群组通话，如果在超时时间内上线，会拉起未处理的20条群消息，如果存在通话邀请，则触发来电邀请。在不同的版本上来电的显示策略有所不同（详见下方：[被叫端来电显示策略](#)）。

应用在后台时，不能自动将通话界面拉取到前台

1. TUICallKit 2.3 及以上版本（[Android&iOS发布日志](#)）调整了被叫端的来电显示策略，见下方：[被叫端来电显示策略](#)。

2. TUICallKit 2.3 版本之前，将应用从后台自动拉取到前台，需要检查 App 是否开启了“后台自启动”或“悬浮窗”权限。

不同厂商、甚至同一厂商不同 Android 版本，其对于应用开放的权限以及权限名称也会存在不一致。例如，小米6只需要开启后台弹出界面权限，而红米需要同时打开后台弹出界面和显示悬浮窗权限。如何开启权限，详见下方：[相关权限开启](#)。

3. 如果遇到以下场景拉不起通话界面，原因是：应用的启动堆栈变化，导致 CallKitActivity 界面被遮挡移除了。

场景一：接通过后退到后台，点击桌面图标进入应用，原通话界面消失；

场景二：应用在后台时，开启banner的情况下，收到通知 View，不点击 View，点击桌面图标进入应用，拉不起通话界面且通知消失；

场景三：应用在后台时，收到离线推送，不点击推送通知，点击桌面图标进入应用，无法拉起通话界面或通话界面闪了一下；

以上几种情况，需要在您自己业务的默认启动的主 Activity 中添加以下代码，每个应用默认启动的 Activity 都不一样，详见 AndroidManifest.xml 配置，以目前大部分应用的启动页 SplashActivity 为例：

添加代码

```
if (!isTaskRoot() && getIntent() != null &&
    getIntent().hasCategory(Intent.CATEGORY_LAUNCHER)
    && Intent.ACTION_MAIN.equals(getIntent().getAction())) {
    finish();
    return;
}
```

具体添加位置

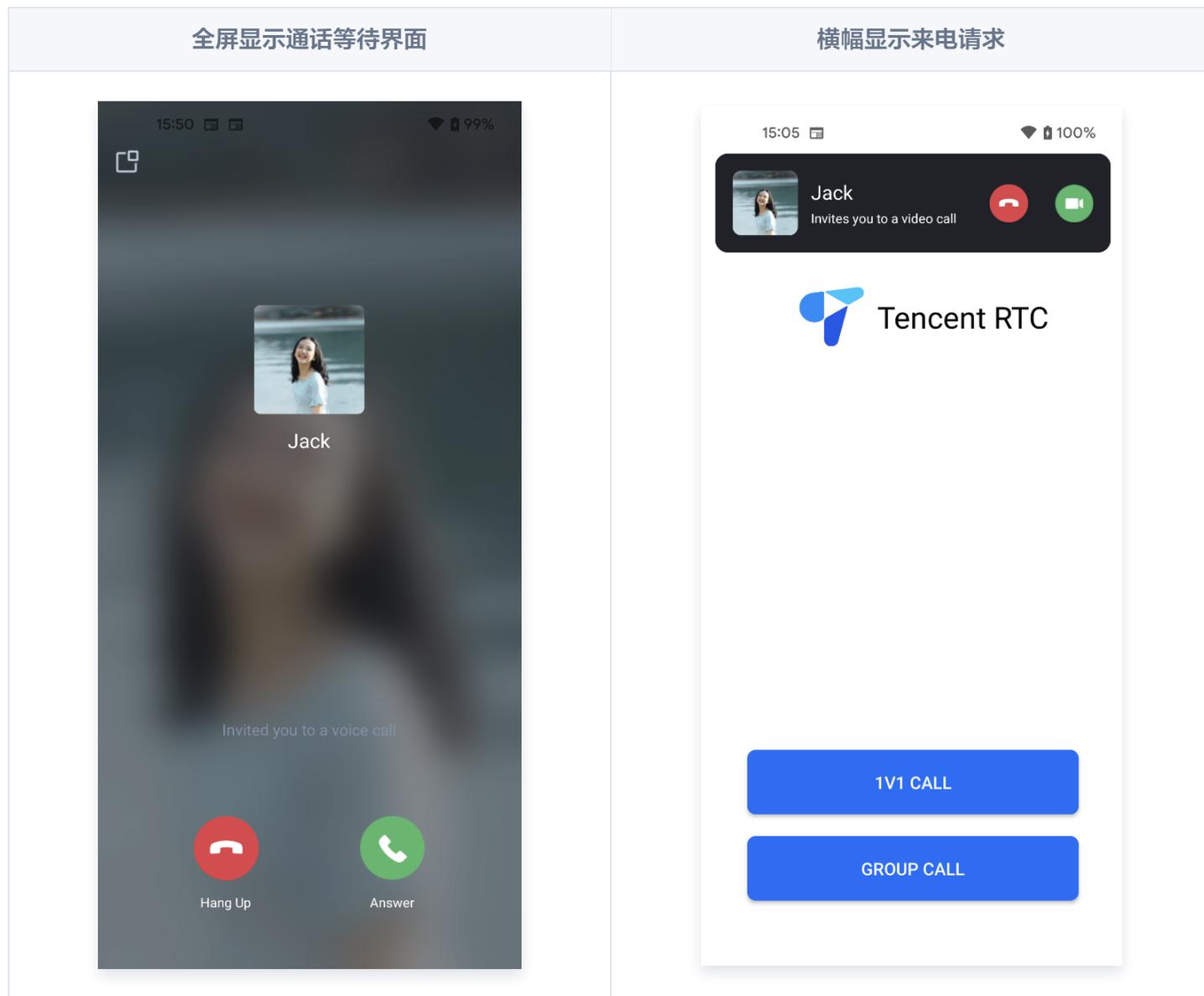
```
SplashActivity.java x
25
26 <> public class SplashActivity extends BaseLightActivity {
27     private static final String TAG = SplashActivity.class.getSimpleName();
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         if (!isTaskRoot() && getIntent() != null && getIntent().hasCategory(Intent.CATEGORY_LAUNCHER) && Intent.ACTION_MAIN.equals(getIntent().getAction())) {
33             finish();
34             return;
35         }
36         setContentView(R.layout.activity_splash);
37
38         getWindow().addFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_NAVIGATION);
39
40         if (Build.VERSION.SDK_INT >= 21) {
41             View decorView = getWindow().getDecorView();
42             decorView.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN | View.SYSTEM_UI_FLAG_LAYOUT_STABLE);
43             getWindow().setStatusBarColor(Color.TRANSPARENT);
44             getWindow().setNavigationBarColor(Color.TRANSPARENT);
45         }
46
SplashActivity > onCreate()
AndroidManifest.xml x
10     android:supportsRtl="true"
11     android:theme="@style/DemoAppTheme">
12
13     <activity
14         android:name="com.tencent.qcloud.tim.demo.SplashActivity"
15         android:screenOrientation="portrait"
16         android:exported="true"
17         android:theme="@style/SplashTheme">
18         <intent-filter>
19             <action android:name="android.intent.action.VIEW" />
20             <action android:name="android.intent.action.MAIN" />
21             <action android:name="com.tencent.qcloud.splash" />
22
23             <category android:name="android.intent.category.DEFAULT" />
24             <category android:name="android.intent.category.LAUNCHER" />
25         </intent-filter>
26     </activity>
27 </application>
28 </manifest>
```

❗ 说明:

如果您在开发中开启了所有权限或做了上述尝试，依然无法自动拉起通话界面到前台，请加入 TUICallKit 技术交流平台 [zhiliao](#)，联系我们协助处理。

被叫端来电显示策略

为使 TUICallKit 适应不同的业务需求，增加产品特色，提升用户的使用体验，TUICallKit 2.3 及以上版本（[Android&iOS发布日志](#)），优化收到来电后的通话页面显示弹出策略，详情如下所示：



2.3.0.920版本及之后的版本

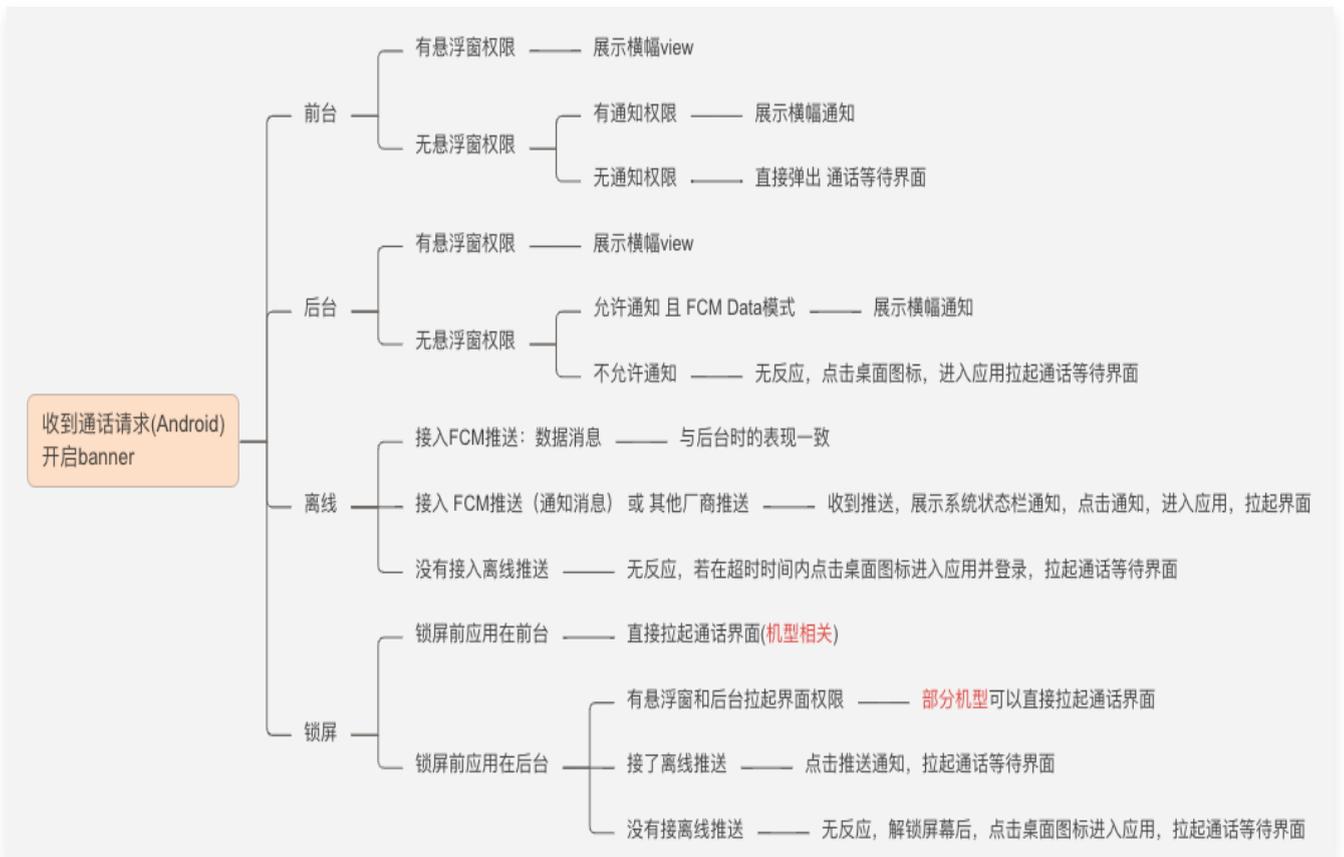
1. 如果您想要被叫端收到邀请时，尽量去拉起全屏通话界面，那么您可以更新 [tuicallkit-kt](#) 代码到最新。该情况下，被叫端的来电显示策略如下：



