

音视频通话 SDK

场景方案



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

场景方案

在线视频客服方案

iOS

Android

Web & H5

小程序

uni-app

场景方案

在线视频客服方案

iOS

最近更新时间：2024-11-25 15:41:12

开发环境要求

- Xcode 10 及以上
- iOS 9.0 及以上

CocoaPods 集成

TUIKit 从 5.7.1435 版本开始支持模块化集成，您可以根据自己的需求选择所需模块集成。

1. 根据实际业务需求在 Podfile 中添加对应的 TUI 组件，比如需要聊天功能，可以添加 `pod 'TUIChat'`；需要会话列表功能，可以添加 `pod 'TUIConversation'`；需要音视频通话功能，可以添加 `pod 'TUICallKit'`。TUI 组件之间相互独立，添加或删除均不影响工程编译。

```
# 防止 TUI 组件里的 *.xcassets 与您项目里面冲突。
install ! 'cocoapods', :disable_input_output_paths => true

# TUI 组件依赖了静态库，需要屏蔽如下设置，如果报错，请参考常见问题说明。
# use_frameworks!

# 集成聊天功能
pod 'TUIChat'
# 集成会话功能
pod 'TUIConversation'
# 集成关系链功能
pod 'TUIContact'
# 集成群组功能
pod 'TUIGroup'
# 集成搜索功能（需要购买旗舰版套餐）
pod 'TUISearch'
# 集成音视频通话功能
pod 'TUICallKit'
```

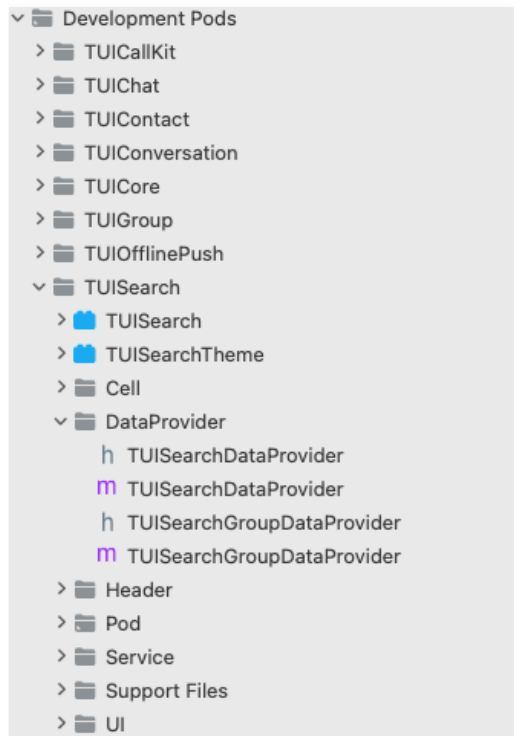
2. 执行以下命令，安装 TUI 组件。

```
pod install
```

如果无法安装 TUIKit 最新版本，执行以下命令更新本地的 CocoaPods 仓库列表。

```
pod repo update
```

TUI 组件集成后效果：



快速搭建

常用的聊天软件都是由会话列表、聊天窗口、好友列表、音视频通话等几个基本的界面组成，参考下面步骤，您仅需几行代码即可在项目中快速搭建这些 UI 界面。

步骤一：组件登录

```
#import "TUILogin.h"
// 您可以在用户 UI 点击登录的时候登录 UI 组件
- (void)loginSDK:(NSString *)userID userSig:(NSString *)sig succ:(TSucc)succ fail:(TFail)fail {
    // SDKAppID 可以在 即时通信 IM 控制台中获取
    // userSig生成见 GenerateTestUserSig.h
    [TUILogin login:SDKAppID userID:userID userSig:sig succ:^(
        NSLog(@"-----> 登录成功");
    ) fail:^(int code, NSString *msg) {
        NSLog(@"-----> 登录失败");
    }];
}
```

步骤二：构建会话列表

会话列表只需要创建 `TUIConversationListController` 对象即可。会话列表会从数据库中读取最近联系人，当用户单击联系人时，`TUIConversationListController` 将该事件回调给上层。

```
@implementation ConversationController // 您自己的 ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    // 创建 TUIConversationListController
    TUIConversationListController *conv = [[TUIConversationListController alloc] init];
    conv.delegate = self;
    // 把 TUIConversationListController 添加到自己的 ViewController
    [self addChildViewController:conv];
    [self.view addSubview:conv.view];
}
```

```
}

- (void)conversationListController:(TUIConversationListController *)conversationController
didSelectConversation:(TUIConversationCell *)conversation
{
    // 会话列表点击事件，通常是打开聊天界面
}

@end
```

步骤三：构建聊天窗口

初始化聊天界面时，上层需要传入当前聊天界面对应的会话信息，示例代码如下：

```
@implementation ChatViewController // 您自己的 ViewController
- (void)viewDidLoad {
    // 创建会话信息
    TUIChatConversationModel *data = [[TUIChatConversationModel alloc] init];
    data.userID = @"userID";
    // 创建 TUIC2CChatViewController
    TUIC2CChatViewController *vc = [[TUIC2CChatViewController alloc] init];
    [vc setConversationData:data];
    // 把 TUIC2CChatViewController 添加到自己的 ViewController
    [self addChildViewController:conv];
    [self.view addSubview:conv.view];
}

@end
```

`TUIC2CChatViewController` 会自动拉取该用户的历史消息并展示出来。

步骤四：构建通讯录界面

通讯录界面不需要其它依赖，只需创建对象并显示出来即可。

```
@implementation ContactController // 您自己的 ViewController
- (void)viewDidLoad {
    // 创建 TUIContactController
    TUIContactController *vc = [[TUIContactController alloc] init];
    // 把 TUIContactController 添加到自己的 ViewController
    [self addChildViewController:conv];
    [self.view addSubview:conv.view];
}

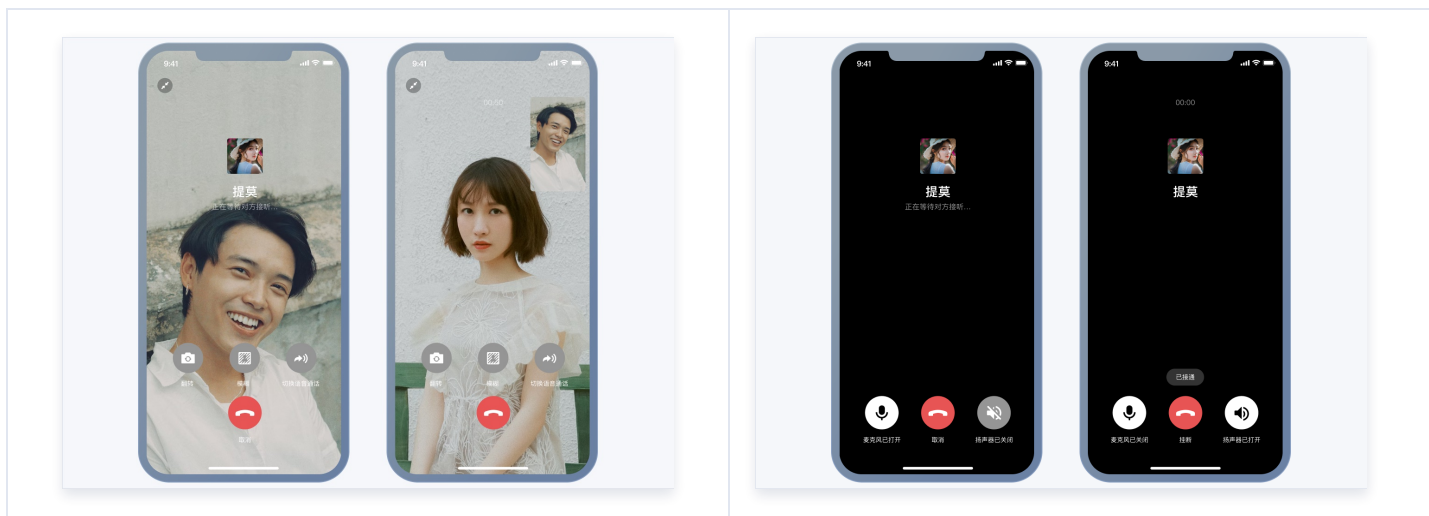
@end
```

步骤五：构建音视频通话功能

TUI 组件支持在聊天界面对用户发起音视频通话，仅需要简单几步就可以快速集成：

视频通话

语音通话



1. 创建音视频服务

1.1 创建项目。创建项目，选择类型及场景，开通相关服务。

1.1.1 登录 [腾讯云视立方控制台](#) > [项目管理](#)，单击创建项目开始创建。



1.1.2 选择项目类型。可选择创建新项目并输入项目名称，或关联已有项目。

1.1.3 选择接入场景。选择快速接入的音视频场景，分为聊天应用和音视频通话。



1.1.4 选择集成方式。选择含 UI 快速集成。

1.1.5 开通相关服务。使用音视频通话 SDK 需提前开通即时通信 IM 和实时音视频 TRTC 服务，仅开通服务，不涉及付费购买产品。

1.1.6 单击创建项目并下一步，完成项目创建，进入开发指引。

详细功能 [展开](#)

选择集成方式

含 UI 快速集成 推荐

提供视频通话、房间管理等多个音视频标准组件 + 标准 SDK + 标准 UI 界面 + 完整的接入指引。仅需 1 天即可快速搭建完成。



不含 UI 集成

仅提供 TRTC SDK 和 IM SDK。UI 界面和高级功能（邀请聊天、房间管理等）需用户调用 API 接口自行实现。

开通相关服务

此处仅为您开通对应云产品服务，不涉及付费购买产品。开通后您可正常使用相关产品控制台和功能服务。已开通产品服务不会重复开通。

即时通信 IM 【已开通】

了解即时通信 IM 产品，[点击查看更多详情](#)

