

即时通信 IM Demo 专区



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內容。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

Demo 专区

体验 Demo

快速跑通

概述

Android

iOS

Web & H5 Vue2/Vue3

小程序

uniapp vue2/vue3

React

React Native

Unity

UE

Flutter

HarmonyOS

Demo 专区

体验 Demo

最近更新时间: 2025-05-15 17:46:53

React Native / React Demo


 

React Native




快速入门


各端原生开发 Demo




Android




快速入门



iOS



快速入门



Web(Vue)

基础聊天

快速入门




H5(Vue)



快速入门



微信小程序



快速入门

uni-app Demo



Android



快速入门



iOS



快速入门



H5



快速入门











微信小程序







快速入门

Flutter Demo

 <p>手机端(经典风格)</p>  <p>快速入门</p>	 <p>手机端(简约风格)</p>  <p>快速入门</p>	 <p>H5</p>  <p>快速入门</p>	 <p>Web</p> <p>在线体验</p> <p>社群聊天</p> <p>快速入门</p>	 <p>桌面端</p> <p>Windows</p> <p>macOS</p> <p>mac 安装 快速入门</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------

Unity Demo

 <p>Android</p>  <p>快速入门</p>	 <p>iOS</p>  <p>快速入门</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

快速跑通

概述

最近更新时间：2024-12-26 12:42:52

请选择您的开发平台，快速运行腾讯云即时通信 IM Demo：

- [Android](#)
- [iOS](#)
- [Web & H5 Vue2/Vue3](#)
- [小程序](#)
- [uni-app Vue2/Vue3](#)
- [React](#)
- [React Native](#)
- [Unity](#)
- [UE](#)
- [Flutter](#)
- [用 Flutter 快速集成至您现有原生应用](#)
- [HarmonyOS](#)

Android

最近更新时间: 2025-04-30 10:26:42

本文主要介绍如何快速运行腾讯云即时通信 IM Demo (Android)。以下视频将帮助您快速入门:

[观看视频](#)

操作步骤

步骤1: 创建应用

1. 登录 [即时通信 IM 控制台](#)。如果您已有应用, 请记录其 SDKAppID 信息, 并直接跳到 [步骤2: 获取 SDKAppID 和密钥信息](#) 参见后续步骤进行操作。
2. 在应用管理页面, 单击创建新应用。



3. 输入应用名称、选择合适的数据中心, 单击确定即可完成应用的创建。

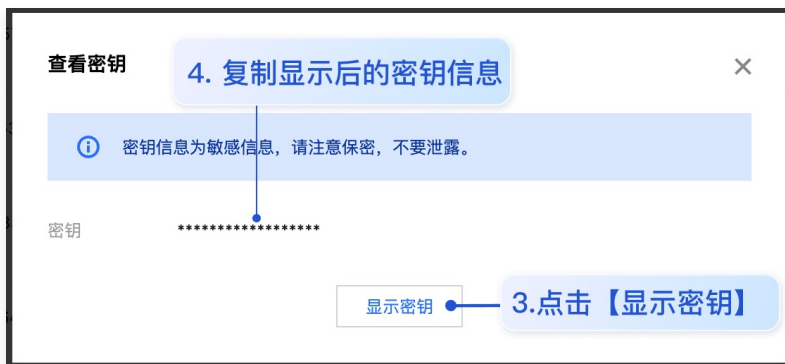


步骤2: 获取 SDKAppID 和密钥信息

1. 在应用管理页面的 SDKAppID 列获取 SDKAppID 信息。



2. 在操作列单击查看密钥, 随后在弹出的对话框中, 单击显示密钥, 复制显示后的密钥信息。



注意：

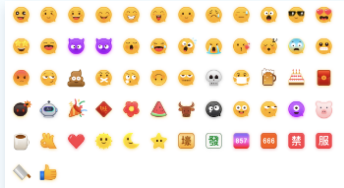
密钥信息为敏感信息，为防止他人盗用，请妥善保管，谨防泄露。

步骤3：下载并配置 Demo 源码

1. 下载即时通信 IM Demo 工程，具体下载地址请参见 [SDK 下载](#)。

说明：

为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



2. 打开终端目录的工程，找到对应的 `GenerateTestUserSig` 文件，路径为

`Android/Demo/app/src/main/java/com/tencent/qcloud/tim/demo/signature/GenerateTestUserSig.java`