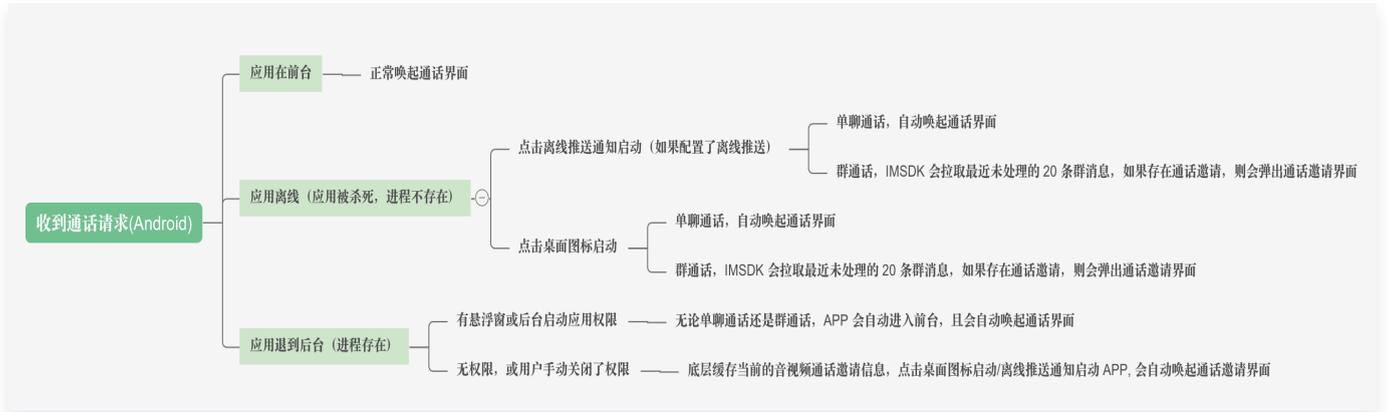
2. 如果您想在被叫端收到通话的时候，先展示一个横幅，然后根据需要拉起全屏通话界面，那么您可以调用以下接口开启该功能。

```
TUICallKit.createInstance(context).enableIncomingBanner(true);
```

开启后，被叫端的来电策略如下所示，只要开启**悬浮窗权限**，就能尽可能的展示来电横幅。



2.3版本之前



说明:

相关权限如何开启, 详见下方: [相关权限开启](#)。

如果没有按照上述策略显示来电界面, 请过滤 `onCallReceived` 日志, 检查是否收到通话邀请, 如果没有该日志的打印, 请加入我们的 TUICallKit 技术交流平台 [zhiliao](#), 联系我们协助处理。

相关权限开启

为实现良好的通话体验, 建议您在应用中开启“通知”权限、“显示在其他应用上层(悬浮窗)”以及“后台拉起界面”权限, 具体方法如下:

代码指引

- **通知权限:** 便于展示推送通知: 请参见 [通知运行时权限](#) 和 [请求运行时权限](#) 根据业务需求自行实现。
- **悬浮窗权限:** 用于展示自定义的来电通知, 以及通话悬浮窗。
- **后台拉起界面权限:** 用于当应用在后台时拉起界面 (例如: VIVO手机)。

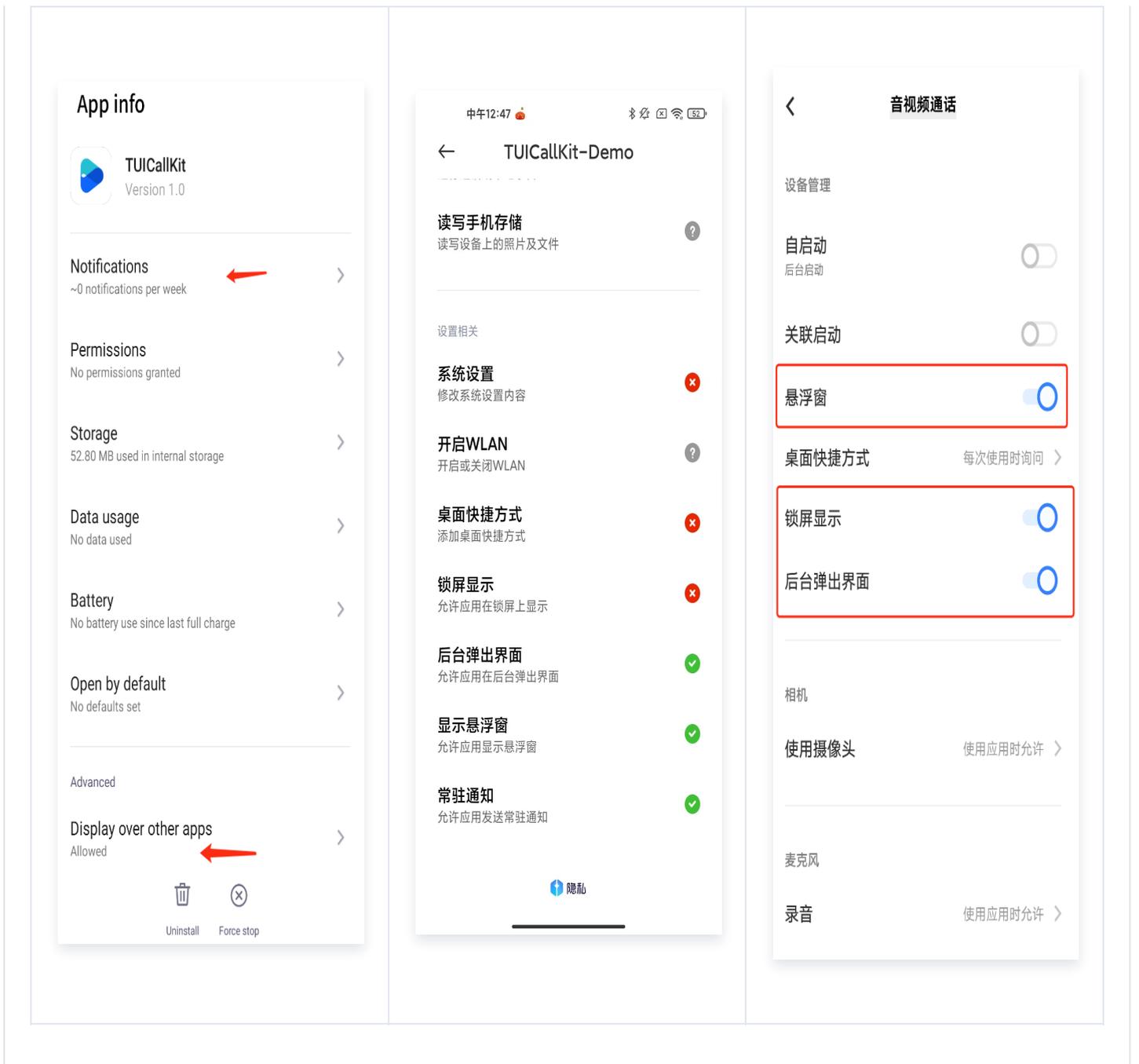
```
fun requestPermission(context: Context?) {
    //In TUICallKit, Please open both OverlayWindows and Background
    pop-ups permission.
    PermissionRequester.newInstance(
```

```
PermissionRequester.FLOAT_PERMISSION,  
PermissionRequester.BG_START_PERMISSION)  
    .request()  
}
```

手动开启

安装应用后，您可以长按应用图标，选择“应用信息”，然后开启“通知”权限、“显示在其他应用上层”以及“后台拉起界面”权限。或者您可以到 [手机 > 系统设置 > 应用管理 > 应用](#) 中手动开启上述权限。

Pixel 4a	小米	vivo
----------	----	------



iOS

最近更新时间：2025-05-22 11:24:22

错误提示 “The package you purchased does not support this ability” ？

如遇以上错误提示，是由于您当前应用的音视频通话能力包过期或未开通，请参见 [开通服务](#)，领取或者开通音视频通话能力，进而继续使用TUICallKit组件。

如何购买音视频通话套餐？

请参见购买链接 [开通正式版](#)，如有其他问题，请点击页面右侧，进行售前套餐咨询；也可以加入我们的TUICallKit 技术交流平台 [zhiliao](#)，进行咨询和反馈。

如何修改 TUICallKit 源码？

使用 CocoaPods 导入组件，具体步骤如下：

1. 在您的工程 Podfile 文件同一级目录下创建 TUICallKit 文件夹。
2. 单击进入 [Github/TUICallKit](#)，选择克隆/下载代码，然后将 iOS 目录下的 TUICallKit-Swift 文件夹和 TUICallKit-Swift.podspec 文件拷贝到您在 [步骤1](#) 创建的 TUICallKit 文件夹下。
3. 在您的 Podfile 文件中添加以下依赖。

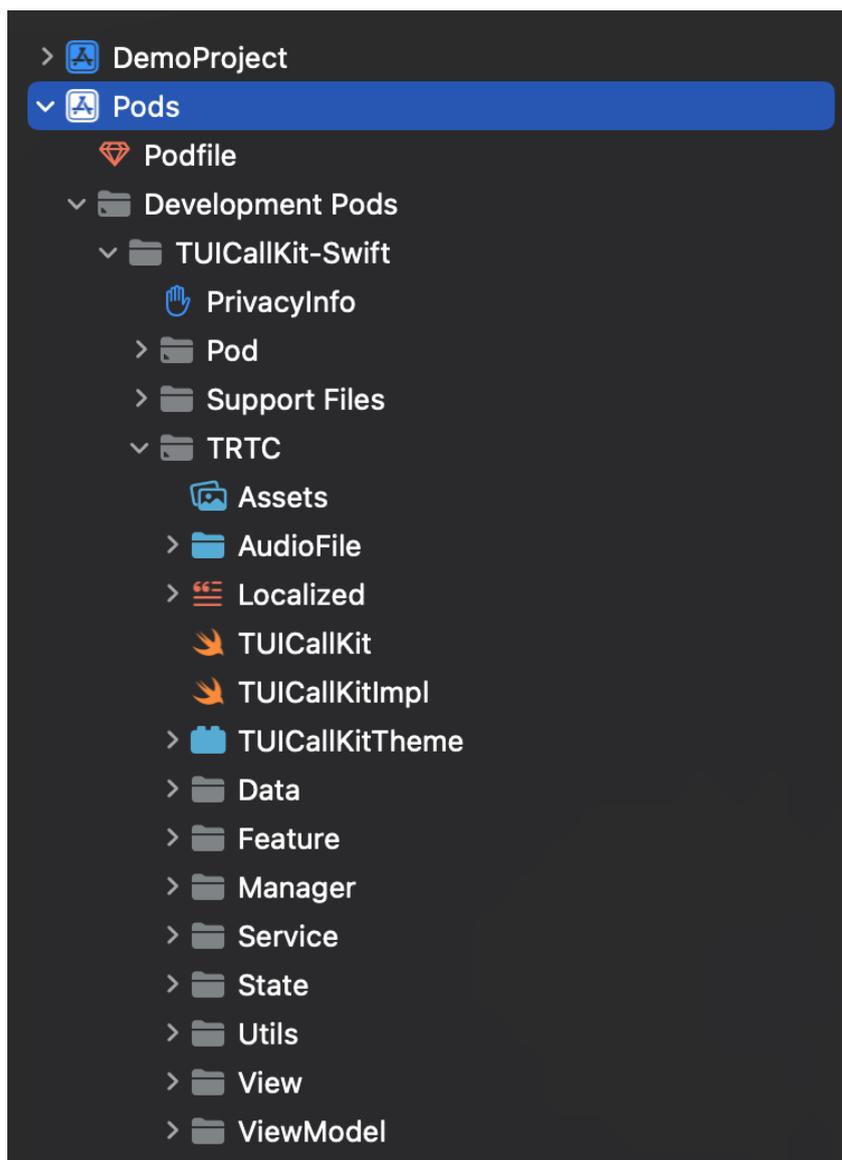
```
# :path => "指向 TUICallKit-Swift.podspec 的相对路径"  
pod 'TUICallKit-Swift', :path => "TUICallKit/TUICallKit-Swift.podspec"
```