实时音视频 TRTC 服务 【已开通】

了解实时音视频 TRTC 产品，[点击查看更多详情](#)

创建项目并下一步

1.2 准备开发。

1.2.1 体验 Demo。您可提前体验 Demo 了解 SDK 应用效果，包括 Android、iOS 和 Web 端。



版权所有：腾讯云计算（北京）有限责任公司


第8 共40页


1.2.2 关联/创建 IM 应用。音视频通话创建需要使用 IM 服务，请选择一个 IM 应用进行关联或者创建新的应用。

✓ 创建项目 > 2 准备开发

● 体验 DEMO


Android



iOS



Web

点击体验

● 关联/创建 IM 应用

音视频通话场景需要使用 IM 服务，请选择一个 IM 应用进行关联或者创建新的应用。

应用类型 ☒ 关联已有 IM 应用 ☐ 创建 IM 应用

应用名称

1.2.3 领取资源。每个应用可免费体验音视频通话 SDK 功能一次，有效期7天，音视频通话 SDK 体验版详细能力支持请参见 [套餐包功能说明](#)。业务上线请升级购买正式版本套餐，享受完整功能。

● 领取资源

含 UI 快速集成方式需使用相关资源，开通项目即赠送音视频通话-体验版（有效期 7 天），有效期内使用体验版进行 SDK 集成测试。业务上线请升级至正式版本套餐，享受完整功能。

音视频通话-体验版 有效期 7 天，限领一次

套餐包内容			
用户上限	100 个	群组上限	100 个
直播群数上限	10 个	免费消息存储时长	7 天
TRTC 通用时长包	1000 分钟	通话状态展示	包含
通话呼叫通知	✓	多人通话	✓
全员推送	✓	创建社群	✓
AI 降噪	✓	视频人像清晰	✓
户外通话卡顿优化	✓	音频低码流畅	✓

前往集成

1.2.4 单击 前往集成。

1.3 集成指南

1.3.1 选择集成环境并下载 TUIKit 开发包。集成环境包括 Android 和 iOS。

1.3.2 集成测试。下载 TUIKit 开发包后，参照集成测试的步骤查看相关集成文档完成集成测试。

1.3.3 正式开发。完成 SDK 接入测试后，若需正式开发并上线音视频应用，可购买音视频通话套餐包，包括基础版、进阶版和尊享版，升级当前项目业务版本，享受完整功能。

集成指南

选择集成环境并下载 TUIKit 开发包

Android SDK 开发包

[Github 下载](#) [集成文档](#)

iOS SDK 开发包

[Github 下载](#) [集成文档](#)

Web SDK 开发包

[Github 下载](#) [集成文档](#)

集成效果可以参考体验 Demo [展开](#)

集成测试

点击下方步骤即可查看相关集成文档

1 参考 源码集成 文档

2 参考 组件登录 文档

3 参考 构建会话列表 文档

4 参考 构建聊天界面 文档

5 参考 集成音视频通话能力 文档

6 参考 设置界面风格 文档

7 参考 添加自定义消息 文档

8 参考 实现本地搜索 文档

9 参考 接入离线推送 文档

1 源码集成

2 组件登录

3 构建会话列表

4 构建聊天界面

5 集成音视频通话能力

6 设置界面风格

7 添加自定义消息

8 实现本地搜索

9 接入离线推送

正式开发

参照上述步骤，您已成功完成含 UI 的集成测试。正式开发并上线音视频应用请点击下方购买按钮，购买音视频通话 - 基础版/进阶版/尊享版，享受完整功能。音视频通话 - 体验版到期后，标准音视频组件和标准 UI 将无法使用，多人通话、多端登录等高级特性也将失效。

	音视频通话（基础版）	音视频通话（进阶版）	音视频通话（尊享版）
	1499 元/月 立即购买	1999 元/月 立即购买	4499 元/月 立即购买
赠送通话时长	11 万分钟	23.5 万分钟	38 万分钟
微信同款 UI 设计	✓	✓	✓
通话呼叫/接听/拒绝/挂断	✓	✓	✓
多人通话	×	✓	✓
中途呼叫/加入三方通话	×	✓	✓
视频通话切换语音通话	✓	✓	✓
通话状态展示	✓	✓	✓
通话呼叫通知	✓	✓	✓
通话悬浮窗	✓	✓	✓
自定义呼叫铃声	✓	✓	✓
AI 降噪	×	✓	✓
户外通话卡顿优化	×	✓	✓
多端登录通话	×	×	✓
同平台多端登录通话	×	×	✓
全员推送	×	×	✓
同平台多设备在线	×	×	✓

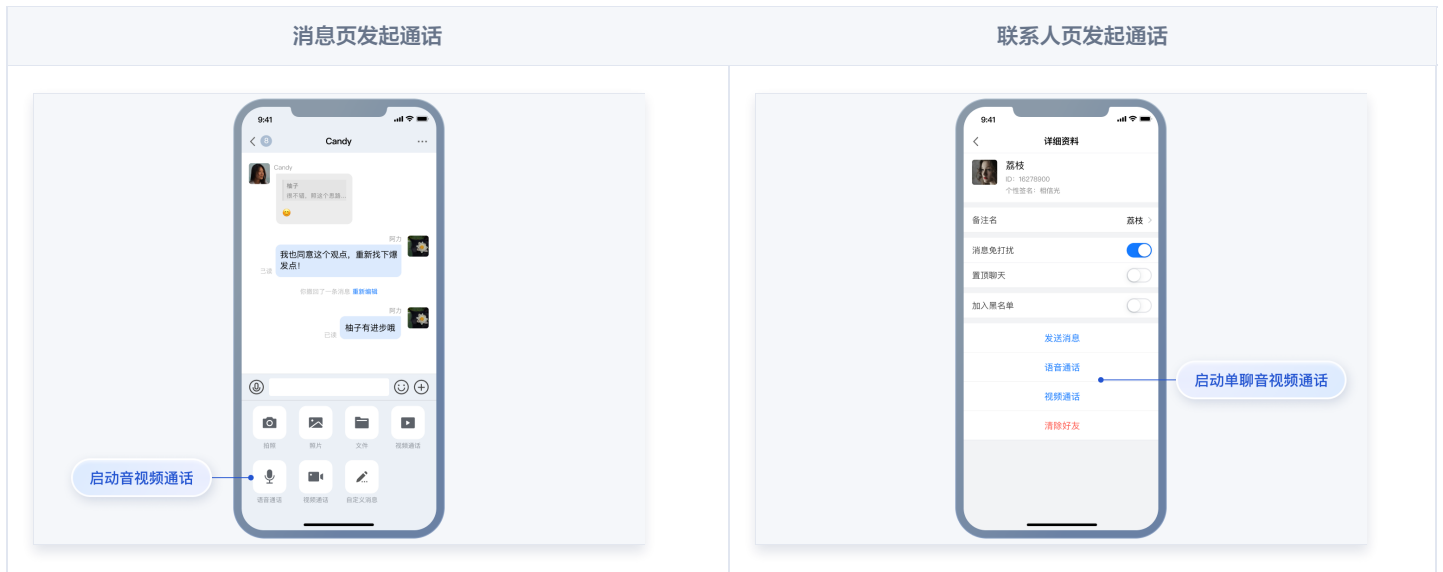
2. 集成 TUICallKit 组件在 podfile 文件中添加以下内容。

```
// 集成音视频通话组件
pod 'TUICallKit'
```

3. 发起和接收视频或语音通话

版权所有：腾讯云计算（北京）有限责任公司

第10 共40页



- 集成 TUICallKit 组件后，聊天界面和联系人资料界面默认会出现“视频通话”和“语音通话”两个按钮，当用户单击按钮时，TUIKit 会自动展示通话邀请 UI，并给对方发起通话邀请请求。
- 当用户在线收到通话邀请时，TUIKit 会自动展示通话接收 UI，用户可以选择同意或者拒绝通话。
- 当用户离线收到通话邀请时，如需唤起 App 通话，就要使用到离线推送能力，离线推送的实现请参见 [添加离线推送](#)。

4. 添加离线推送

在使用离线推送之前，您需要开通 [IM 离线推送](#) 服务。

关于 App 的配置，您可以参考文档：[集成 TUIOfflinePush 跑通离线推送功能](#)。

配置完成后，当单击接收到的音视频通话离线推送通知时，TUICallKit 会自动拉起音视频通话邀请界面。

❗ 说明：

更多实操教学视频请参见：[极速集成 TUIKit \(iOS\)](#)。

常见问题

1、提示 "target has transitive dependencies that include statically linked binaries" 如何处理？

如果在 pod 过程中出现该错误，是因为 TUIKit 使用到了第三方静态库，需要在 podfile 中注释掉 `use_frameworks!`。

如果在某种情况下，需要使用 `use_frameworks!`，则请使用 cocoapods 1.9.0 及以上版本进行 `pod install`，并修改为：

```
use_frameworks! :linkage => :static
```