3. 设置 `GenerateTestUserSig` 文件中的相关参数：

- **SDKAPPID**：请设置为 [步骤1](#) 中获取的实际应用 SDKAppID。
- **SECRETKEY**：请设置为 [步骤2](#) 中获取的实际密钥信息。

```

public class GenerateTestUserSig {
    /**
     * 腾讯云 SDKAppID，需要替换为您自己账号下的 SDKAppID。
     *
     * 进入腾讯云通信(控制台)(https://console.cloud.tencent.com/avc ) 创建应用，即可看到 SDKAppID。
     * 它是腾讯云用于区分客户端的唯一标识。
     */
    public static final int SDKAPPID = 0;

    /**
     * 签名过期时间，建议不要设置的过短
     * <code>
     * 时间单位：秒
     * 默认时间：7 x 24 x 60 x 60 = 604800 = 7 天
     */
    private static final int EXPIRETIME = 604800;

    /**
     * 计算签名用的加密密钥，获取步骤如下：
     *
     * step1. 进入腾讯云通信(控制台)(https://console.cloud.tencent.com/avc )，如果还没有应用就创建一个。
     * 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。
     * 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，请将拷贝并复制如下变量的值。
     *
     * 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。
     * 文档：https://cloud.tencent.com/document/product/269/32688#Server
     */
    private static final String SECRETKEY = "";
        
```

基本信息

SDKAppID	1400
状态	应用 ✔ 停用
应用名称	test
应用类型	视频

应用名称	test
应用类型	视频
密钥	72c9... 987ab

隐藏密钥 复制

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 Demo 和功能调试。

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：编译运行

用 Android Studio 导入工程直接编译运行即可。

更多详情可参见 [步骤3](#) 克隆的 Demo 工程中对应目录下的 `README.md` 文件。

开发环境要求

- Android Studio–Chipmunk
- Gradle–6.7.1
- Android Gradle Plugin Version–4.2.0
- kotlin–gradle–plugin–1.5.31

⚠ 注意:

Demo 默认集成了音视频通话功能，由于该功能依赖的音视频 SDK 暂不支持模拟器，请使用真机调试或者运行 Demo。

步骤5: 开通云端审核功能

在消息发送、资料修改等场景中，可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，还有可能违法并导致业务被监管部门查封。

即时通信 IM 支持 [云端审核](#)（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。您可直接前往 [云端审核控制台](#) 开通免费试用，每个应用可免费试用7天（含10000次免费抵扣量）。

交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

iOS

最近更新时间：2025-05-28 17:45:52

本文主要介绍如何快速运行腾讯云即时通信 IM Demo（iOS）。以下视频将帮助您快速入门：

[观看视频](#)

操作步骤

步骤1：创建应用

1. 登录 [即时通信 IM 控制台](#)。如果您已有应用，请记录其 SDKAppID 信息，并直接跳到 [步骤2：获取 SDKAppID 和密钥信息](#) 进行操作后续步骤。
2. 在应用管理页面，单击创建新应用。