4. 执行 `pod install` 命令，完成导入。

⚠ 注意

TUICallKit-Swift 文件夹和 TUICallKit-Swift.podspec 文件必须在同一目录下。

TUICallKit_Swift 组件集成后效果：

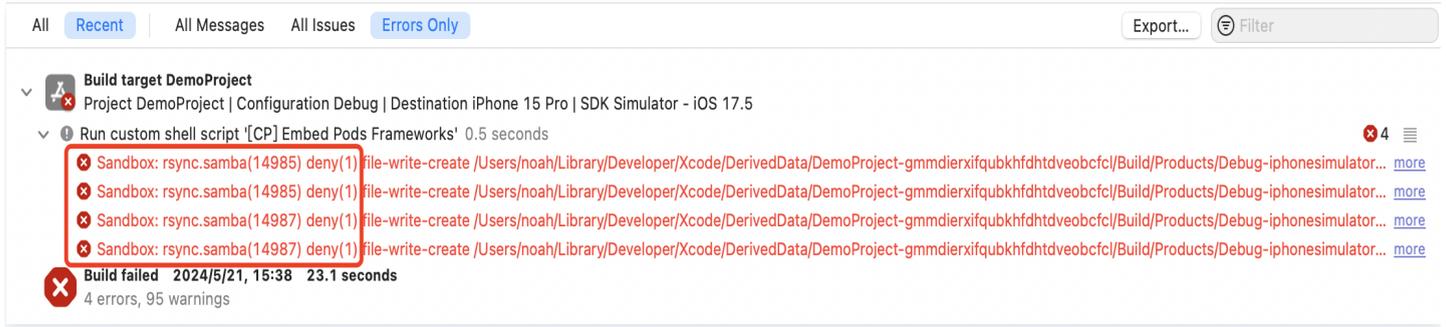


! 说明

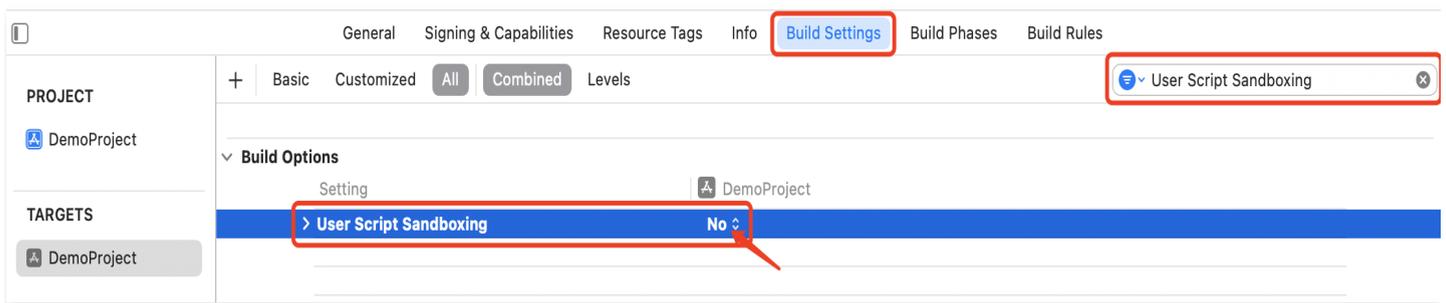
TUICallKit_Swift 组件集成后支持文件夹分层显示，方便您阅读和修改源代码。

Xcode 15 编译报错?

1、出现 Sandbox: rsync 编译报错



可以在 **Build Settings** 中把 **User Script Sandboxing** 设置为 **NO**:



2、出现 SDK does not contain 编译报错



可以在 Podfile 添加如下代码:

```

# target 'xxxx' do
#   ...
#   pod 'TUICallKit_Swift'
# end

post_install do |installer|
  installer.pods_project.targets.each do |target|
    target.build_configurations.each do |config|
      config.build_settings['IPHONEOS_DEPLOYMENT_TARGET'] = '13.0'
    end
  end
end
    
```

```
end
```

3、如果在 M 系列电脑上运行模拟器，可能会出现 Linker command failed with exit code 1 (use -v to see invocation) 编译报错



The screenshot shows the Xcode console with the following content:

```
Users/yuxiwei/Library/Developer/Xcode/DerivedData/DemoProject-bjpnmbwrcvinmgfihijgffeqoc/Build/Intermediates.noindex/Pods.build/Debug-iphonesimulator/TUICore.build/Objects-normal/x86_64/TUICore_dependency_info.dat -o /Users/yuxiwei/Library/Developer/Xcode/DerivedData/DemoProject-bjpnmbwrcvinmgfihijgffeqoc/Build/Products/Debug-iphonesimulator/TUICore/TUICore.framework/TUICore
```

```
ld: warning: ignoring file '/Users/yuxiwei/Library/Developer/Xcode/DerivedData/DemoProject-bjpnmbwrcvinmgfihijgffeqoc/Build/Products/Debug-iphonesimulator/SDWebImage/SDWebImage.framework/SDWebImage': found architecture 'arm64', required architecture 'x86_64'
```

```
Undefined symbols for architecture x86_64:  
  "_OBJC_CLASS_$_SDImageCoderHelper", referenced from:  
      in TUITool.o  
ld: symbol(s) not found for architecture x86_64  
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

Below the console output, there are three error messages:

- Ignoring file '/Users/yuxiwei/Library/Developer/Xcode/DerivedData/DemoProject-bjpnmbwrcvinmgfihijgffeqoc/Build/Products/Debug-iphonesimulator/SDWebImage/SDWebImage.fra... [more](#)
- Undefined symbol: _OBJC_CLASS_\$_SDImageCoderHelper
- Linker command failed with exit code 1 (use -v to see invocation)

可以在 Podfile 添加如下代码：

```
# target 'xxxx' do  
# ...  
# pod 'TUICallKit_Swift'  
# end  
  
post_install do |installer|  
  installer.pods_project.targets.each do |target|  
    target.build_configurations.each do |config|  
      config.build_settings['EXCLUDED_ARCHS[sdk=iphonesimulator*]'] =  
"arm64"  
    end  
  end  
end
```

TUICallKit 和自己集成的音视频库冲突了？

腾讯云的 音视频库 不能同时集成，可能存在符号冲突，可以按照下面的场景处理。

1. 如果您使用了 `TXLiteAVSDK_TRTC` 库，不会发生符号冲突。可直接在 `Podfile` 文件中添加依赖，

```
pod 'TUICallKit_Swift'
```

2. 如果您使用了 `TXLiteAVSDK_Professional` 库，会产生符号冲突。您可在 `Podfile` 文件中添加依赖，

```
pod 'TUICallKit_Swift/Professional'
```

3. 如果您使用了 `TXLiteAVSDK_Enterprise` 库，会产生符号冲突。建议升级到 `TXLiteAVSDK_Professional` 后使用 `TUICallKit_Swift /Professional`。

TUICallKit 是否可以不引入 IM SDK，只使用 TRTC？

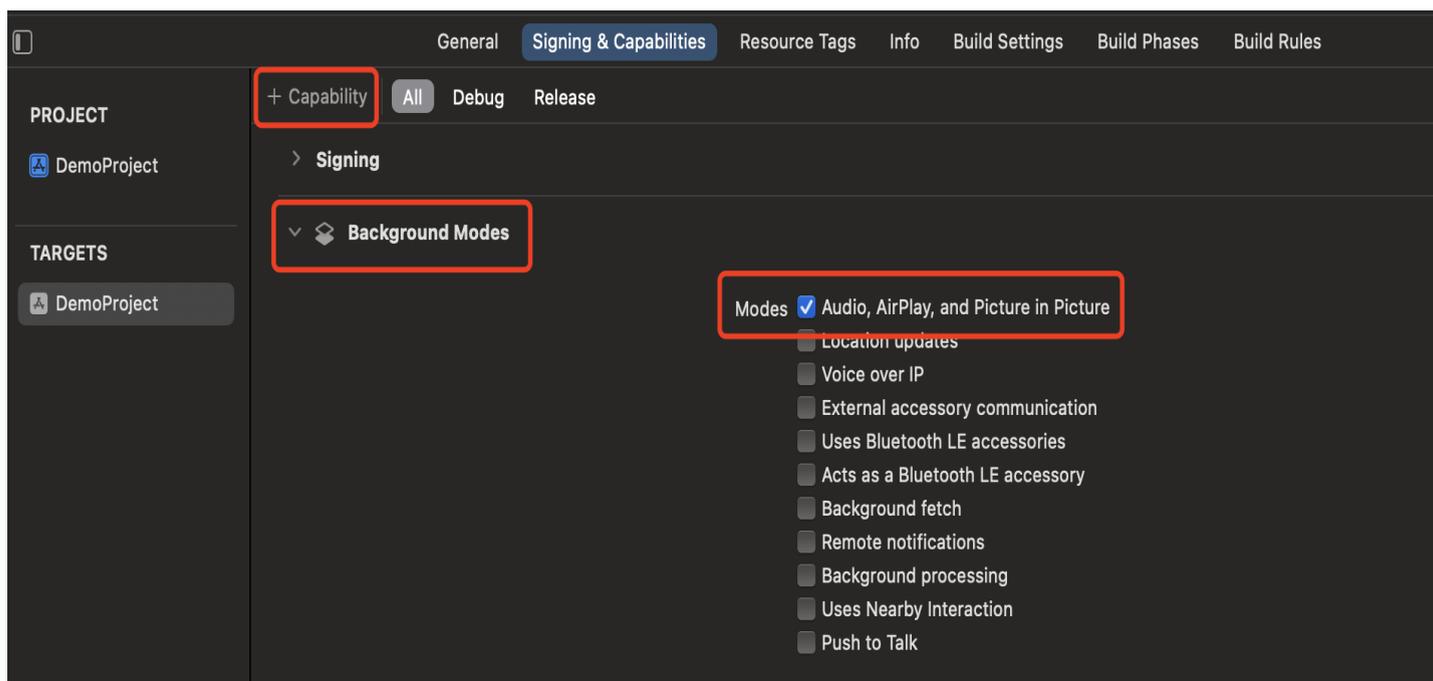
不可以。`TUICallKit` 全系组件都使用了腾讯云 IM SDK 作为通信的基础服务，比如通话拨打信令、通话忙线信令等核心逻辑，如果您已经购买有其他 IM 产品，也可以参照 `TUICallKit` 逻辑进行适配。

TUICallKit 组件支持自定义铃声吗？

支持，调用 `setCallingBell` 即可。

TUICallKit 是否支持后台运行？

支持，如需要进入后台仍然运行相关功能，可选中当前工程项目，在 **Capabilities** 下的 **Background Modes** 模块中勾选 **Audio, AirPlay and Picture in Picture**，如下图所示：



如何查看 TRTC 日志？

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

iOS: `sandbox的Documents/log`

说明:

- 查看 .xlog 文件需要下载 [解密工具](#)，在 Python 2.7环境中放到 xlog 文件同目录下直接使用 `python decode_mars_log_file.py` 运行即可。
- 查看 .clog 文件（9.6 版本以后新的日志格式）需要下载 [解密工具](#)，在 Python 2.7 环境中放到 clog 文件同目录下直接使用 `python decompress_clog.py` 运行即可。

Web

最近更新时间：2024-11-29 09:13:12

说明：

- 如果并没有解决您的问题，也欢迎您加入我们的 TUICallKit 技术交流 QQ 群：**605115878** 来进行交流讨论。
- 我们团队即将在 Web 端丰富更多不同的组件，以满足您的开发需求。
希望您能抽出几分钟时间，为您自己的需求投上一票，我们将优先开发！
问卷地址：[TUICallKit Web 问卷调查](#)

一、基础环境问题

说明：

以下问题对于 TUICallKit 与 TUICallEngine SDK 同样适用。

TUICallEngine 和 TUICallKit 分别是什么？

TUICallKit 是含 UI 音视频通话组件，底层是用 TUICallEngine SDK，目前支持 Typescript+Vue2 / Typescript+Vue3，可直接将组件放到页面中，调用简单的接口即可直接实现音视频通话，开源地址为：[TUICallKit/Web](#)。

TUICallEngine SDK 是音视频通话组件的无 UI SDK，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。npm 地址为 [tuicall-engine-webrtc](#)。

Web 端支持哪些浏览器？报错“获取设备权限失败”？

请先确保页面已被授权使用麦克风或摄像头，参见 [设备授权说明](#)。

对浏览器的详细支持度，请参见 [浏览器兼容信息](#)。

对于上述没有列出的环境，您可以在需要检测的浏览器打开 [能力测试](#) 测试是否完整的支持 WebRTC 的功能。

为什么本地开发测试能正常使用，但是部署到线上用 IP 访问后无法正常视频/语音通话？

对网站域名协议的要求

出于对用户安全、隐私等问题的考虑，浏览器限制网页在 HTTPS 协议下才能正常使用本文档中所对接组件的全部功能。为确保生产环境中的用户能够顺畅体验产品功能，请将您的网站部署在 `https://` 协议的域名下。更多请参见 [页面访问协议说明](#)。

对网络环境的要求

在使用 TUICallKit 时，用户可能因防火墙限制导致无法正常进行音视频通话，请参考 [应对防火墙策略](#) 将相应端口及域名添加至防火墙白名单中。

在接通过程中报: "is not included in the current tim's package"?

TUICallKit (含 TUICallEngine SDK) 依赖的 `tim-js-sdk` 版本需要 $\geq 2.21.2$ 。

如果 `tim-js-sdk` 依赖包版本正确, 则可能是 SDKAppID 未购买音视频套餐或套餐包不支持所调用的功能, 请访问 [音视频通话功能出现了未开通的错误提示](#)。

TUICallKit (含 TUICallEngine) 是否支持接收离线消息?

不支持接收离线消息。

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

TUICallEngine init 未完成, 需要在 init 完成后使用此API

未调用 `login` 接口, 所有功能需要先进行登录完成后才能使用, 具体参考 [TUICallEngine login](#)。

⚠ 注意:

`TUICallEngine login` 是一个异步接口, 避免在 `login` 还未完成, 直接调用 `TUICallEngine` 的接口。

当前通话状态无法使用该 API

API 与通话状态对照表:

API	idle	calling	connected
<code>call</code>	✓	×	×
<code>groupCall</code>	✓	×	×
<code>accept</code>	×	✓	×
<code>reject</code>	×	✓	×
<code>hangup</code>	×	✓	✓
<code>inviteUser</code>	×	✓	✓
<code>joinInGroupCall</code>	✓	×	×

二、TUICallKit 问题 (含 UI)

1. TUICallKit Web 支持什么框架? 支持 H5 吗?

TUICallKit 适用于 `Vue2.7 + Typescript` 或者 `Vue3 + Typescript` 项目, 若您采用其他语言或者技术栈, 请访问 [界面定制指引](#)。

TUICallKit 支持 H5, 通过页面 UA 自动修改适应移动端的布局, 为了移动端的良好体验, 推荐通过修改 CSS 在 H5 时将 `<TUICallKit/>` 组件放大至全屏。

2. TUICallKit 打包失败?

对于 Vite 项目，您需要在 `vite.config.js` 中添加 `base: "./"`。

对于 Vue-CLI 创建的 webpack 项目，您需要在 `vue.config.js` 中添加 `publicPath: "./"`。

3. TUICallKit 报错“获取设备权限失败”?

请先确保页面已被授权使用麦克风或摄像头，参见 [设备授权说明](#)。

可以尝试 [官方 Demo](#) 是否可以正常通话，然后在 [设备检测页面](#) 检查是否支持 webrtc。如果还有问题，请联系我们 QQ群：646165204。

4. timeout 字段设置无效原因?

该字段目前在 call / groupcall API 中会被使用到。

目前全平台 TUICallKit 的策略是被叫登录后，仅拉取 30s 内的历史消息。所以被叫登录后无法拉到主叫 30s 前的呼叫信息，从而导致被叫无法拉起邀请页面进行通话。

5. 视频通话切换大小屏出现闪屏现象?

ios 17 设备和 WebRTC 底层兼容性问题，等待后续修复。

6. safari 浏览器省电模式出现暂停键?

浏览器层面针对省电模式的视频播放策略，SDK 无法处理。

7.如何生成 UserSig?

UserSig 是腾讯云为其云服务设计的一种安全保护签名，是一种登录凭证，由 SDKAppID 与 SecretKey 等信息组合加密得到。

- 方式一：控制台获取，参考 [获取临时 userSig](#)。
- 方式二：部署临时生成脚本。

警告：

此方式是在前端代码中配置 SecretKey，该方法中 SecretKey 很容易被反编译逆向破解，一旦您的密钥泄露，攻击者就可以盗用您的腾讯云流量，因此该方法仅适合本地跑通功能调试，生产环境请看方式三。

为方便初期调试，userSig 可临时使用 `GenerateTestUserSig-es.js` 中 `genTestUserSig(params)` 函数来计算，例如：