如果您使用的是 swift，请将头文件引用改成 `@import` 模块名形式引用。

2、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，会有符号冲突，如果您使用了非 [TRTC](#) 版本的音视频库，建议先去掉，然后 pod 集成 `TUICallKit/Professional` 版本，该版本依赖的 [LiteAV_Professional](#) 音视频库包含了音视频的所有基础能力。如果您使用了 [LiteAV_Enterprise](#) 音视频库，暂不支持和 TUICallKit 共存。具体解决方案可以参考文档：[音视频常见问题](#)。

3、通话邀请的超时时间默认是多久？

通话邀请的默认超时时间是 30 秒。

4、在邀请超时时间内，被邀请者如果离线再上线，能否立即收到邀请？

- 如果是单聊通话邀请，被邀请者离线再上线可以收到通话邀请，TUIKit 内部会自动唤起通话邀请界面。
- 如果是群聊通话邀请，被邀请者离线再上线后会自动拉取最近 30 秒内的邀请，TUIKit 会自动唤起群通话界面。

交流与反馈

欢迎加入 QQ 群进行技术交流和反馈问题。



Android

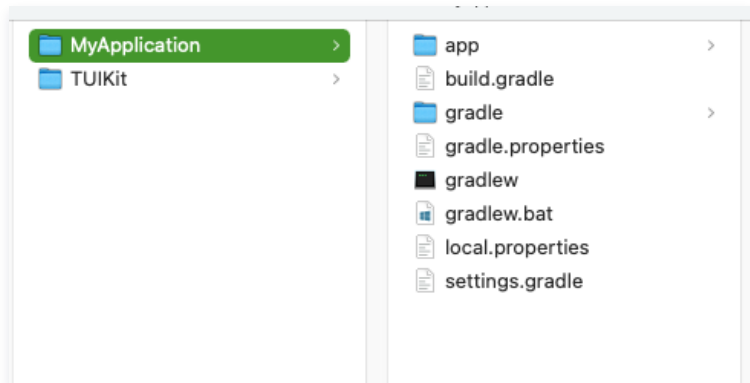
最近更新时间：2025-05-22 10:11:41

开发环境要求

- Android Studio 3.6.1
- Gradle-5.1.1
- Android Gradle Plugin Version-3.4.0

module 源码集成

1. 从 [GitHub 下载](#) TUIKit 源码。使 TUIKit 文件夹跟自己的工程文件夹同级，例如：



2. 根据实际业务需求在 settings.gradle 中添加对应的 TUI 组件，比如需要聊天功能，可以添加 tuichat；需要会话列表功能，可以添加 tuiconversation；需要音视频通话功能，可以添加 tuicallkit。TUI 组件之间相互独立，添加或删除均不影响工程编译。

```
// 引入上层应用模块
include ':app'

// 引入内部组件通信模块（必要模块）
include ':tuicore'
project(':tuicore').projectDir = new File(settingsDir, '../TUIKit/TUICore/tuicore')

// 引入聊天功能模块（基础功能模块）
include ':tuichat'
project(':tuichat').projectDir = new File(settingsDir, '../TUIKit/TUIChat/tuichat')

// 引入关系链功能模块（基础功能模块）
include ':tuicontact'
project(':tuicontact').projectDir = new File(settingsDir, '../TUIKit/TUIContact/tuicontact')

// 引入会话功能模块（基础功能模块）
include ':tuiconversation'
project(':tuiconversation').projectDir = new File(settingsDir,
'../TUIKit/TUIConversation/tuiconversation')

// 引入搜索功能模块（需要购买旗舰版套餐）
include ':tuisearch'
project(':tuisearch').projectDir = new File(settingsDir, '../TUIKit/TUISearch/tuisearch')

// 引入群组功能模块
include ':tuigroup'
project(':tuigroup').projectDir = new File(settingsDir, '../TUIKit/TUIGroup/tuigroup')
```

```
// 引入离线推送功能模块
include ':tuiofflinepush'
project(':tuiofflinepush').projectDir = new File(settingsDir,
'../TUIKit/TUIOfflinePush/tuiofflinepush')

// 引入音视频通话功能模块
include ':tuicallkit'
project(':tuicallkit').projectDir = new File(settingsDir, '../TUIKit/TUICallKit/tuicallkit')
```

3. 在 APP 的 build.gradle 中添加:

```
dependencies {
    api project(':tuiconversation')
    api project(':tuicontact')
    api project(':tuichat')
    api project(':tuisearch')
    api project(':tuigroup')
    api project(':tuiofflinepush')
    api project(':tuicallkit')
}
```

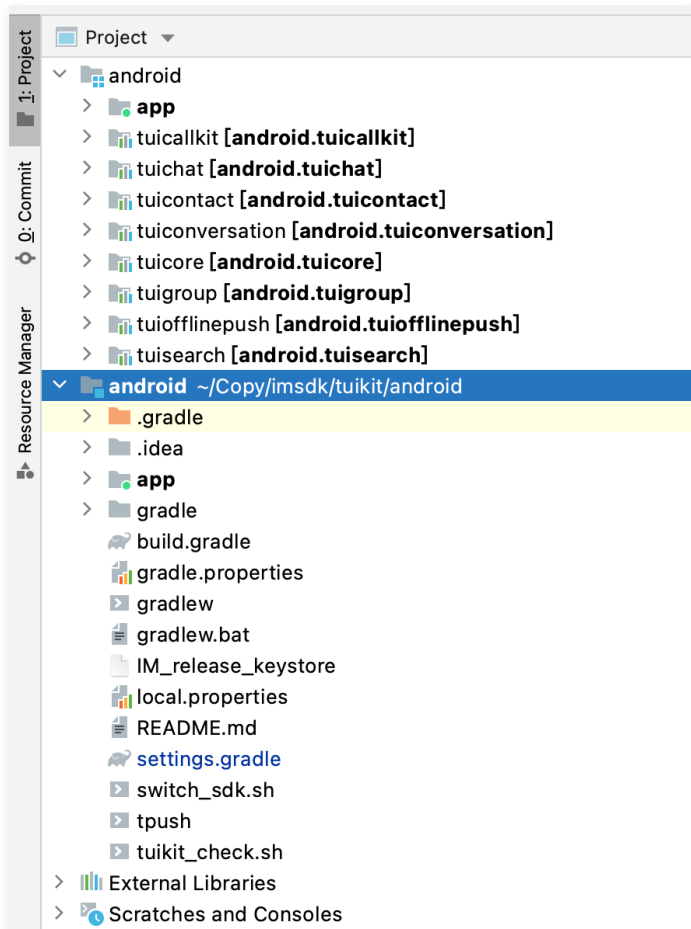
4. 在 gradle.properties 文件中加入下行, 表示自动转换三方库以兼容 AndroidX:

```
android.enableJetifier=true
```

5. 添加 maven 仓库, 在 root 工程的 build.gradle 文件中添加:

```
allprojects {
    repositories {
        mavenCentral()
        maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
    }
}
```

6. 同步工程, 编译运行。工程结构预期效果如图所示:



快速搭建

常用的聊天软件都是由会话列表、聊天窗口、好友列表、音视频通话等几个基本的界面组成，参考下面步骤，您仅需几行代码即可在项目中快速搭建这些 UI 界面。

步骤一：组件登录

```
// 在用户 UI 点击登录的时候调用
TUILogin.login(context, sdkAppID, userID, userSig, new TUICallback() {
    @Override
    public void onError(final int code, final String desc) {
    }

    @Override
    public void onSuccess() {
    }
});
```

⚠ 注意：

context 必须传 Application 对象，否则部分图片无法加载。

步骤二：创建 viewPager

1. 在 activity_main.xml 中添加界面布局：

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/view_pager"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight = "1"/>

</LinearLayout>
```

2. 创建 FragmentAdapter.java 用来配合 ViewPager2 展示会话和联系人界面。

```
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentActivity;
import androidx.fragment.app.FragmentManager;
import androidx.lifecycle.Lifecycle;
import androidx.viewpager2.adapter.FragmentStateAdapter;

import java.util.List;

public class FragmentAdapter extends FragmentStateAdapter {
    private static final String TAG = FragmentAdapter.class.getSimpleName();

    private List<Fragment> fragmentList;

    public FragmentAdapter(@NonNull FragmentActivity fragmentActivity) {
        super(fragmentActivity);
    }

    public FragmentAdapter(@NonNull Fragment fragment) {
        super(fragment);
    }

    public FragmentAdapter(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle) {
        super(fragmentManager, lifecycle);
    }

    public void setFragmentList(List<Fragment> fragmentList) {
        this.fragmentList = fragmentList;
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        if (fragmentList == null || fragmentList.size() <= position) {
            return new Fragment();
        }
        return fragmentList.get(position);
    }