3. 输入应用名称、选择合适的数据中心，单击确定即可完成应用的创建。

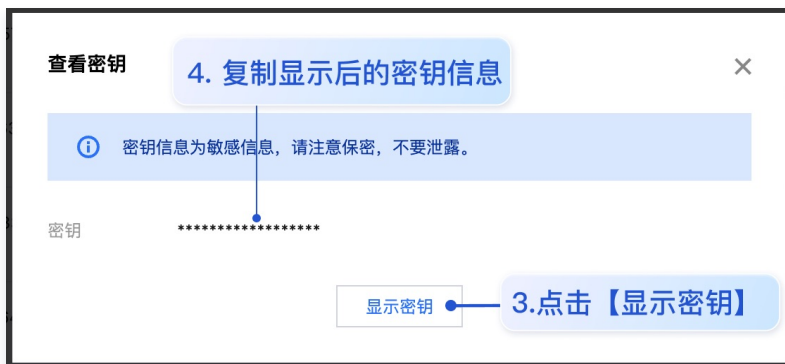


步骤2：获取 SDKAppID 和密钥信息

1. 在应用管理页面的 SDKAppID 列获取 SDKAppID 信息。



2. 在操作列单击查看密钥，随后在弹出的对话框中，单击显示密钥，复制显示后的密钥信息。



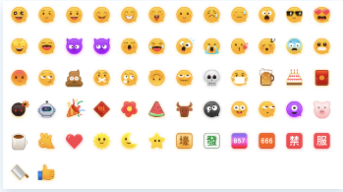
注意：

密钥信息为敏感信息，为防止他人盗用，请妥善保管，谨防泄露。

步骤3：下载并配置 Demo 源码

说明：

为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 **IM 企业版套餐** 免费使用该表情包。



下载 Swift Demo

1. 下载 **Swift Chat UIKit 工程**。
2. 打开所属终端目录的工程，找到对应的 `GenerateTestUserSig.swift` 文件。
iOS 路径：`Chat_Uikit/Swift/TUIKitDemo/TUIKitDemo/Private/GenerateTestUserSig.swift`
3. 设置 `GenerateTestUserSig.swift` 文件中的相关参数：
 - `public_SDKAPPID`：请设置为 **步骤1** 中获取的实际应用 SDKAppID。
 - `public_SECRETKEY`：请设置为 **步骤2** 中获取的实际密钥信息。

下载 Objective-C Demo

1. 下载即时通信 IM Demo 工程，具体下载地址请参见 **SDK 下载**。
2. 打开所属终端目录的工程，找到对应的 `GenerateTestUserSig.h` 文件。
iOS 路径：`iOS/Demo/TUIKitDemo/Private/GenerateTestUserSig.h`
Mac 路径：`Mac/Demo/TUIKitDemo/Debug/GenerateTestUserSig.h`
3. 设置 `GenerateTestUserSig.h` 文件中的相关参数：
 - `SDKAPPID`：请设置为 **步骤1** 中获取的实际应用 SDKAppID。
 - `SECRETKEY`：请设置为 **步骤2** 中获取的实际密钥信息。



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 Demo 和功能调试。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：编译运行

可参见 **步骤3** 克隆的 Demo 工程中对应目录下的 `README.md` 文件。

1. 终端执行以下命令，检查 pod 版本。

```
pod --version
```

若提示 pod 不存在，或 pod 版本小于 1.7.5，请执行以下命令安装最新 pod。

更换 gem 源

```
gem sources --remove https://rubygems.org/  
gem sources --add https://gems.ruby-china.com/
```

安装 pod

```
sudo gem install cocoapods -n /usr/local/bin
```

如果安装了多个 Xcode，请使用下面的命令选择 Xcode 版本（一般选择最新的 Xcode 版本）。

```
sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer
```

更新 pod 本地库

```
pod setup
```

2. 终端执行以下命令，安装依赖库。

Swift

```
// iOS  
cd Swift/TUIKitDemo  
pod install
```

Objective-C

```
// iOS  
cd iOS/TUIKitDemo  
pod install  
  
// Mac  
cd Mac/TUIKitDemo  
pod install
```

如果安装失败，执行以下命令更新本地的 CocoaPods 仓库列表。

```
pod repo update
```

3. 编译运行：

- iOS
 - Swift 项目进入 Swift/TUIKitDemo 文件夹，打开 `TUIKitDemo.xcworkspace` 编译运行。
 - OC 项目进入 iOS/TUIKitDemo 文件夹，打开 `TUIKitDemo.xcworkspace` 编译运行。
- Mac 进入 Mac/TUIKitDemo 文件夹，打开 `TUIKitDemo.xcworkspace` 编译运行。

注意：

Demo 默认集成了音视频通话功能，由于该功能依赖的音视频 SDK 暂不支持模拟器，请使用真机调试或者运行 Demo。

步骤5：开通云端审核功能

在消息发送、资料修改等场景中，可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，还有可能违法并导致业务被监管部门查封。

即时通信 IM 支持 [云端审核](#)（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。您可直接前往 [云端审核控制台](#) 开通免费试用，每个应用可免费试用7天（含10000次免费抵扣量）。

交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

Web & H5 Vue2/Vue3

最近更新时间: 2025-06-12 10:52:42

TUIKit 介绍

TUIKit 是基于腾讯云 IM SDK 的一款 UI 组件库，它提供了一些通用的 UI 组件，包含会话、聊天、关系链、群组、音视频通话等功能。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。