```
import { genTestUserSig } from "@tencentcloud/call-uikit-vue/debug/GenerateTestUserSig-es.js";
const { userSig } = genTestUserSig({ userID: "Alice", SDKAppID: 0, SecretKey: "YOUT_SECRETKEY" });
```

- **方式三：正式环境使用。**

正确的 UserSig 签发方式是将 UserSig 的计算代码集成到您的服务端，并提供面向项目的接口，在需要 UserSig 时由您的项目向业务服务器发起请求获取动态 UserSig。更多详情请参见 [服务端生成 UserSig](#)。

8.如何创建 userID?

- 通过 userID 与 UserSig 登录过一次，会默认创建该用户。
- 通过 [即时通信 IM 控制台](#) 进行创建和获取，单击目标应用卡片，进入应用的账号管理页面，也可创建账号并获取 userID。



9. 解决源码拷贝可能导致的报错

如果您在使 TUICallKit 组件时遇到了报错，请不要担心，大多数情况下这是由于 ESLint 和 TSConfig 配置不一致造成的。您可以查阅文档，按照要求正确配置即可。如果您需要帮助，请随时联系我们，我们将确保您能够成功地使用此组件。以下是几个常见的问题：

ESLint 报错

若 TUICallKit 与您项目的代码风格不一致导致报错，可将本组件目录屏蔽，如在项目根目录增加

`.eslintignore` 文件，如：

```
# .eslintignore
src/components/TUICallKit
```

TypeScript 报错

如遇 `Cannot find module '../package.json'` 报错，是因为 TUICallKit 内引用了 JSON 文件，可在 `tsconfig.json` 中添加相关配置。

其他 TSConfig 问题请参见 [TSConfig Reference](#)。

```
{
  "compilerOptions": {
    "resolveJsonModule": true
  }
}
```

10. 如何源码集成 TUICallKit?

步骤一：下载 TUICallKit 组件源码及相关插件

Vue3

```
npm install @tencentcloud/call-uikit-vue
```

Vue2.7

```
npm install @tencentcloud/call-uikit-vue2
```

Vue2.6

```
npm install @tencentcloud/call-uikit-vue2.6 @vue/composition-api
```

源码集成需要安装 [unplugin-vue2-script-setup](#) 插件。

```
npm i -D unplugin-vue2-script-setup
```

修改构建工具配置文件。

Vue Cli

```
// vue.config.js
const ScriptSetup = require('unplugin-vue2-script-setup/webpack').default

module.exports = {
  parallel: false, // disable thread-loader, which is not
compactible with this plugin
  configureWebpack: {
    plugins: [
      ScriptSetup({ /* options */ }),
    ],
  },
}
```

vite

```
// vite.config.ts
import { defineConfig } from 'vite'
import { createVuePlugin as Vue2 } from 'vite-plugin-vue2'
import ScriptSetup from 'unplugin-vue2-script-setup/vite'

export default defineConfig({
  plugins: [
    Vue2(),
    ScriptSetup({ /* options */ }),
  ],
})
```

Webpack

```
// webpack.config.js
const ScriptSetup = require('unplugin-vue2-script-setup/webpack').default

module.exports = {
  /* ... */
  plugins: [
```

```
ScriptSetup({ /* options */ }),  
]  
}
```

步骤二：源码集成 TUICallKit 组件包

将源码拷贝到自己的项目中，以拷贝到 `src/components/` 目录为例：

macOS + Vue3

```
mkdir -p ./src/components/TUICallKit && cp -r  
./node_modules/@tencentcloud/call-uikit-vue/*  
./src/components/TUICallKit
```

macOS + Vue2

```
mkdir -p ./src/components/TUICallKit && cp -r  
./node_modules/@tencentcloud/call-uikit-vue2/*  
./src/components/TUICallKit
```

Windows + Vue3

```
xcopy .\node_modules\@tencentcloud\call-uikit-vue  
.\src\components\TUICallKit /i /e
```

Windows + Vue2

```
xcopy .\node_modules\@tencentcloud\call-uikit-vue2  
.\src\components\TUICallKit /i /e
```

步骤三：修改 TUICallKit 引入路径

修改 TUICallKit 引入路径为本地文件引入。

⚠ 注意：

此方法可能与您的 ESLint、TypeScript 配置冲突，若出现报错，可参考 [解决源码拷贝导致的报错](#)。

```
import { TUICallKit, TUICallKitServer } from
"./components/TUICallKit/src/index";
```

接下来请参考上面的步骤及说明，使用 TUICallKit 组件提供的功能。

三、套餐问题

1. 错误提示 “The package you purchased does not support this ability” ？

由于您当前应用的音视频通话能力包过期或未开通，请参见 [开通服务](#)，领取或者开通音视频通话能力，进而继续使用 TUICallKit 组件。

2. 如何购买套餐？

请参考购买链接 [音视频通话 SDK 价格总览](#)，如有其他问题，请点击页面右侧，进行售前套餐咨询，或 [进入 IM 社群](#) 进行咨询和反馈。

四、内网代理

内网的环境下如何使用 callkit？

1. 通过 getTRTCCloudInstance 获取 TRTCCloud 实例。

注意：
v3.1.3+ 支持。

```
const trtcCloud =
TUICallKitServer.getTUICallEngineInstance().getTRTCCloudInstance();
```

2. 通过 TRTCCloud 实例调用 callExperimentalAPI 完成代理服务器的设置（注意：需要在通话前调用）。具体如下：

```
trtcCloud.callExperimentalAPI(JSON.stringify({
  api: 'setNetworkProxy',
  params: {
    websocketProxy: 'wss://proxy.example.com/ws/',
    turnServer: [{
      url: '14.3.3.3:3478',
      username: 'turn',
      credential: 'turn',
    }],
    iceTransportPolicy: 'relay',
```