    @Override
```

```
public int getItemCount() {
    return fragmentList == null ? 0 : fragmentList.size();
}
```

步骤三：构建核心 Fragment

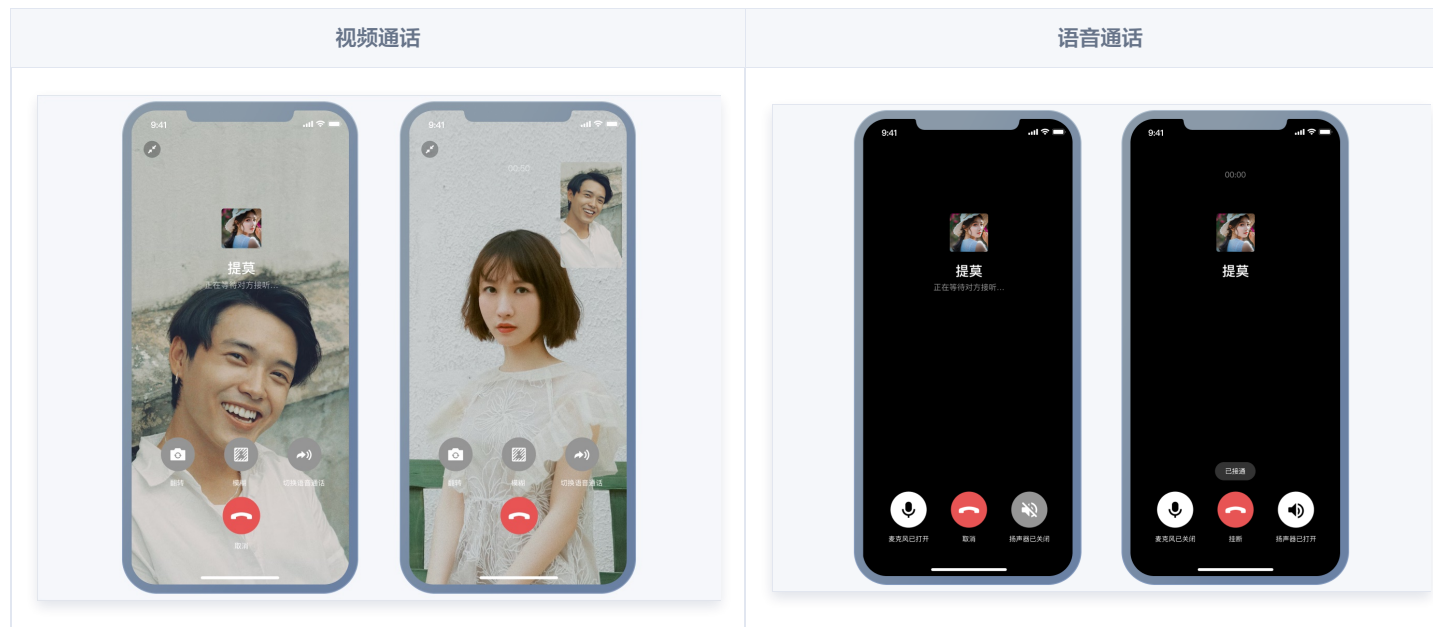
会话列表 TUIConversationFragment 以及联系人列表 TUIContactFragment 界面数据的获取、同步、展示以及交互均已在组件内部封装，UI 的使用与 Android 的普通 Fragment 一样方便。

在 MainActivity.java 的 onCreate 方法中添加：

```
List<Fragment> fragments = new ArrayList<>();
// tuiconversation 组件提供的会话界面
fragments.add(new TUIConversationFragment());
// tuicontact 组件提供的联系人界面
fragments.add(new TUIContactFragment());
ViewPager2 mainViewPager = findViewById(R.id.view_pager);
FragmentAdapter fragmentAdapter = new FragmentAdapter(this);
fragmentAdapter.setFragmentList(fragments);
mainViewPager.setOffscreenPageLimit(2);
mainViewPager.setAdapter(fragmentAdapter);
mainViewPager.setCurrentItem(0, false);
```

步骤四：构建音视频通话功能

TUI 组件支持在聊天界面对用户发起音视频通话，仅需要简单几步就可以快速集成：



1. 开通音视频服务

1.1 创建项目。创建项目，选择类型及场景，开通相关服务。

1.1.1 登录 [腾讯云视立方控制台](#) > [项目管理](#)，单击创建项目开始创建。



1.1.2 选择项目类型。可选择创建新项目并输入项目名称，或关联已有项目。

1.1.3 选择接入场景。选择快速接入的音视频场景，分为聊天应用和音视频通话。

1 创建项目 > 2 准备开发

创建/关联项目

使用 SDK 快速接入需要您创建一个新的项目或者关联已有项目。

项目类型 ☒ 创建新项目 ☐ 关联已有项目

项目名称

名称仅用于管理项目，和接入资源无关，也不影响您接入 SDK 的流程。限数字、中英文和下划线，不能超过 15 个字符。

选择接入场景

请选择快速接入的音视频场景，在对应的场景图中展示了极速接入项目提供的相关功能。

☒ 聊天应用（类微信 App）

包含搜索、会话、聊天、关系链、群组等等功能



详细功能 [展开](#)

☐ 音视频通话（类 FaceTime 通话）

包含语音通话、视频通话、邀请聊天等等功能



1.1.4 选择集成方式。选择含 UI 快速集成。

1.1.5 开通相关服务。使用音视频通话 SDK 需提前开通即时通信 IM 和实时音视频 TRTC 服务，仅开通服务，不涉及付费购买产品。

版权所有：腾讯云计算（北京）有限责任公司

第18 共40页

1.1.6 单击创建项目并下一步，完成项目创建，进入开发指引。

详细功能 [展开](#) ▶

● 选择集成方式

☒ 含 UI 快速集成 推荐
提供视频通话、房间管理等多个音视频标准组件 + 标准 SDK + 标准 UI 界面 + 完整的接入指引。仅需 1 天即可快速搭建完成。


☐ 不含 UI 集成
仅提供 TRTC SDK 和 IM SDK。UI 界面和高级功能（邀请聊天、房间管理等）需用户调用 API 接口自行实现。

● 开通相关服务

此处仅为您开通对应云产品服务，不涉及付费购买产品。开通后您可正常使用相关云产品控制台和功能服务。已开通产品服务不会重复开通。

即时通信 IM 【已开通】 ✓
了解即时通信 IM 产品，[点击查看更多详情](#)

实时音视频 TRTC 服务 【已开通】 ✓
了解实时音视频 TRTC 产品，[点击查看更多详情](#)

创建项目并下一步

1.2 准备开发。

1.2.1 体验 Demo。您可提前体验 Demo 了解 SDK 应用效果，包括 Android、iOS 和 Web 端。


1.2.2 关联/创建 IM 应用。音视频通话创建需要使用 IM 服务，请选择一个 IM 应用进行关联或者创建新的应用。

创建项目


2 准备开发

体验 DEMO

Android



iOS



Web

点击体验

关联/创建 IM 应用

音视频通话场景需要使用 IM 服务，请选择一个 IM 应用进行关联或者创建新的应用。

应用类型 ☒ 关联已有 IM 应用 ☐ 创建 IM 应用

应用名称

请选择

1.2.3 领取资源。每个应用可免费体验音视频通话 SDK 功能一次，有效期7天，音视频通话 SDK 体验版详细能力支持请参见 [套餐包功能说明](#)。业务上线请升级购买正式版本套餐，享受完整功能。

领取资源

含 UI 快速集成方式需使用相关资源，开通项目即赠送音视频通话-体验版（有效期 7 天），有效期内使用体验版进行 SDK 集成测试。业务上线请升级至正式版本套餐，享受完整功能。

音视频通话-体验版 有效期 7 天，限领一次

套餐包内容

用户上限	100 个	群组上限	100 个
直播群数上限	10 个	免费消息存储时长	7 天
TRTC 通用时长包	1000 分钟	通话状态展示	包含
通话呼叫通知	✓	多人通话	✓
全员推送	✓	创建社群	✓
AI 降噪	✓	视频人像清晰	✓
户外通话卡顿优化	✓	音频低码流畅	✓

前往集成

1.2.4 单击[前往集成](#)。

1.3 集成指南

1.3.1 选择集成环境并下载 TUIKit 开发包。集成环境包括 Android 和 iOS。

1.3.2 集成测试。下载 TUIKit 开发包后，参照集成测试的步骤查看相关集成文档完成集成测试。

1.3.3 正式开发。完成 SDK 接入测试后，若需正式开发并上线音视频应用，可购买音视频通话套餐包，包括基础版、进阶版和尊享版，升级当前项目业务版本，享受完整功能。

集成指南

选择集成环境并下载 TUIKit 开发包

Android SDK 开发包

Github 下载 集成文档

iOS SDK 开发包

Github 下载 集成文档

Web SDK 开发包

Github 下载 集成文档

集成效果可以参考体验 Demo 展开

集成测试

点击以下步骤即可查看相关集成文档

1 参考 源码集成 文档

2 参考 组件登录 文档

3 参考 构建会话列表 文档

4 参考 构建聊天界面 文档

5 参考 集成音视频通话能力 文档

6 参考 设置界面风格 文档

7 参考 添加自定义消息 文档

8 参考 实现本地搜索 文档

9 参考 接入离线推送 文档

1 源码集成

2 组件登录

3 构建会话列表

4 设置界面风格

5 构建聊天界面

6 集成音视频通话能力

7 添加自定义消息

8 实现本地搜索

9 接入离线推送

正式开发

参照上述步骤，您已成功完成含 UI 的集成测试。正式开发并上线音视频应用请点击下方购买按钮，购买音视频通话 - 基础版/进阶版/尊享版，享受完整功能。音视频通话 - 体验版到期后，标准音视频组件和标准 UI 将无法使用，多人通话、多端登录等高级特性也将失效。

	音视频通话（基础版）	音视频通话（进阶版）	音视频通话（尊享版）
	1499 元/月 立即购买	1999 元/月 立即购买	4499 元/月 立即购买
赠送通话时长	11 万分钟	23.5 万分钟	38 万分钟
微信同款 UI 设计	✓	✓	✓
通话呼叫/接听/拒绝/挂断	✓	✓	✓
多人通话	×	✓	✓
中途呼叫/加入三方通话	×	✓	✓
视频通话切换语音通话	✓	✓	✓
通话状态展示	✓	✓	✓
通话呼叫通知	✓	✓	✓
通话悬浮窗	✓	✓	✓
自定义呼叫铃声	✓	✓	✓
AI 降噪	×	✓	✓
户外通话卡顿优化	×	✓	✓
多端登录通话	×	×	✓
同平台多端登录通话	×	×	✓
全员推送	×	×	✓
同平台多设备在线	×	×	✓

2. 集成 TUICallKit 组件

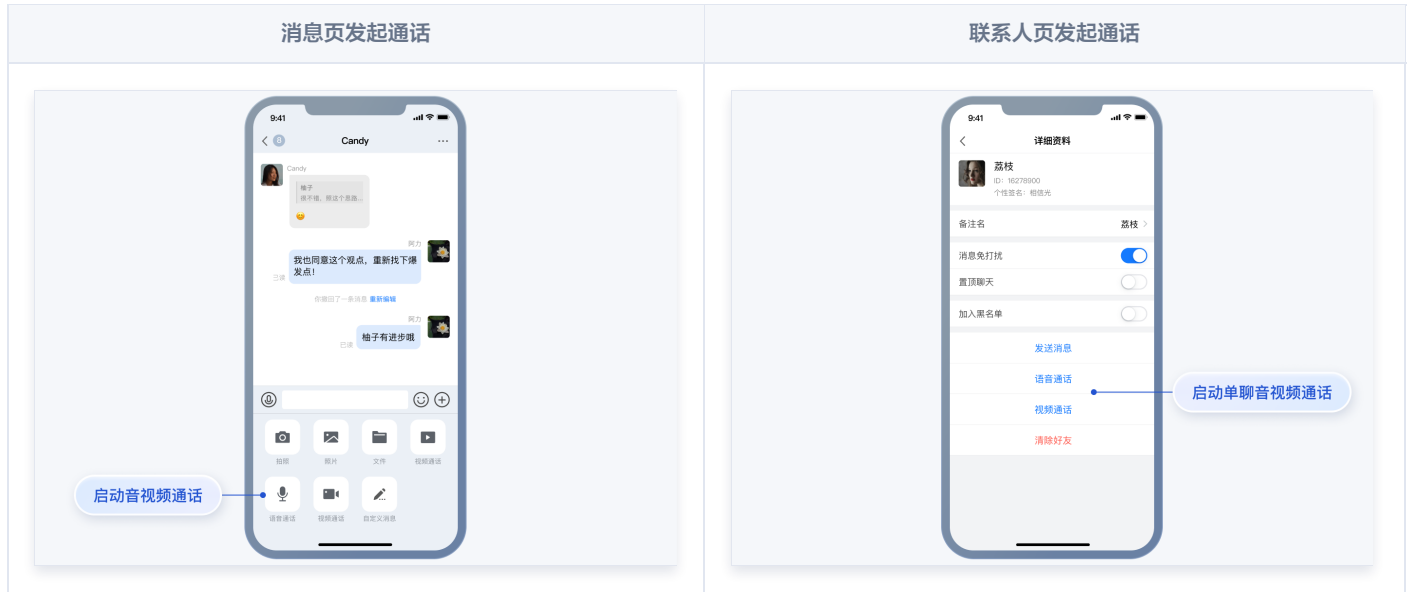
在 APP 的 build.gradle 文件中添加对 tuicallkit 的依赖：

版权所有：腾讯云计算（北京）有限责任公司

第21 共40页