TUIKit 中的组件在实现 UI 功能的同时，会调用 IM SDK 相应的接口实现 IM 相关逻辑和数据的处理，因而开发者在使用 TUIKit 时只需关注自身业务或个性化扩展即可。

TUIKit 主要功能介绍

TUIKit 主要分为 TUIChat、TUIConversation、TUIGroup、TUIContact、TUISearch 几个 UI 子组件，每个 UI 组件负责展示不同的内容。

说明:

Vue2 & Vue3 版本 TUIKit 全新发布，支持消息云端搜索、在线客服、表情回应功能。

- 在线体验消息云端搜索、在线客服、表情回应功能: [Web & H5 Demo](#)。
- 开通消息云端搜索、在线客服插件: [概述及开通指引](#)。
- 使用表情回应功能: 请参见 [购买指引](#) 购买旗舰版或企业版，并且升级 chat-uikit-vue 版本 $\geq 2.0.5$ ，可以默认使用表情回应功能。

TUIKit Web 端界面效果如下图所示:



TUIKit H5 端界面效果如下图所示:



消息云端搜索功能界面效果如下图所示:



表情回应功能界面效果如下图所示：



开发环境要求

- Vue (全面支持 Vue2 & Vue3，请您在下方接入时选择您所匹配的 Vue 版本接入指引进行接入)
- TypeScript (如果您是 js 项目，请参见 [常见问题 > js 工程如何接入 TUIKit 组件](#) 进行配置 ts 渐进式支持)
- sass (sass-loader 版本 ≤ 10.1.1)
- node (node.js ≥ 16.0.0)
- npm (版本请与 node 版本匹配)

TUIKit 源码集成 (Web & H5)

步骤1: 创建项目

TUIKit 支持使用 webpack 或 vite 创建项目工程，配置 Vue3 / Vue2 + TypeScript + sass。以下是几种项目工程搭建示例：

```
vue-cli
```

注意：
请您务必保证您的 @vue/cli 版本在 5.0.0 以上，您可使用以下示例代码升级您的 @vue/cli 版本至 v5.0.8。

使用 vue-cli 方式创建项目，配置 Vue2 / Vue3 + TypeScript + sass。
如果您尚未安装 vue-cli 或者 vue-cli 版本低于 5.0.0，可以在 terminal 或 cmd 中采用如下方式进行安装：

```
npm install -g @vue/cli@5.0.8 sass sass-loader@10.1.1
```

通过 vue-cli 创建项目，并选择下图中所选配置项。

```
vue create chat-example
```

请务必保证按照如下配置选择：

```
Vue CLI v5.0.8
? Please pick a preset:
  Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
> Manually select features
? Check the features needed for your project: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
   Babel
   TypeScript
   Progressive Web App (PWA) Support
   Router
   Vuex
>  CSS Pre-processors
   Linter / Formatter
   Unit Testing
   E2E Testind
? Choose a version of Vue.js that you want to start the project with (Use arrow keys)
> 3.X 如您需创建 vue3 项目，请选择 3.x
  2.X 如您需创建 vue2 项目，请选择 2.x
? Use class-style component syntax? Yes
? 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): (Use arrow keys)
> Sass/SCSS (with dart-sass)
  Less
  Stylus
? Pick a linter / formatter config: (Use arrow keys)
> ESLint with error prevention only
  ESLint + Airbnb config
  ESLint + Standard config
  ESLint + Prettier
? Pick additional lint features: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
>  Lint on save
   Lint and fix on commit
? Where do you prefer placing config for Babel, ESLint, etc.? (Use arrow keys)
> In dedicated config files
  In package.json
? Save this as a preset for future projects? No
```

创建完成后，切换到项目所在目录：

```
cd chat-example
```

如果您是 vue2 项目，请根据您所使用的 Vue 版本进行以下相应的环境配置，vue3 项目请忽略。

vue2.7

```
npm i vue@2.7.9 vue-template-compiler@2.7.9
```

vue2.6及以下

```
npm i @vue/composition-api unplugin-vue2-script-setup vue@2.6.14 vue-template-compiler@2.6.14
```

vite

注意：
Vite 需要 **Node.js 版本 18+**，20+。当您的包管理器发出警告时，请注意升级您的 Node 版本，详情请参考 [vite 官网](#)。