```
},  
});
```

⚠ 注意:

TRTC 应对防火墙受限参见 [设置代理服务器](#)。

五、其它

1. 如何关闭美颜?

- web 端默认无美颜。
- 小程序端调用 [setBeautyLevel API](#)，传入参数为0时，会关闭默认美颜。

小程序

最近更新时间：2024-08-01 11:09:41

📌 说明：

如果并没有解决您的问题，也欢迎您加入我们的 TUICallKit 技术交流 QQ 群：**605115878**，来进行交流讨论。

1. 是否支持 uniapp、Taro 等开发环境？

Taro 暂不支持，uniapp 请参见 [快速集成方案（uni-app）](#)。

2. 如何全局监听，在所有页面都可以唤起通话界面？

⚠️ 注意：

TUICallKit 默认集成全局监听。

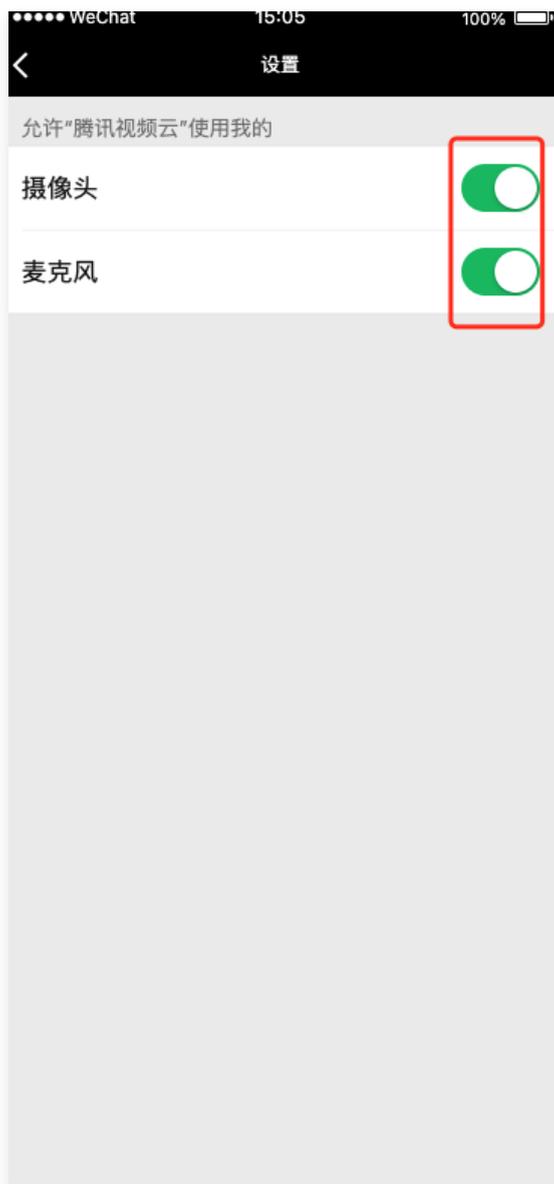
3. 当摄像头/麦克风权限被拒绝，如何处理？

首次进入小程序需要默认是没有设备权限的，需要手动开启权限。

1. 打开小程序的设置。



2. 开启摄像头/麦克风权限。



4. 报错 "authorize:fail:appid privacy api banned", 如何处理?

由于 TRTC 涉及到麦克风与摄像头的权限问题，需要用户在小程序控制台中更新隐私保护政策，具体步骤如下：

1. 登录腾讯平台的小程序后台，点击设置栏目。



2. 选择用户隐私保护指引设置功能，点击更新。

服务内容声明			
用户隐私保护指引	已更新	基于微信提供的标准化用户隐私保护指引，根据小程序实际情况更新并展示给用户。了解详情	更新
用户生成内容场景声明	未声明	对小程序是否包含用户生成内容（UGC）进行使用场景说明	更新
小程序订单中心path设置	待设置	将小程序的订单页对应的path进行设置。了解详情	设置

3. 在弹出的表格中，勾选小程序会涉及到的类型。

使用用户信息类型 ✕

在《小程序用户隐私保护指引内容介绍》中可查阅用户信息与使用接口对应关系

可补充选择

<input type="checkbox"/> 用户信息（微信昵...	<input type="checkbox"/> 地址	<input type="checkbox"/> 发票信息
<input type="checkbox"/> 微信运动数据	<input checked="" type="checkbox"/> 麦克风	<input type="checkbox"/> 选中的照片或视频...
<input checked="" type="checkbox"/> 摄像头	<input type="checkbox"/> 手机号	<input type="checkbox"/> 通讯录（仅写入）...
<input type="checkbox"/> 设备信息	<input type="checkbox"/> 身份证号码	<input type="checkbox"/> 订单信息
<input type="checkbox"/> 发布内容	<input type="checkbox"/> 所关注账号	<input type="checkbox"/> 操作日志
<input type="checkbox"/> 相册（仅写入）权限	<input type="checkbox"/> 车牌号	<input type="checkbox"/> 蓝牙
<input type="checkbox"/> 日历（仅写入）权限	<input type="checkbox"/> 邮箱	<input type="checkbox"/> 选中的文件
<input type="checkbox"/> 选择的位置信息	<input type="checkbox"/> 加速传感器	<input type="checkbox"/> 磁场传感器
<input type="checkbox"/> 方向传感器	<input type="checkbox"/> 陀螺仪传感器	<input type="checkbox"/> 剪切板
<input type="checkbox"/> 自定义		

取消
确认

5. 出现推流黑屏，样式错乱怎么回事？

警告：

微信小程序 Skyline 渲染引擎与 live-pusher 存在兼容性问题，如果您使用微信开发者工具 1.06.2307260+ 版本以上的新创建的项目，请在 app.json 中删除 Skyline 的相关配置。

```
"renderer": "skyline",
"rendererOptions": {
  "skyline": {
    "defaultDisplayBlock": true,
    "disableABTest": true,
    "sdkVersionBegin": "3.0.0",
    "sdkVersionEnd": "15.255.255"
  }
},
"componentFramework": "glass-easel",
"sitemapLocation": "sitemap.json",
"lazyCodeLoading": "requiredComponents"
```

6. 什么是 SDKAppID、SecretKey？

- SDKAppID: IM 的应用 ID，用于业务隔离，即不同的 SDKAppID 的通话彼此不能互通。
- Secretkey: IM 的应用密钥，需要和 SDKAppID 配对使用，用于签出合法使用 IM 服务的鉴权用票据 UserSig。

7. 什么是 UserSig，如何生成 UserSig？

- UserSig 是用户登录即时通信 IM 的密码，其本质是对 UserID 等信息加密后得到的密文。
- UserSig 签发方式是将 UserSig 的计算代码集成到您的服务端，并提供面向项目的接口，在需要 UserSig 时由您的项目向业务服务器发起请求获取动态 UserSig。更多详情请参见 [服务端生成 UserSig](#)。

8. 错误提示 “The package you purchased does not support this ability”，如何处理？

如遇以上错误提示，是由于您当前应用的音视频通话能力包过期或未开通，领取或者开通音视频通话能力，进而继续使用 TUICallKit 组件。

注意：

单击免费体验以后，部分之前使用过 [实时音视频 TRTC](#) 服务的用户会提示：

```
[ -100013]:TRTC service is suspended. Please check if the package balance is 0 or the Tencent Cloud account is in arrears
```

当实时音视频 TRTC 的免费额度（10000分钟）已经过期或者耗尽，就会导致开通此项服务失败，这里您可以单击 [TRTC 控制台](#)，找到对应 SDKAppID 的应用管理页，开通后付费功能后，再次启用应用即可正常体验音视频通话能力。具体参见下图：



9. 微信开发者工具有路径报错，如何处理？

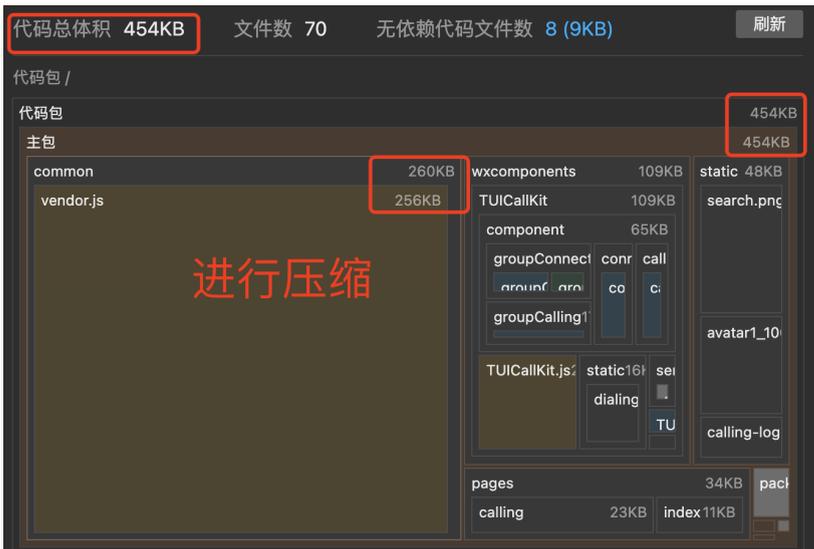
如果微信开发者工具有路径保存，请清除缓存，避免开发者工具的缓存造成渲染异常。具体参见下图：



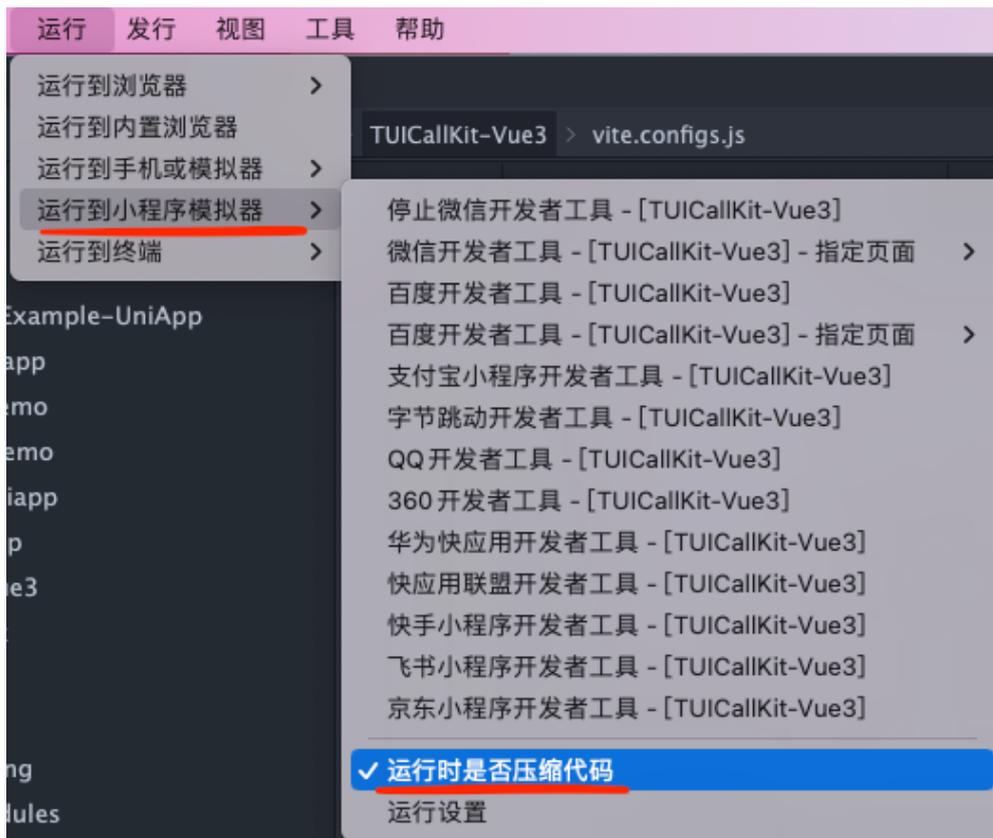
10. vue2 和 vue3 打包小程序包体积常见问题和方案

10.1 是否压缩代码？

以 [TUICallKit-Vue2 demo](#) 举例，下图给出了是否勾选运行时是否压缩代码的效果对比（851KB -> 454KB，降低 50%）：



因此：HBuilder 中 运行 > 运行到小程序模拟器 > 勾选运行时是否压缩代码，具体参考下图：



≤ v1.4.4

10.2 依赖包如何不打入到主包的 `common/vendor.js` ?

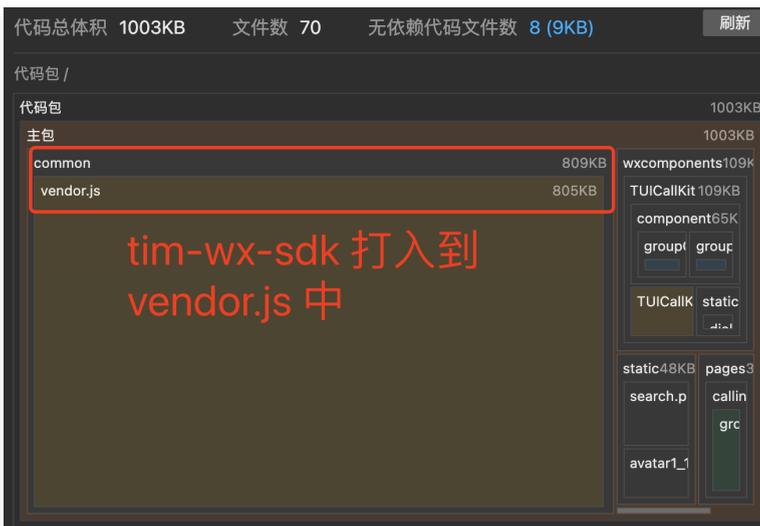
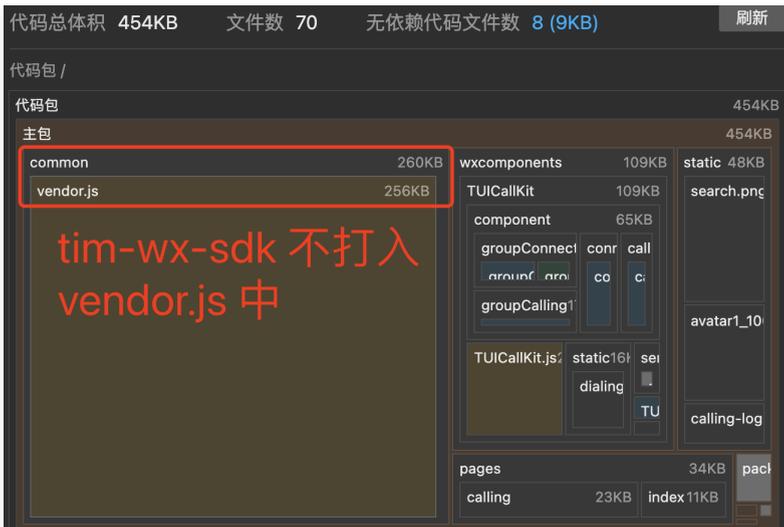
在使用 `微信开发者工具` 里 `构建 npm` 时，我们希望 HBuilder 打包小程序时，不将 `node_module` 中的依赖打入到主包的 `common/vendor.js` 中（这会导致依赖存在多份，影响主包体积）。

Vue2

推荐通过配置 `vue.config.js` 实现。

- `vue.config.js` 可选配置文件需放置在项目根目录下。
- `vue.config.js` 一般用于配置 `webpack` 等编译选项。HBuilder 采用 `vue2` 开发小程序，构建、打包时 `vue.config.js` 会被自动加载。

以 `TUICallKit-Vue2 demo` 举例，下图给出依赖包不打入主包的 `common/vendor.js` 的效果对比（805KB -> 256KB，降低 68%）：



具体使用如下:

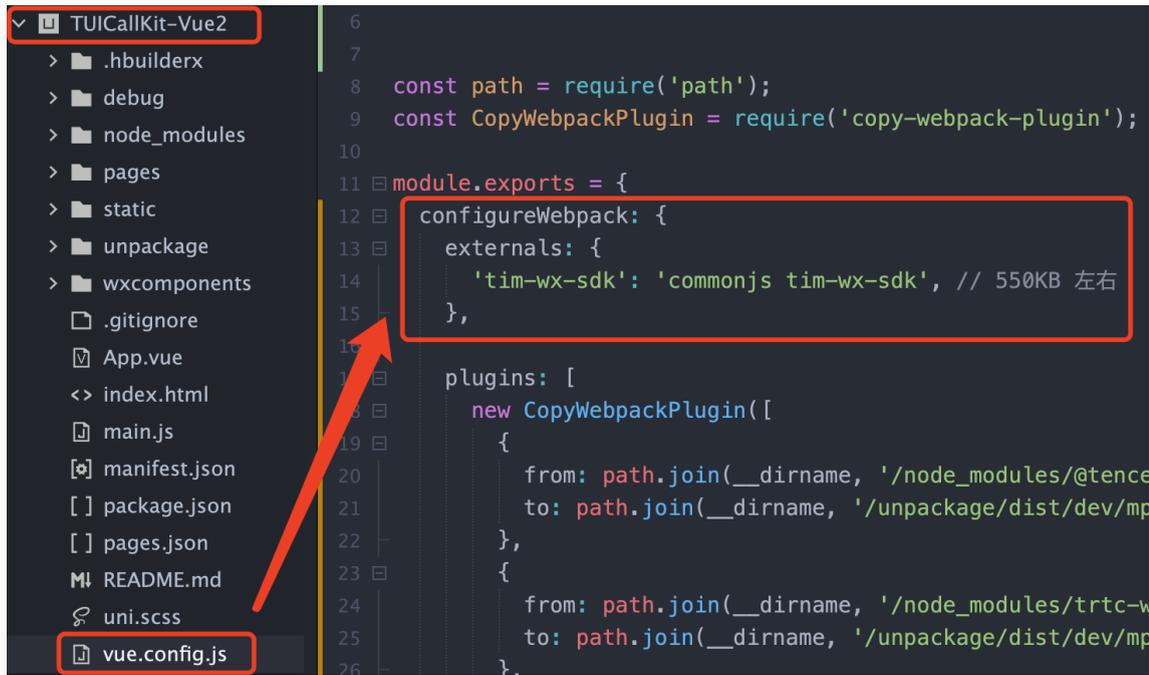
1. 页面中通过 `require` 引入包依赖:

```
// 页面中引入方式必须使用 require, 不能使用 import 的方式
const TIM = require('tim-wx-sdk');
```

2. `vue.config.js` 配置如下:

```
// vue.config.js 配置
module.exports = {
  configureWebpack: {
    externals: {
      'tim-wx-sdk': 'commonjs tim-wx-sdk', // 550KB 左右
    },
  },
}
```

具体目录结构如下截图：



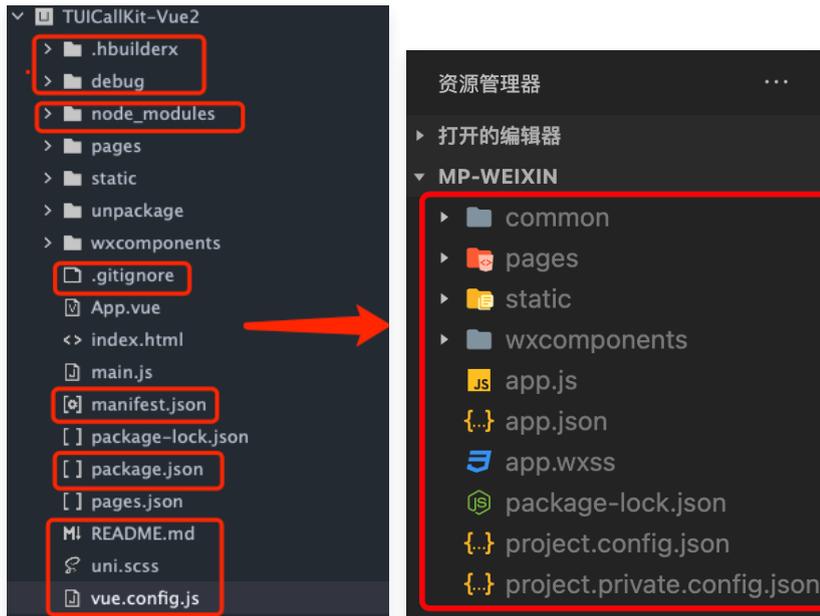
注意：

页面中引入依赖包的方式必须使用 `require`，不能使用 `import` 方式！

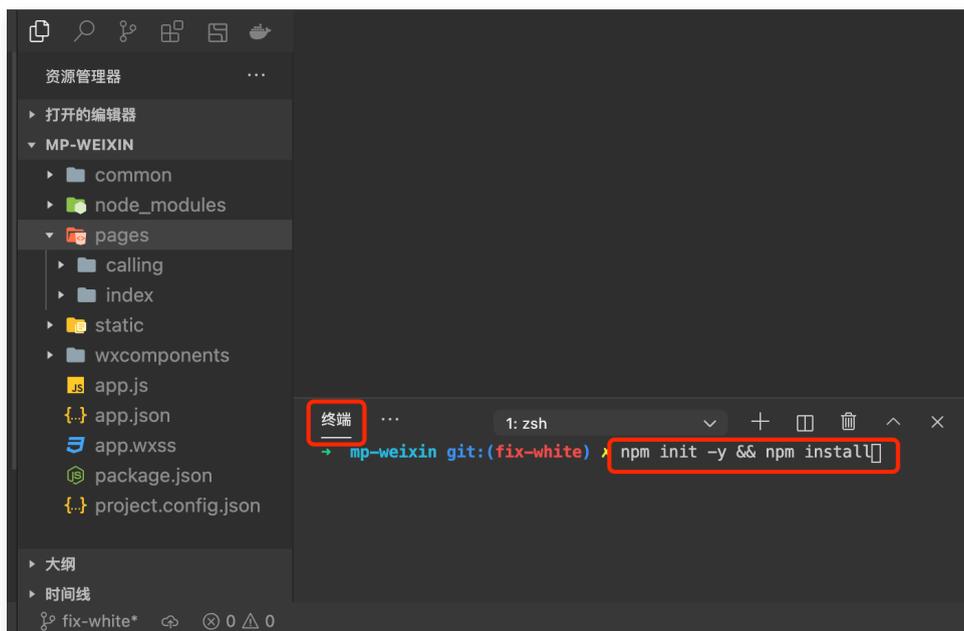
原因：通过 `import` 引入时，HBuilder 打包小程序后，会将 `import` 引入方式进行编译。从而导致无法使用 `node_modules` 中的 `tim-wx-sdk`。而 `require` 的方式则不会被编译。

10.3 如何构建 npm?

1. 通过和 uni-app 的技术人员沟通了解到，目前 HBuilder 中不支持类似微信开发者工具中的构建 `npm`。而且他们也没有支持的计划。
2. HBuilder 中运行 > 运行到小程序模拟器生成的小程序项目，会清除 `package.json`、`node_modules` 等文件。具体如下图：



如果想在微信开发者工具 > 工具 > 构建 npm，就必须在微信开发者工具的终端里创建 `package.json`、`npm install` 安装依赖。然后才能构建 npm。具体如下：

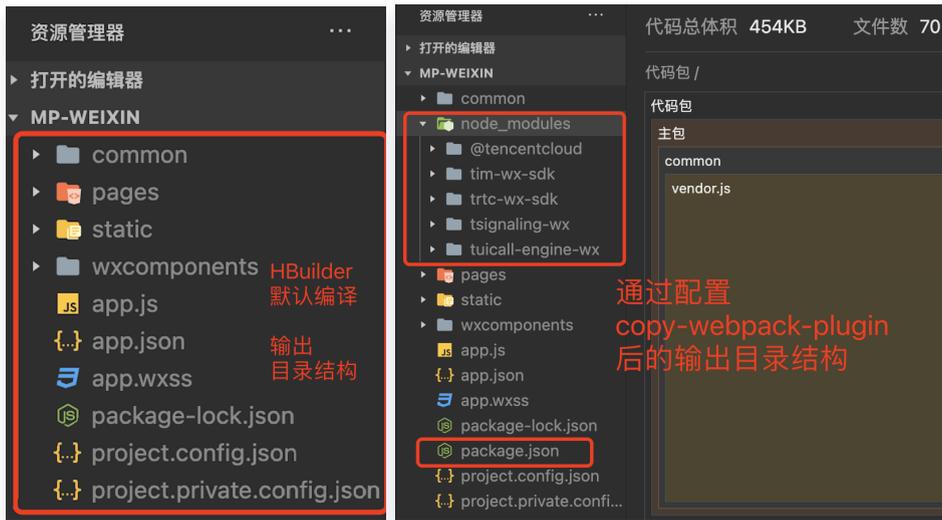