```
api project(':tuicallkit')
```

3. 发起和接收视频或语音通话



- 集成 TUICallKit 组件后，聊天界面和联系人资料界面默认会显示“视频通话”和“语音通话”两个按钮，当用户单击按钮时，TUIKit 会自动展示通话邀请 UI，并给对方发起通话邀请请求。
- 当用户在线并且应用在前台时收到通话邀请时，TUIKit 会自动展示通话接收 UI，用户可以选择同意或则拒绝通话。
- 当用户离线收到通话邀请时，如需唤起 App 通话，需要使用离线推送能力。离线推送的实现请参考下一步。

4. 添加离线推送：实现音视频通话的离线推送，请参考以下几个步骤：

- 4.1 配置 App 的 [离线推送](#)。
- 4.2 集成 TUICallKit 组件。
- 4.3 通过 TUICallKit 发起通话邀请的时候，默认会生成一条离线推送消息。

❗ 说明：

更多实操教学视频请参见：[极速集成 TUIKit（Android）](#)。

常见问题

1、提示 "Manifest merger failed : Attribute application@allowBackup value=(true) from AndroidManifest.xml" 如何处理？

IM SDK 中默认 `allowBackup` 的值为 `false`，表示关闭应用的备份和恢复功能。

您可以在您的 `AndroidManifest.xml` 文件中删除 `allowBackup` 属性，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。

例如：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.tencent.qcloud.tuikit.myapplication">

    <application
        android:allowBackup="true"
        android:name=".MApplication"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
```

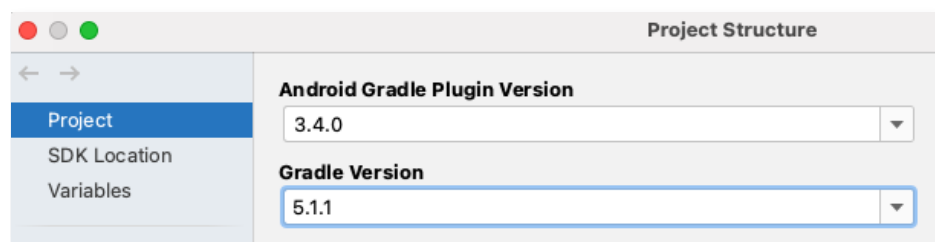
```
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:replace="android:allowBackup">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

2、提示 "NDK at /Users/***/Library/Android/sdk/ndk-bundle did not have a source.properties file" 如何处理？

出现此问题可能是您使用了较高版本的 Gradle 和 Gradle 插件，您可以使用推荐的版本：



此外，您也可以继续使用您当前版本的 Gradle，只需要在 local.properties 文件中加入您的 NDK 路径，例如：

```
ndk.dir=/Users/***/Library/Android/sdk/ndk/16.1.4479499。
```

3、提示 "Cannot fit requested classes in a single dex file" 如何处理？

出现此问题可能是您的 API 级别设置比较低，需要在 App 的 build.gradle 文件中开启 MultiDex 支持，添加 multiDexEnabled true 和对应依赖：

```
android {
    defaultConfig {
        ...
        minSdkVersion 15
        targetSdkVersion 28
        multiDexEnabled true
    }
    ...
}

dependencies {
    implementation "androidx.multidex:multidex:2.0.1"
}
```

同时，在您的 Application 文件中添加以下代码：

```
public class MyApplication extends SomeOtherApplication {
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        MultiDex.install(this);
    }
}
```

交流与反馈

欢迎加入 QQ 群进行技术交流和反馈问题。



Web & H5

最近更新时间：2023-09-11 17:12:42

开发环境要求

- Vue3
- TypeScript
- sass (sass-loader 版本 <= 10.1.1)

TUIKit 源码集成

步骤1：创建项目

使用 vue-cli 创建项目，配置Vue3 + TypeScript +sass。

```
Vue CLI v4.5.0

New version available 4.5.0 → 5.0.8
Run npm i -g @vue/cli to update!

? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel, TS, CSS Pre-processors, Linter
? Choose a version of Vue.js that you want to start the project with 3.x (Preview)
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS (with dart-sass)
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? No
```

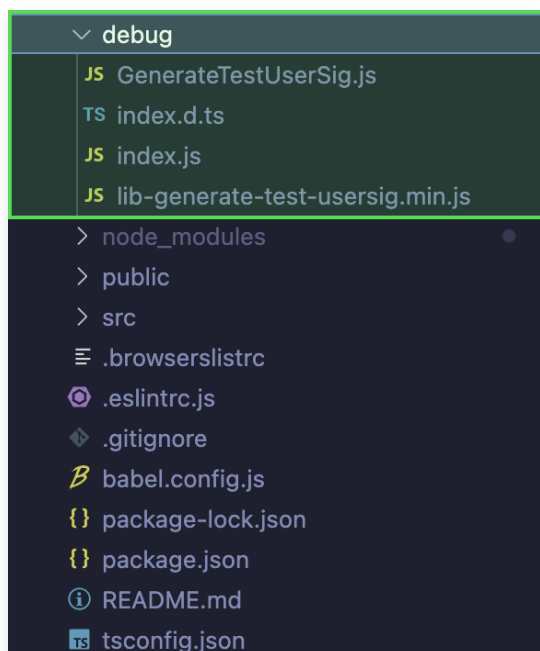
步骤2：下载 TUIKit 组件

从 [GitHub 下载](#) TUIKit源码。复制 TUIKit 文件夹放置到自己到工程的 src 文件夹中，例如：

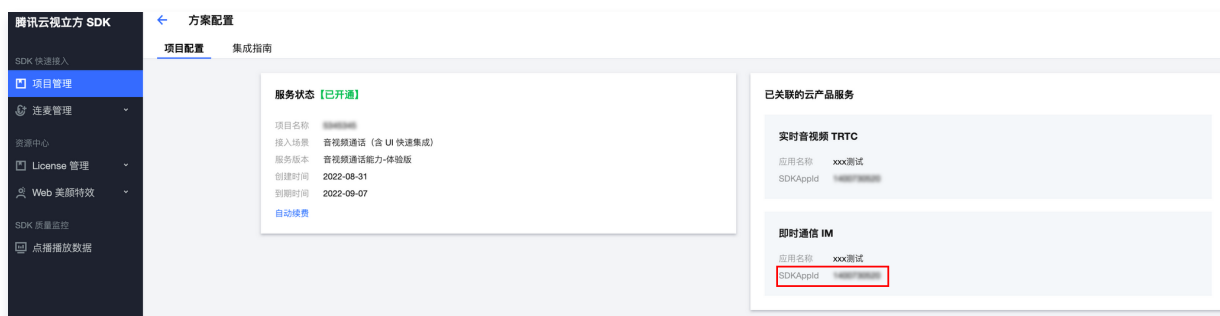
```
> node_modules
> public
└─ src
   └─ assets
   └─ components
   └─ TUIKit
   └─ App.vue
   └─ TS main.ts
   └─ TS shims-vue.d.ts
   └─ .browserslistrc
   └─ .eslintrc.js
   └─ .gitignore
   └─ babel.config.js
   └─ package-lock.json
   └─ package.json
   └─ README.md
   └─ tsconfig.json
```

步骤3：生成 UserSig

1. 从 [GitHub 下载](#) GenerateTestUserSig 工具包，并复制到项目中，例如：



2. 设置 GenerateTestUserSig 文件中的相关参数，其中 SDKAppID 和密钥等信息，可通过 [腾讯云视立方控制台](#) > [项目管理](#) 获取，单击目标项目右侧项目配置，进入配置页面。



3. 在基本信息区域，单击显示密钥，复制并保存密钥信息至 `GenerateTestUserSig` 文件。