使用 vite 方式创建项目，按照下图选项配置 Vue + TypeScript。

```
npm create vite@latest
```



```
✓ Project name: ... chat-example
✓ Select a framework: > Vue
✓ Select a variant: > TypeScript
```

之后切换到项目目录，安装项目依赖：

```
cd chat-example
npm install
```

安装 TUIKit 所需 sass 环境依赖：

```
npm i -D sass sass-loader
```

步骤2：下载 TUIKit 组件

通过 [npm](#) 方式下载 TUIKit 组件，为了方便您后续的拓展，建议您将 TUIKit 组件复制到自己工程的 src 目录下：

macOS 端

```
npm i @tencentcloud/chat-uikit-vue
mkdir -p ./src/TUIKit && rsync -av --exclude={'node_modules','package.json','excluded-list.txt'}
./node_modules/@tencentcloud/chat-uikit-vue/ ./src/TUIKit
```

Windows 端

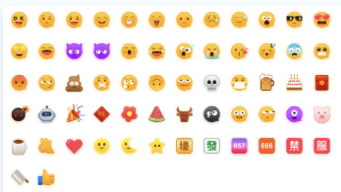
```
npm i @tencentcloud/chat-uikit-vue
```

```
xcopy .\node_modules\@tencentcloud\chat-uikit-vue .\src\TUIKit /i /e
/exclude:.\node_modules\@tencentcloud\chat-uikit-vue\excluded-list.txt
```

步骤3：引入 TUIKit 组件

📌 说明：

为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



在需要展示的页面，引入 TUIKit 的组件即可使用。

例如：在 App.vue 页面中实现以下代码，即可快速搭建聊天界面（以下示例代码同时支持 Web 端与 H5 端）：

⚠️ 注意：

以下示例代码使用了 setup 语法，如果您的项目没有使用 setup 语法，请按照 Vue3/Vue2 的标准方式注册组件。

vue3

```
<template>
  <div id="app">
    <TUIKit :SDKAppID="0" userID="xxx" userSig="xxx" />
    <TUICallKit class="callkit-container" :allowedMinimized="true" :allowedFullScreen="false" />
  </div>
</template>
<script lang="ts" setup>
import { TUIKit } from './TUIKit';
import { TUICallKit } from '@tencentcloud/call-uikit-vue';
</script>
<style lang="scss">
</style>
```

vue2.7

```
<template>
  <div id="app">
    <TUIKit :SDKAppID="0" userID="xxx" userSig="xxx" />
    <TUICallKit class="callkit-container" :allowedMinimized="true" :allowedFullScreen="false" />
  </div>
</template>
<script lang="ts" setup>
import { TUIKit } from './TUIKit';
import { TUICallKit } from '@tencentcloud/call-uikit-vue2';
</script>
<style lang="scss">
</style>
```

vue2.6及以下

```
<template>
  <div id="app">
    <TUIKit :SDKAppID="0" userID="xxx" userSig="xxx" />
    <TUICallKit class="callkit-container" :allowedMinimized="true" :allowedFullScreen="false" />
  </div>
</template>
<script lang="ts" setup>
import { TUIKit } from './TUIKit';
import { TUICallKit } from '@tencentcloud/call-uikit-vue2.6';
</script>
<style lang="scss">
</style>
```

1. 安装支持 composition-api 以及 script setup 的相关依赖，以及 vue2.6 相关依赖。

```
npm i @vue/composition-api unplugin-vue2-script-setup vue@2.6.14 vue-template-compiler@2.6.14
```

2. 在 main.ts/mian.js 中引入 VueCompositionAPI。

```
import VueCompositionAPI from "@vue/composition-api";
Vue.use(VueCompositionAPI);
```

3. 在 `vue.config.js` 中增加, 若没有该文件请新建。

```
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 */
      }),
    ],
  },
  chainWebpack(config) {
    // disable type check and let `vue-tsc` handles it
    config.plugins.delete("fork-ts-checker");
  },
};
```

4. 在 `src/TUIKit/adapters-adapter-vue.ts` 文件最后, 替换导出源:

```
// 初始写法
export * from "vue";
// 替换为
export * from "@vue/composition-api";
```