```
npm init -y && npm install xx
```

10.4 如何使用 `copy-webpack-plugin` 提升效率?

上面 [3. 如何构建 npm](#) 中，需要在 `微信开发者工具` 的终端里创建 `package.json`，然后安装依赖才行。通过 `copy-webpack-plugin` 可减少该步骤，并提升效率。

以 `TUICallKit-Vue2 demo` 举例，下图给出使用 `copy-webpack-plugin` 的前后对比：



具体使用如下：

1. 安装 copy-webpack-plugin：

注意：

最新版本 copy-webpack-plugin 插件暂不兼容，推荐 v5.0.0（锁死版本）

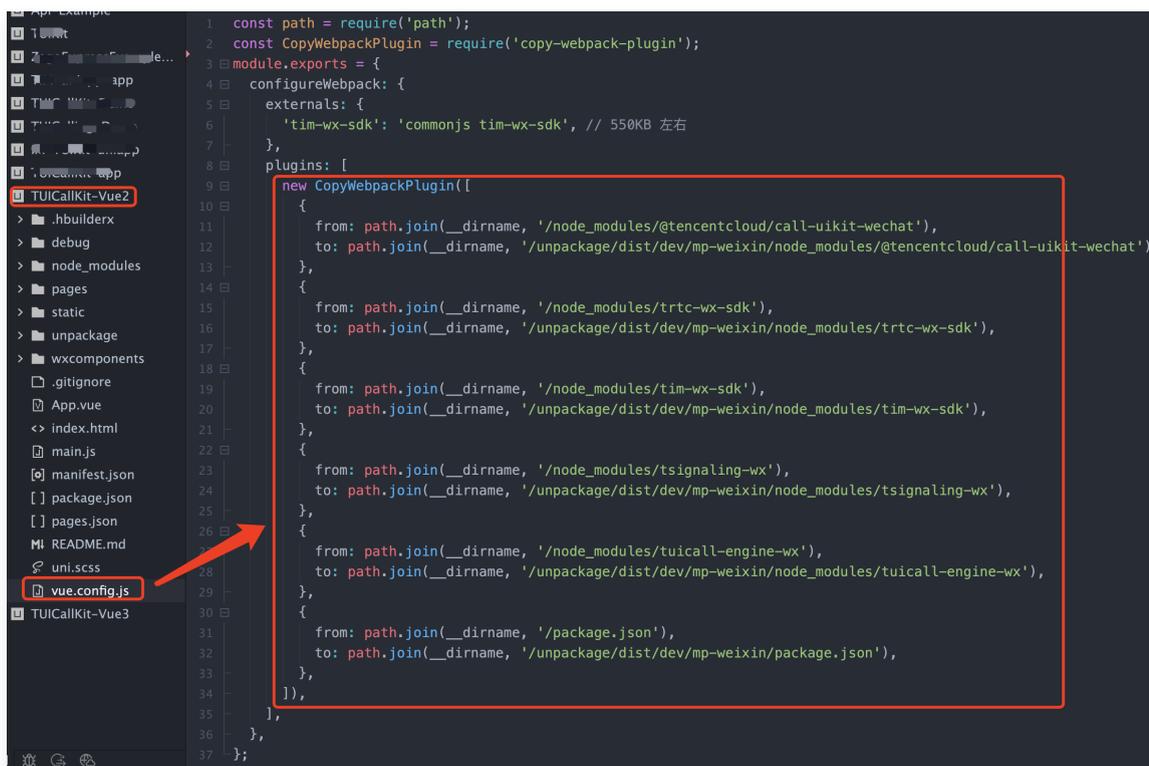
```
npm install -D copy-webpack-plugin@5.0.0
```

2. 在 vue.config.js 中使用 copy-webpack-plugin 插件，具体如下：

```
// vue.config.js 配置
const path = require('path');
const CopyWebpackPlugin = require('copy-webpack-plugin');
module.exports = {
  configureWebpack: {
    plugins: [
      new CopyWebpackPlugin([
        {
          from: path.join(__dirname,
            '/node_modules/@tencentcloud/call-uikit-wechat'),
          to: path.join(__dirname,
            '/unpackage/dist/dev/mp-weixin/node_modules/@tencentcloud/call-uikit-wechat')
        },
        {
          from: path.join(__dirname,
            '/node_modules/trtc-wx-sdk'),
          to: path.join(__dirname,
            '/unpackage/dist/dev/mp-weixin/node_modules/trtc-wx-sdk')
        }
      ])
    ]
  }
}
```

```
    },
    {
      from: path.join(__dirname,
        '/node_modules/tim-wx-sdk'),
      to: path.join(__dirname,
        '/unpackage/dist/dev/mp-weixin/node_modules/tim-wx-sdk')
    },
    {
      from: path.join(__dirname,
        '/node_modules/tsignaling-wx'),
      to: path.join(__dirname,
        '/unpackage/dist/dev/mp-weixin/node_modules/tsignaling-wx')
    },
    {
      from: path.join(__dirname,
        '/node_modules/tuicall-engine-wx'),
      to: path.join(__dirname,
        '/unpackage/dist/dev/mp-weixin/node_modules/tuicall-engine-
wx')
    },
    {
      from: path.join(__dirname, '/package.json'),
      to: path.join(__dirname,
        '/unpackage/dist/dev/mp-weixin/package.json')
    }
  ])
}
}
```

具体目录结构如下截图：



```
1 const path = require('path');
2 const CopyWebpackPlugin = require('copy-webpack-plugin');
3 module.exports = {
4   configureWebpack: {
5     externals: {
6       'tim-wx-sdk': 'commonjs tim-wx-sdk', // 550KB 左右
7     },
8     plugins: [
9       new CopyWebpackPlugin([
10        {
11          from: path.join(__dirname, '/node_modules/@tencentcloud/call-uikit-wechat'),
12          to: path.join(__dirname, '/unpackage/dist/dev/mp-weixin/node_modules/@tencentcloud/call-uikit-wechat'),
13        },
14        {
15          from: path.join(__dirname, '/node_modules/trtc-wx-sdk'),
16          to: path.join(__dirname, '/unpackage/dist/dev/mp-weixin/node_modules/trtc-wx-sdk'),
17        },
18        {
19          from: path.join(__dirname, '/node_modules/tim-wx-sdk'),
20          to: path.join(__dirname, '/unpackage/dist/dev/mp-weixin/node_modules/tim-wx-sdk'),
21        },
22        {
23          from: path.join(__dirname, '/node_modules/tsignaling-wx'),
24          to: path.join(__dirname, '/unpackage/dist/dev/mp-weixin/node_modules/tsignaling-wx'),
25        },
26        {
27          from: path.join(__dirname, '/node_modules/tuicall-engine-wx'),
28          to: path.join(__dirname, '/unpackage/dist/dev/mp-weixin/node_modules/tuicall-engine-wx'),
29        },
30        {
31          from: path.join(__dirname, '/package.json'),
32          to: path.join(__dirname, '/unpackage/dist/dev/mp-weixin/package.json'),
33        },
34      ]),
35    ],
36  },
37};
```

Vue3

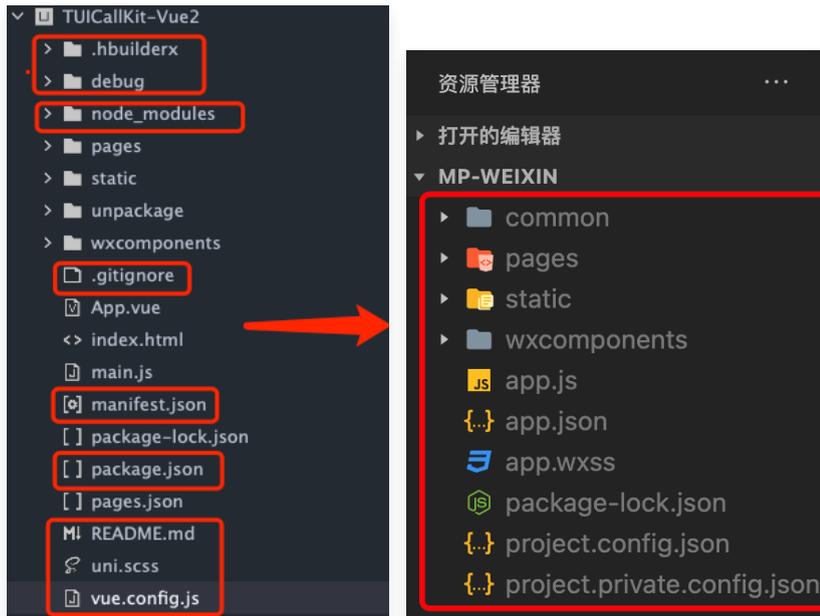
不用配置 `vite.config.js`，只需通过 `require` 引入即可。具体如下：

```
// 页面中引入方式必须使用 require, 不使用 import
const TIM = require('tim-wx-sdk');
```

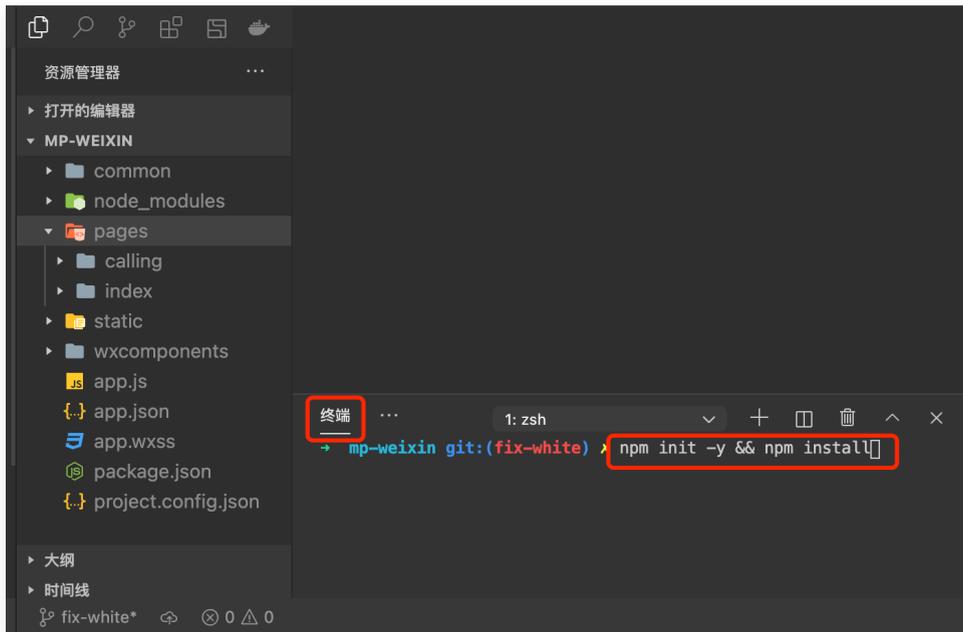
具体可参考：[TUICallKit-Vue3 demo](#)。

如何构建 npm?

1. 通过和 uni-app 的技术人员沟通了解到，目前 HBuilder 中不支持类似微信开发者工具中的构建 npm。而且他们也没有支持的计划。
2. HBuilder 中运行 > 运行到小程序模拟器生成的小程序项目，会清除 `package.json`、`node_modules` 等文件。具体如下图：



如果想在微信开发者工具 > 工具 > 构建 npm，就必须在微信开发者工具的终端里创建 `package.json`、`npm install` 安装依赖。然后才能构建 npm。具体如下：