```
JS GenerateTestUserSig.js x
test > quick-integration > debug > JS GenerateTestUserSig.js > ...
3  /**
4   * Tencent Cloud SDKAppId, which should be replaced with user's SDKAppId.
5   * Enter Tencent Cloud TRTC [Console] (https://console.cloud.tencent.com/trtc) to create an application,
6   * and you will see the SDKAppId.
7   * It is a unique identifier used by Tencent Cloud to identify users.
8   *
9   * 腾讯云 SDKAppId, 需要替换为您自己账号下的 SDKAppId。
10  * 进入腾讯云实时音视频[控制台] (https://console.cloud.tencent.com/rav) 创建应用, 即可看到 SDKAppId,
11  * 它是腾讯云用于区分客户的唯一标识。
12  */
13  const SDKAPPID = 0; 替换成 即时通信 IM 控制台 - 基本配置 - 应用资料 - SDKAppID
14
15
16  /**
17   * Signature expiration time, which should not be too short
18   * Time unit: second
19   * Default time: 7 * 24 * 60 * 60 = 604800 = 7days
20   *
21   * 签名过期时间, 建议不要设置的过短
22   * 时间单位: 秒
23   * 默认时间: 7 * 24 * 60 * 60 = 604800 = 7 天
24   */
25  const EXPIRETIME = 604800;
26
27  /**
28   * Encryption key for calculating signature, which can be obtained in the following steps:
29   *
30   * Step1. Enter Tencent Cloud TRTC [Console] (https://console.cloud.tencent.com/rav),
31   * and create an application if you don't have one.
32   * Step2. Click your application to find "Quick Start".
33   * Step3. Click "View Secret Key" to see the encryption key for calculating UserSig,
34   * and copy it to the following variable.
35   *
36   * Notes: this method is only applicable for debugging Demo. Before official launch,
37   * please migrate the UserSig calculation code and key to your backend server to avoid
38   * unauthorized traffic use caused by the leakage of encryption key.
39   * Document: https://intl.cloud.tencent.com/document/product/647/35166#Server
40   *
41   * 计算签名用的加密密钥, 获取步骤如下:
42   *
43   * step1. 进入腾讯云实时音视频[控制台] (https://console.cloud.tencent.com/rav), 如果还没有应用就创建一个,
44   * step2. 单击“应用配置”进入基础配置页面, 并进一步找到“帐号体系集成”部分。
45   * step3. 点击“查看密钥”按钮, 就可以看到计算 UserSig 使用的加密的密钥了, 请将其拷贝并复制到如下的变量中
46   *
47   * 注意: 该方案仅适用于调试Demo, 正式上线前请将 UserSig 计算代码和密钥迁移到您的后台服务器上, 以避免加密密钥泄露导致的流量盗用。
48   * 文档: https://cloud.tencent.com/document/product/647/17275#Server
49   */
50  const SECRETKEY = ''; 替换成 即时通信 IM 控制台 - 基本配置 - 基础信息 - 密钥
51
```

⚠ 注意:

本文提到的获取 UserSig 的方案是在客户端代码中配置 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。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 App 的接口, 在需要 UserSig 时由您的 App 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](#)。

步骤4: 下载 TUIKit 组件依赖

```
cd src/TUIKit
npm i --legacy-peer-deps
```

步骤5: 引入 TUIKit 组件

在 main.ts 中，引入 TUIKit，并注册到 vue 项目实例中：

```
import { createApp } from 'vue'
import App from './App.vue'
import { TUICore, TUIComponents } from './TUIKit';
import { genTestUserSig } from '../debug';
const config = {
  SDKAppID: 0, // Replace 0 with the SDKAppID of your IM application when connecting. Value type:
  Number
};
// init TUIKit
const TUIKit = TUICore.init(config);
// TUIKit add TUIComponents
TUIKit.use(TUIComponents);
const userID = 'xxxx'; // User ID
const userInfo = {
  userID: userID,
  userSig: genTestUserSig(userID).userSig, // The password with which the user logs in to IM. It
  is the ciphertext generated by encrypting information such as userID. For the detailed generation
  method, see Generating UserSig
};
// login TUIKit
TUIKit.login(userInfo);
// register
createApp(App).use(TUIKit).mount('#app')
```

⚠ 注意：

SDKAppID 需与 GenerateTestUserSig 文件中 SDKAppID 一致。

步骤6：调用 TUIKit 组件

在需要展示的页面，调用 TUIKit 的组件即可使用。

例如：在 App.vue 页面中，使用 TUIConversation、TUIChat 搭建聊天界面。

```
<template>
  <div class="home-TUIKit-main">
    <div class="conversation">
      <TUIConversation />
    </div>
    <div class="chat">
      <TUIChat>
        <h1>欢迎使用腾讯云即时通信IM</h1>
      </TUIChat>
    </div>
  </div>
</template>

<style scoped>
.home-TUIKit-main {
  display: flex;
  height: 800px;
}
.conversation {
  min-width: 285px;
  flex: 0 0 24%;
}
```



```
border-right: 1px solid #f4f5f9;
}
.chat {
  flex: 1;
  height: 100%;
  position: relative;
}
</style>
```

步骤7：启动项目

```
npm run serve
```

常见问题

1. 如何生成 UserSig?

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](#)。

2. 提示 Module not found: Error: Can't resolve 'sass-loader'?

- IM TUIKit web 样式依赖 sass，需在项目全局安装 sass 和 sass-loader。
- 其中 sass-loader 的版本 $\leq 10.1.1$ 。

```
npm install sass sass-loader@10.1.1 --save-dev
```

小程序

最近更新时间：2023-09-11 17:12:42

如何集成 TUIKit？

步骤1：下载源码

TUIKit 支持以原生 js 的方式集成。可从 Github 下载 TUIKit 源码。命令行执行：

```
git clone https://github.com/tencentyun/TIMSDK.git
```

步骤2：初始化 TUIKit

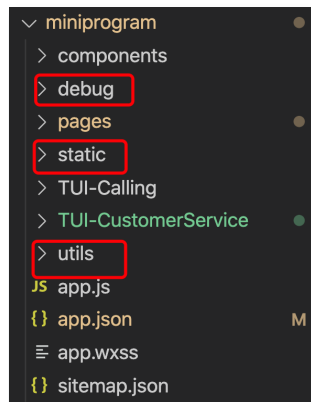
```
cd TIMSDK/MiniProgram/TUIKit
```

找到并打开 TUIKit/miniprogram/debug/GenerateTestUserSig.js 文件，并填写 SDKAppID 以及 SECRETKEY (默认为空字符串，请设置为实际的密钥信息)。

```
import LibGenerateTestUserSig from './lib-generate-test-usersig-es.min.js';  
const SDKAPPID = 0;  
const SECRETKEY = "";
```

步骤3：集成静态资源文件

在自己的项目中集成静态资源文件（工具、图片等）。

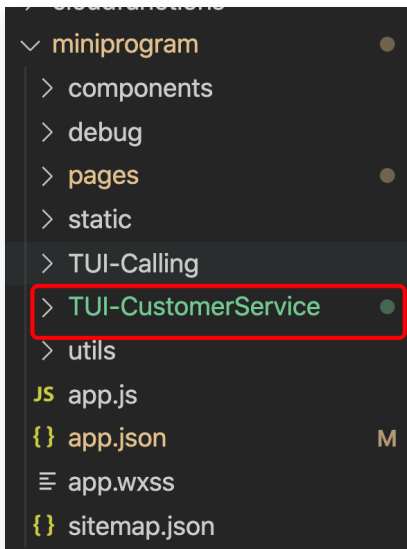


步骤4：集成所需模块

由于微信对于小程序的主包体积要求不得超过2M，所以 TUIKit 推出分包的解决方案，为客户解决体积超出这一问题。

集成整个分包

1. 客户自己设置主包的内容并放置在 page 文件夹内，建议主包尽可能的只放首页，减少首屏渲染时间以及主包的体积大小。
2. 将分包引入，并和主包置于同一层级。分包内部独立并包含整个模块的逻辑。



- 在 app.json 中配置分包路径。主包路径为原来的 page 路径不变。

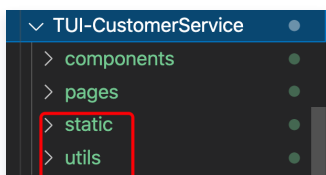
```
"subPackages": [
  {
    "root": "TUI-CustomerService",
    "name": "TUI-CustomerService",
    "pages": [
      "pages/TUI-Conversation/conversation/conversation",
      "pages/TUI-Chat/chat",
      "pages/TUI-Conversation/create-conversation/create",
      "pages/TUI-Group/create-group/create",
      "pages/TUI-Group/join-group/join",
      "pages/TUI-Group/memberprofile-group/memberprofile"
    ],
    "independent": false
  }
],
```

- 是否启用预加载（启用后，在进入主包后就会预加载分包内的资源，否则，进入分包后才会加载分包内的资源）。

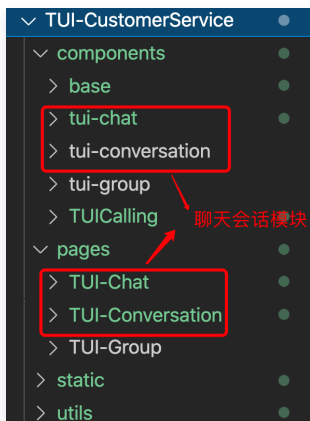
```
"preloadRule": {
  "pages/TUI-Index/index" : {
    "network": "all",
    "packages": [
      "TUI-CustomerService"
    ]
  }
},
```

只集成目标模块

1. 客户自己设置主包的内容并放置在 page 文件夹内，建议主包尽可能的只放首页，减少首屏渲染时间以及主包的体积大小。
2. 将引入的模块设置为分包，并和主包置于同一层级。分包内部独立并包含整个模块的逻辑。
 - 引入模块所需静态资源



- 引入自己所需的模块



- 在 app.json 中配置分包路径。主包路径为原来的 page 路径不变。

```
"subPackages": [
  {
    "root": "TUI-CustomerService",
    "name": "TUI-CustomerService",
    "pages": [
      "pages/TUI-Conversation/conversation/conversation",
      "pages/TUI-Chat/chat",
      "pages/TUI-Conversation/create-conversation/create"
    ],
    "independent": false
  }
],
```

- 是否启用预加载（启用后，在进入主包后就会预加载分包内的资源，否则，进入分包后才会加载分包内的资源）。

```
"preloadRule": {
  "pages/TUI-Index/index" : {
    "network": "all",
    "packages": [
      "TUI-CustomerService"
    ]
  }
},
```

常见问题

小程序如果需要上线或者部署正式环境怎么办？

请在微信公众平台 > 开发 > 开发设置 > 服务器域名中进行域名配置：

- 将以下域名添加到 **request 合法域名**：
 - 从v2.11.2起 SDK 支持了 WebSocket，WebSocket 版本须添加以下域名：

域名	说明	是否必须
wss://wss.im.qqcloud.com	Web IM 业务域名	必须
wss://wss.tim.qq.com	Web IM 业务域名	必须
https://web.sdk.qqcloud.com	Web IM 业务域名	必须
https://webim.tim.qq.com	Web IM 业务域名	必须

- v2.10.2及以下版本使用 HTTP，HTTP 版本须添加以下域名：

域名	说明	是否必须
https://webim.tim.qq.com	Web IM 业务域名	必须
https://yun.tim.qq.com	Web IM 业务域名	必须
https://events.tim.qq.com	Web IM 业务域名	必须
https://grouptalk.c2c.qq.com	Web IM 业务域名	必须
https://pingtas.qq.com	Web IM 统计域名	必须
https://aegis.qq.com	Web IM 统计域名	必须

- 将以下域名添加到 **uploadFile 合法域名**：

域名	说明	是否必须
https://cos.ap-shanghai.myqcloud.com	文件上传域名	必须

- 将以下域名添加到 **downloadFile 合法域名**：

域名	说明	是否必须
https://cos.ap-shanghai.myqcloud.com	文件下载域名	必须

相关文档

- [SDK API 手册](#)
- [SDK 更新日志](#)

uni-app

最近更新时间：2024-11-25 15:41:12

uni-app TUIKit 简介

uni-app TUIKit 是基于 IM SDK 实现的一套 UI 组件，其包含会话、聊天、群组管理等功能，基于 UI 组件您可以像搭积木一样快速搭建起自己的业务逻辑。

目前我们提供了示例客服群、示例好友的基础模板，在线客服功能包括：

- 支持发送文本消息、图片消息、语音消息、视频消息等常见消息。
- 支持双人语音、视频通话功能。
- 支持常用语、订单、服务评价等自定义消息。
- 支持创建群聊会话、群成员管理等。



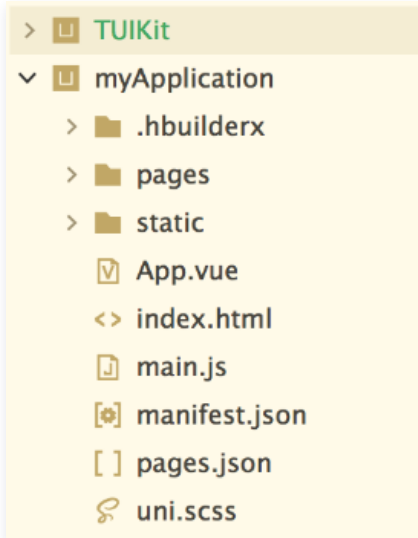
uni-app TUIKit 支持平台

- Android
- iOS
- 微信小程序

集成 TUIKit

步骤1：安装依赖

1. uni-app TUIKit 支持源码集成，下载 [uni-app TUIKit 源码](#)。将 TUIKit 文件夹与自己的工程文件夹置于同级，例如：



2. 根据 package.json 进行对应依赖安装。

```
1 {
2   "name": "云通信 IM",
3   "version": "1.0.0",
4   "description": "",
5   "main": "main.js",
6   "dependencies": {
7     "cos-wx-sdk-v5": "^1.0.10",
8     "tim-wx-sdk": "^2.16.1",
9   },
10  "devDependencies": {},
11  "scripts": {
12    "test": "echo \"Error: no test specified\" && exit 1"
13  },
14  "keywords": [],
15  "author": "",
16  "license": "ISC"
17 }
```

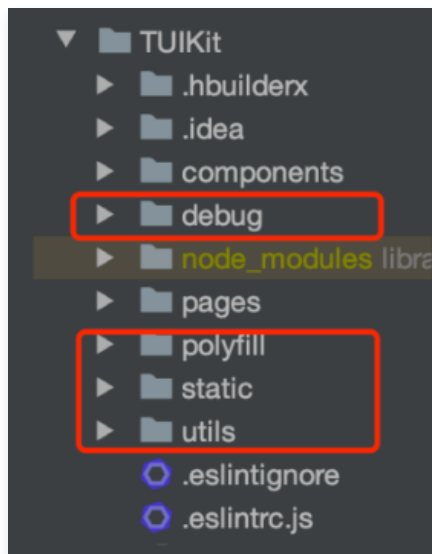
步骤2：初始化 TUIKit

将 app.vue 中的代码复制到 myApplication 项目中，填写 SDKAppID。

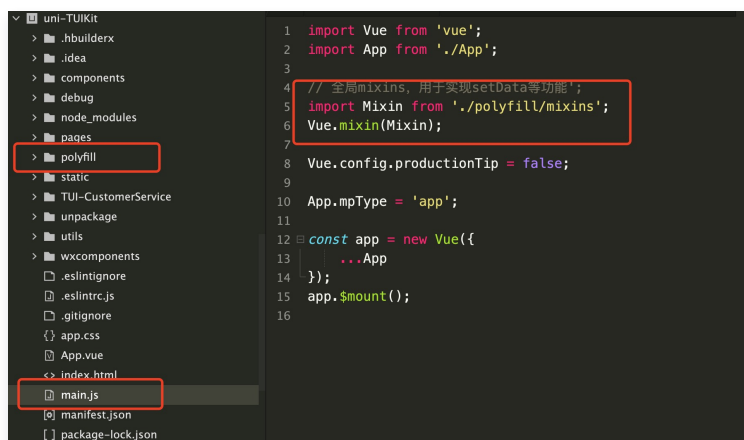
```
onLaunch() {
  uni.setStorageSync('islogin', false);
  // 重点注意：为了 uni-app 更好地接入使用 tim，快速定位和解决问题，请勿修改 uni.$TUIKit 命名
  // 如果您已经接入 tim，请将 uni.tim 修改为 uni.$TUIKit。
  uni.$TUIKit = TIM.create({
    SDKAppID: 0
  });
  uni.$TUIKit.registerPlugin({
    'cos-wx-sdk': COS
  });
  uni.$TUIKitTIM = TIM;
  uni.$TUIKitEvent = TIM.EVENT;
  uni.$TUIKitVersion = TIM.VERSION;
  uni.$TUIKitTypes = TIM.TYPES; // 监听系统级事件
  uni.$resetLoginData = this.resetLoginData();
  uni.$TUIKit.on(uni.$TUIKitEvent.SDK_NOT_READY, this.onSdkNotReady);
  uni.$TUIKit.on(uni.$TUIKitEvent.KICKED_OUT, this.onKickedOut);
  uni.$TUIKit.on(uni.$TUIKitEvent.ERROR, this.onTIMError);
  uni.$TUIKit.on(uni.$TUIKitEvent.NET_STATE_CHANGE, this.onNetStateChange);
  uni.$TUIKit.on(uni.$TUIKitEvent.SDK_RELOAD, this.onSDKReload);
}
```

步骤3：集成静态资源文件

1. 在 myApplication 项目中集成静态资源文件（工具、图片等）。

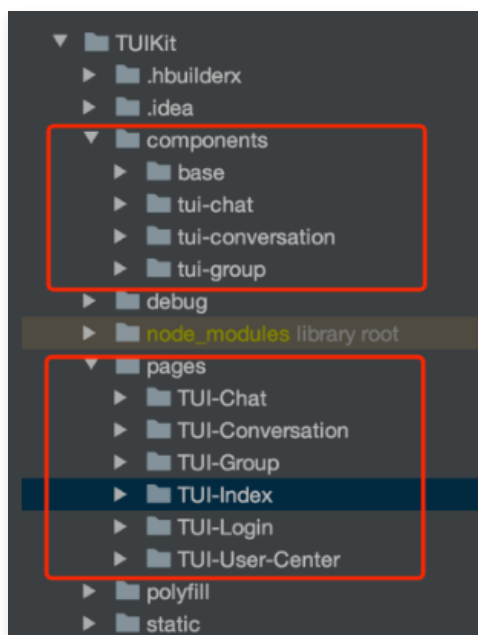


2. 在 myApplication 引入 mixins，用于实现 setData 等功能。

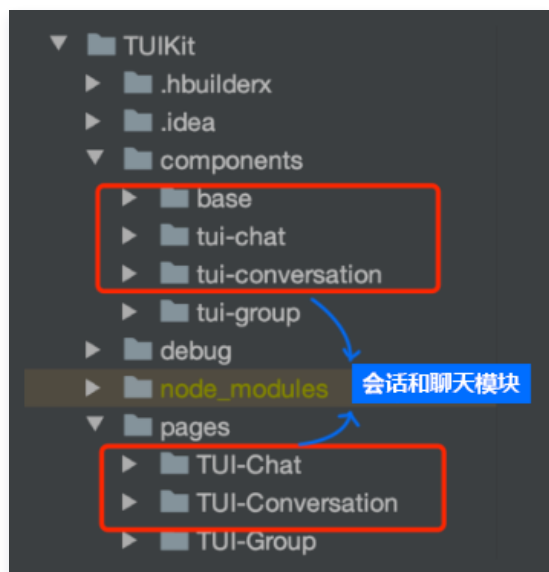


步骤4：集成所需模块

1. 将 pages 和 components 复制到 myApplication 项目中。



2. 也可以只集成自己所需要的模块，将 pages 和其对应的 components 复制到 myApplication 项目目录下。



步骤5：更新路由

根据页面更新路由：更新 pages.json 中的 pages 路由。



步骤6：获取签名和登录

⚠ 注意：

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 App 的接口，在需要 UserSig 时由您的 App 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](#)。