```
npm init -y && npm install tuicall-engine-wx@1.5.6
```

11. Vue2 uni-app 打包微信小程序项目中，铃声没有声音？

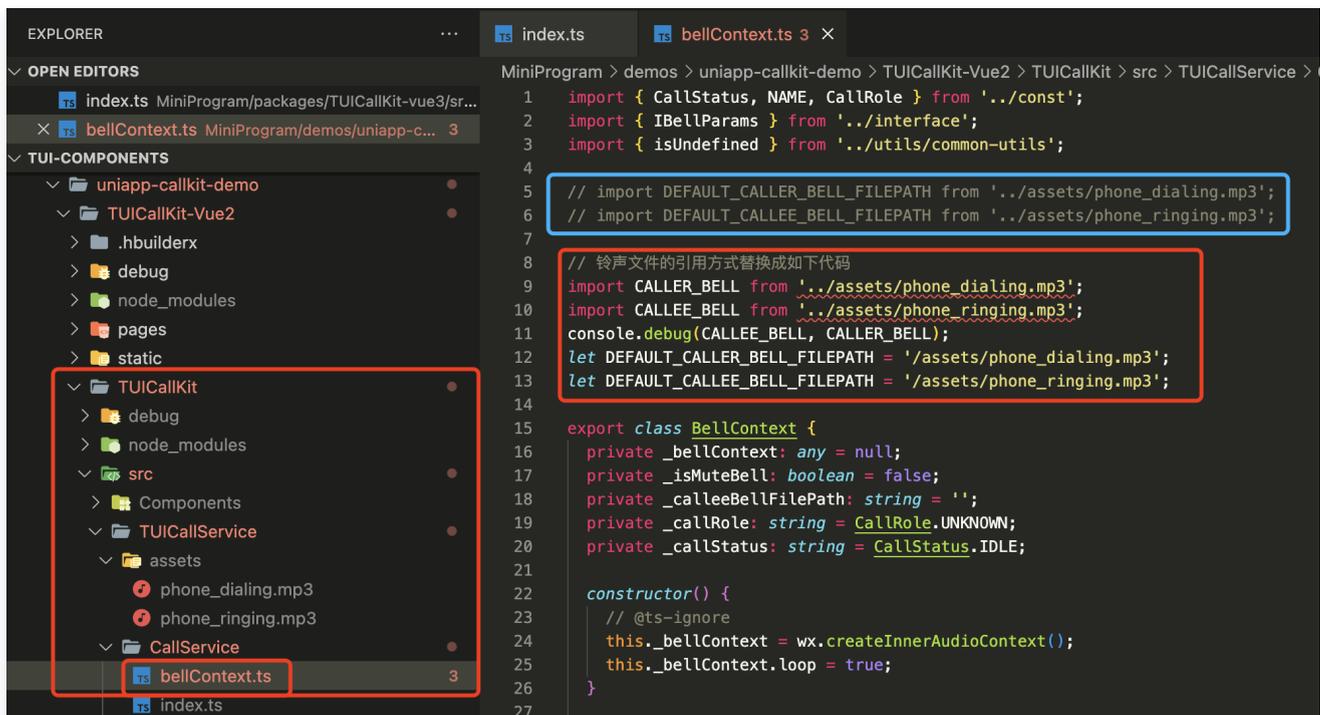
这个是因为在 vue2 中，uni-app 打包，将铃声文件编码为 base64 编码，这个编码在播放铃声的 api 那里是不支持的。

您可采取如下步骤解决该问题：

- 第一步，在项目的 TUICallKit 源码中，修改铃声的引入方式。

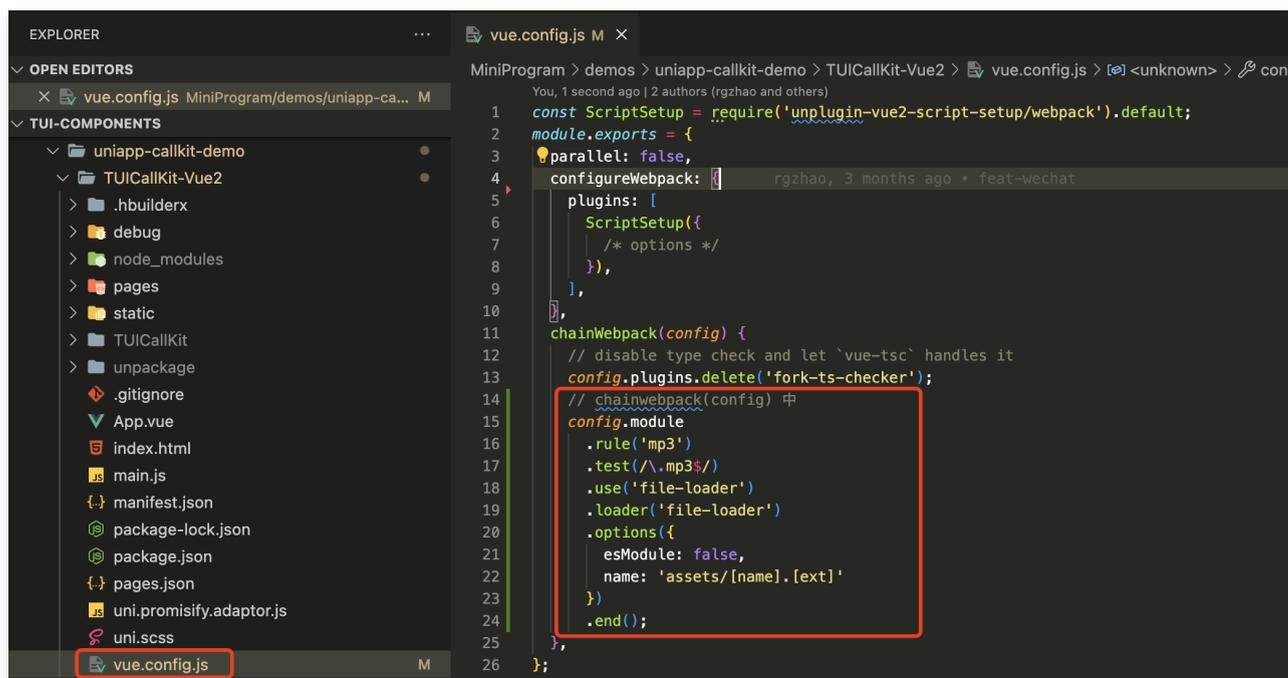
// 铃声文件的引用方式替换成如下代码

```
import CALLER_BELL from '../assets/phone_dialing.mp3';
import CALLEE_BELL from '../assets/phone_ringing.mp3';
console.debug(CALLEE_BELL, CALLER_BELL);
let DEFAULT_CALLER_BELL_FILEPATH = '/assets/phone_dialing.mp3';
let DEFAULT_CALLEE_BELL_FILEPATH = '/assets/phone_ringing.mp3';
```



- 第二步，修改项目中的 `vue.config.js` 文件。

```
// chainwebpack(config) 中
config.module
  .rule('mp3')
  .test(/\.(mp3$)/)
  .use('file-loader')
  .loader('file-loader')
  .options({
    esModule: false,
    name: 'assets/[name].[ext]'
  })
  .end();
```



```
vue.config.js
1  const ScriptSetup = require('unplugin-vue2-script-setup/webpack').default;
2  module.exports = {
3    parallel: false,
4    configureWebpack: {
5      plugins: [
6        ScriptSetup({
7          /* options */
8        }),
9      ],
10   },
11   chainWebpack(config) {
12     // disable type check and let `vue-tsc` handles it
13     config.plugins.delete('fork-ts-checker');
14     // chainwebpack(config) 中
15     config.module
16       .rule('mp3')
17       .test(/\.(mp3)/)
18       .use('file-loader')
19       .loader('file-loader')
20       .options({
21         esModule: false,
22         name: 'assets/[name].[ext]'
23       })
24       .end();
25   },
26 };
```

12. 包体积过大?

⚠ 注意:

TUICallKit 版本须 $\geq 3.1.2$ 。

- [体积优化文档](#)。

13. 什么是 SDKAppID、SecretKey?

- SDKAppID: IM 的应用 ID, 用于业务隔离, 即不同的 SDKAppID 的通话彼此不能互通。
- Secretkey: IM 的应用密钥, 需要和 SDKAppID 配对使用, 用于签出合法使用 IM 服务的鉴权用票据 UserSig。

14. 什么是 UserSig, 如何生成 UserSig?

- UserSig 是用户登录即时通信 IM 的密码, 其本质是对 UserID 等信息加密后得到的密文。
- UserSig 签发方式是将 UserSig 的计算代码集成到您的服务端, 并提供面向项目的接口, 在需要 UserSig 时由您的项目向业务服务器发起请求获取动态 UserSig。更多详情请参见 [服务端生成 UserSig](#)。

15. 处理 TUICallKit 的 ts 语法校验错误

如果出现如下所示 error, 请在项目中添加如下配置。

```
11:25:53.965 TS7034:Variable 'groupMemberList' implicitly has type 'any[]' in some locations where its type cannot be determined.
11:25:53.965 [ts1] ERROR at components\TUICallKit\src\TUICallService\CallServiceUtils.ts:122
11:25:53.971 TS7005:Variable 'groupMemberList' implicitly has an 'any[]' type.
11:25:53.971 [ts1] ERROR at components\TUICallKit\src\TUICallService\CallServiceUtils.ts:126
11:25:53.978 TS7005:Variable 'groupMemberList' implicitly has an 'any[]' type.
11:25:53.979 [ts1] ERROR at components\TUICallKit\src\TUICallService\serve\callManager.ts:16
11:25:53.985 TS7006:Parameter 'params' implicitly has an 'any' type.
11:25:53.985 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:218
11:25:53.992 TS7006:Parameter 'input' implicitly has an 'any' type.
11:25:53.992 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:219
11:25:53.999 TS2531:Object is possibly 'null'.
11:25:54.008 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:225
11:25:54.009 TS7006:Parameter 'obj' implicitly has an 'any' type.
11:25:54.015 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:225
11:25:54.016 TS7006:Parameter 'oldKey' implicitly has an 'any' type.
11:25:54.022 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:225
11:25:54.023 TS7006:Parameter 'newKey' implicitly has an 'any' type.
11:25:54.031 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:232
11:25:54.033 TS7053:Element implicitly has an 'any' type because expression of type 'any' can't be used to index type '{}'.
11:25:54.039 [ts1] ERROR at components\TUICallKit\src\TUICallService\utils\common-utils.ts:234
```

.eslintignore

```
TUICallKit
```

tsconfig.json

```
{
  "compilerOptions": {
    "noImplicitAny": true
  }
}
```

Flutter

最近更新时间：2024-08-07 10:04:42

同时集成 `tencent_calls_uikit` 和 `tencent_trtc_cloud`，或同时集成 `tencent_calls_uikit` 和 `live_flutter_plugin`，出现符号冲突报错，怎么解决？

问题详情：当引入 flutter "tencent_calls_uikit" 进我们现有的 project 后，Android 构建 APK 时出现如下报错：

```
Duplicate class com.tencent.liteav.LiveSettingJni found in modules
jetified-LiteAVSDK_Professional-10.7.0.13053-runtime
(com.tencent.liteav:LiteAVSDK_Professional:10.7.0.13053) and jetified-
LiteAVSDK_TRTC-10.3.0.11225-runtime
(com.tencent.liteav:LiteAVSDK_TRTC:10.3.0.11225)
```

iOS 在执行 `pod install` 出现如下报错：

```
[!] The 'Pods-Runner' target has frameworks with conflicting names:
txsoundtouch.xcframework and txffmpeg.xcframework.
```

这个问题原因是您使用的 `tencent_calls_uikit` 和 `tencent_trtc_cloud` 分别依赖于 TRTC Android SDK 的专业版和精简版，这个问题我们已经在最新的版本中解决，只需要您将 `tencent_calls_uikit` 和 `tencent_trtc_cloud` 升级到最新版本即可。

Flutter Android 未添加混淆设置，怎么设置？

如果您需要编译运行在 Android 平台，由于我们在 SDK 内部使用了 Java 的反射特性，需要将 SDK 中的部分类加入不混淆名单，因此需要您在 `proguard-rules.pro` 文件中添加如下代码：

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

从1.8.0以下的版本升级到1.8.0及以上的版本，导致编译报错或者拉不起页面问题修复？

如果是从1.8.0以下升级到1.8.0及以上的版本需要您检查以下步骤是否正常：

1. 将 `navigatorObservers` 添加到 `MateriaApp`。目的是在收到呼叫邀请时导航到 `TUICallKit` 页面。示例代码如下：

```
import 'package:tencent_calls_uikit/tuicall_kit.dart';

MaterialApp (
```

```
navigatorObservers: [TUICallKit.navigatorObserver],
    ...
)
```

2. tencent_calls_engine 插件中的导入文件统一替换为新的。

```
import 'package:tencent_calls_engine/tuicall_engine.dart';
import 'package:tencent_calls_engine/tuicall_observer.dart';
import 'package:tencent_calls_engine/tuicall_define.dart';
```

上图内容替换为下图所示：

```
import 'package:tencent_calls_engine/tencent_calls_engine.dart';
```

3. 登录 API 调整更规范，不需要再指定参数。

<pre>return await callsUIKitPlugin.login(sdkAppId: GenerateTestUserSig.sdkAppId, userId: userInfo.userId, userSig: GenerateTestUserSig.genTestSig(userInfo.userId));</pre>	<p>161 164</p> <p>162 165</p> <p>163 166</p> <p>164 167</p>	<pre>return await callsUIKitPlugin.login(GenerateTestUserSig.sdkAppId, userInfo.userId, GenerateTestUserSig.genTestSig(userInfo.userId));</pre> <p>18.0以上版本不需要指定参数</p>
---	---	--

4. 离线推送参数构造优化。

<pre>TUIOfflinePushInfo offlinePushInfo = TUIOfflinePushInfo(); offlinePushInfo.title = "Flutter TUICallKit"; offlinePushInfo.desc = "This is an incoming call from Flutter TUICallkit"; TUICallParams params = TUICallParams(offlinePushInfo: offlinePushInfo); callsUIKitPlugin.call(callUserId, TUICallMediaType.audio, params);</pre>	<p>146 144</p> <p>147 145</p> <p>148 146</p> <p>149 147</p> <p>150 148</p> <p>151 149</p>	<pre>TUIOfflinePushInfo offlinePushInfo = TUIOfflinePushInfo(); offlinePushInfo.title = "Flutter TUICallKit"; offlinePushInfo.desc = "This is an incoming call from Flutter TUICallkit"; TUICallParams params = TUICallParams(); params.offlinePushInfo = offlinePushInfo; callsUIKitPlugin.call(callUserId, TUICallMediaType.audio, params);</pre>
---	---	---