```
uni.$TUIKit.login({userID: 'your userID', userSig: 'your userSig'})
.then(function(imResponse) {
  console.log(imResponse.data); // 登录成功
  if (imResponse.data.repeatLogin === true) {
    // 标识账号已登录，本次登录操作为重复登录。v2.5.1 起支持
    console.log(imResponse.data.errorInfo);
  }
})
.catch(function(imError) {
  console.warn('login error:', imError); // 登录失败的相关信息
```

```
});
```

步骤7：开启音视频通话

- 打包 App 集成，请参见原生音视频插件接入 [原生音视频插件](#)。
- 打包小程序集成，请参见小程序音视频插件接入 [腾讯云小程序音视频插件](#)。

常见问题

1. uni-app 同时支持 Android，iOS，微信小程序平台，IM SDK 如何选择？

请选择 `tim-wx-sdk`，npm 安装或者静态引入：

```
// 从v2.11.2起，SDK 支持了 WebSocket，推荐接入；v2.10.2及以下版本，使用 HTTP
npm install tim-wx-sdk@latest --save
import TIM from 'tim-wx-sdk';
// 创建 SDK 实例，`TIM.create()`方法对于同一个 `SDKAppID` 只会返回同一份实例
uni.$TUIKit = TIM.create({
  SDKAppID: 0 // 接入时需要将0替换为您的即时通信 IM 应用的 SDKAppID
});
// 设置 SDK 日志输出级别，详细分级请参见 setLogLevel 接口的说明
uni.$TUIKit.setLogLevel(0); // 普通级别，日志量较多，接入时建议使用
// uni.$TUIKit.setLogLevel(1); // release 级别，SDK 输出关键信息，生产环境时建议使用
```

如果您的项目需要关系链功能，请使用 `tim-wx-friendship.js`：

```
import TIM from 'tim-wx-sdk/tim-wx-friendship.js';
```

❗ 说明：

为了 uni-app 更好地接入使用 tim，快速定位和解决问题，请勿修改 uni.\$TUIKit 命名，如果您已经接入 tim，请将 uni.tim 修改为 uni.\$TUIKit。

- 请将 IM SDK 升级到 [2.15.0](#)，该版本支持了 iOS 语音播放。
- 若同步依赖过程中出现问题，请切换 npm 源后再次重试。
- 切换 cnpm 源：

```
npm config set registry http://r.cnpmjs.org/
```

2. 如何上传图片、视频、语音消息等富媒体消息？

请使用 `cos-wx-sdk-v5`：

```
// 发送图片、语音、视频等消息需要 cos-wx-sdk-v5 上传插件
npm install cos-wx-sdk-v5@0.7.11 --save
import COS from 'cos-wx-sdk-v5';
// 注册 COS SDK 插件
uni.$TUIKit.registerPlugin({
  'cos-wx-sdk': COS
});
```

3. uni-app 打包 iOS 语音消息无法播放怎么办？

请将 IM SDK 升级到 [2.15.0](#)，该版本支持了 iOS 语音消息播放。

4. uni-app 打包 app 发送语音消息时间显示错误怎么办？

uni-app 打包 app，`recorderManager.onStop` 回调中没有 `duration` 和 `fileSize`，需要用户自己补充 `duration` 和 `fileSize`。

- 通过本地起定时器记录时间，计算出 `duration`。
 - 本地计算文件大小，`fileSize = (音频码率) x 时间长度(单位:秒) / 8`，粗略估算。
- 详细代码请参见 [uni-app TUIKit](#)。

⚠ 注意：

语音消息对象中必须包括 `duration` 和 `fileSize`，如果没有 `fileSize`，语音消息时长是一串错误的数字。

5. video 视频消息层级过高无法滑动怎么办？

在项目中通过视频图片代替，没有直接渲染 `video`，在播放时渲染的方式规避了层级过高问题。

- 详细代码请参见 [uni-app TUIKit](#)。
- 请参见官方 [原生组件说明](#)。

6. 微信小程序环境，真机预览，报系统错误，体积过大怎么办？

运行时请勾选代码压缩，运行到小程序模拟器>运行时是否压缩代码。

7. 引入原生音视频插件报以下错怎么办？

- 报错截图：

```
09 TIM 09:46:50.460:ModuleManager._startChainModule
09 TIM 09:46:50.463:NetMonitorModule.start networkType:wifi __INFO
09 TIM 09:46:50.465:SignInModule.login userID:ML
21 [Vue warn]: Error in on handler: "TypeError: undefined is not an object (evaluating 'uni.$TUICalling.login')"
```

- 解决方法：根据 uni-app [原生插件调试](#) 制作 [自定义基座](#)



8. 微信小程序如果需要上线或者部署正式环境怎么办？

请在微信公众平台>开发>开发设置>服务器域名中进行域名配置：

- 将以下域名添加到 `request 合法域名`：
 - 从v2.11.2起 SDK 支持了 WebSocket，WebSocket 版本须添加以下域名：

域名	说明	是否必须
wss://wss.im.qqcloud.com	Web IM 业务域名	必须
wss://wss.tim.qq.com	Web IM 业务域名	必须
https://web.sdk.qqcloud.com	Web IM 业务域名	必须
https://webim.tim.qq.com	Web IM 业务域名	必须

- v2.10.2及以下版本使用 HTTP，HTTP 版本须添加以下域名：

域名	说明	是否必须
https://webim.tim.qq.com	Web IM 业务域名	必须
https://yun.tim.qq.com	Web IM 业务域名	必须
https://events.tim.qq.com	Web IM 业务域名	必须
https://grouptalk.c2c.qq.com	Web IM 业务域名	必须
https://pingtas.qq.com	Web IM 统计域名	必须

- 将以下域名添加到 **uploadFile 合法域名**：

域名	说明	是否必须
https://cos.ap-shanghai.myqcloud.com	文件上传域名	必须

- 将以下域名添加到 **downloadFile 合法域名**：

域名	说明	是否必须
https://cos.ap-shanghai.myqcloud.com	文件下载域名	必须

9. uni-app 是否支持离线推送？

目前官方暂未提供 uni-app 离线推送方法。

推荐方案：

- 方案一：将消息通过 [第三方回调给您的服务器](#)，再使用 [UniPush](#) 完成离线推送。
- 方案二：使用第三方插件库相关的 [离线推送插件](#)（非官方插件）。

技术咨询

了解更多详情您可 QQ 咨询：309869925（技术交流群）

参考文档

- [SDK API 手册](#)
- [SDK 更新日志](#)
- [uni-app TUIKit 源码](#)
- [一分钟跑通 Demo \(uni-app\)](#)
- [快速集成微信小程序原生 TUIKit](#)
- [微信小程序原生 TUIKit 源码](#)