步骤4: 获取 SDKAppID、userID、userSig

设置 `App.vue` 中 TUIKit 组件的 props 属性 SDKAppID、userID、userSig。

- SDKAppID 信息, 可通过 [即时通信 IM 控制台](#) 获取:



- userID 信息, 可单击 [即时通信 IM 控制台 > 账号管理](#), 切换至目标应用所在账号, 即可创建账号并获取 userID。



- userSig 信息, 可单击 [即时通信 IM 控制台 > 开发工具 > UserSig生成&校验](#), 填写创建的 userID, 即可生成 userSig。



步骤5: 启动项目

执行以下命令启动项目:

vue-cli

④ 说明:

由于 vue-cli 默认开启 webpack 全局 overlay 报错信息提示, 为了您有更好的体验, 建议您在 `vue.config.js` 或其他您工程中的 webpack 配置文件中关闭全局 overlay 报错提示。

webpack4及以上

```

...
module.exports = defineConfig({
  ...
  // 新增关闭 overlay 配置代码
  devServer: {
    client: {
      overlay: false,
    },
  },
},
);

```

webpack3

```

...
module.exports = {
  ...
  // 新增关闭 overlay 配置代码
  devServer: {
    overlay: false,
  },
},
);

```

```
};
```

```
npm run serve
```

vite

```
npm run dev
```

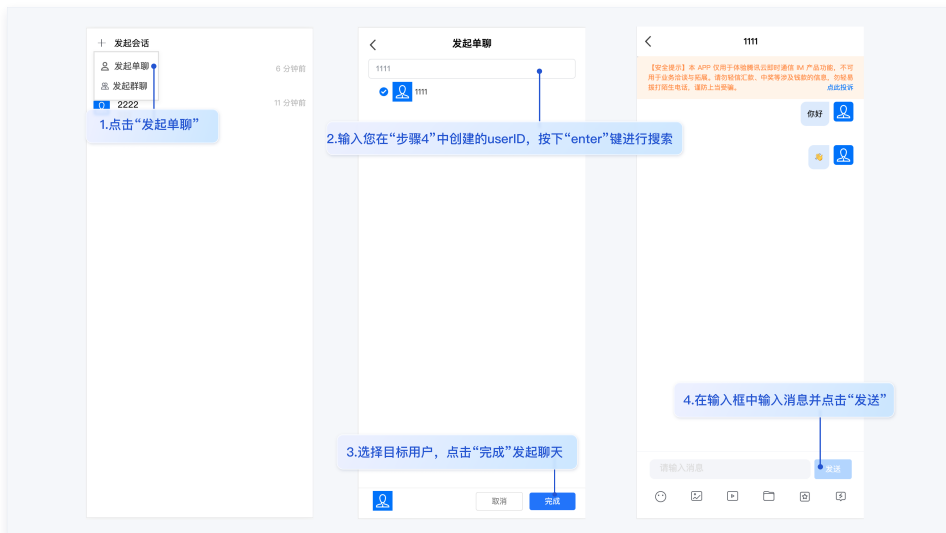
步骤6: 发送您的第一条消息

1. 项目启动之后单击左上角发起单聊。
2. 进入发起单聊弹窗。在搜索栏输入 步骤4 中创建的 userID，选中后单击完成。
3. 在输入框中输入消息并单击发送。

Web 端 “发送您的第一条消息” 具体步骤示例:



H5 端 “发送您的第一条消息” 具体步骤示例:



步骤7：拨打您的第一通电话



生产环境部署指引

1. 部署生产环境时您需要通过服务端生成 userSig，详情请参见 [生成 UserSig](#)。
2. 删除 `App.vue` 中 TUIKit 组件的 props 属性：SDKAppID、userID、userSig，代码修改如下：

```
<template>
  <div id="app">
    <TUIKit />
    <TUICallKit class="callkit-container" :allowedMinimized="true" :allowedFullScreen="false" />
  </div>
</template>
```

3. 在 `main.ts/main.js` 中引入登录模块，并进行登录。

```
import { TUILogin } from '@tencentcloud/tui-core';
import { framework } from './TUIKit/adapter-vue';
```

```
TUILogin.login({
  SDKAppID: 0,
  userID: 'xxx',
  userSig: 'xxx',
  useUploadPlugin: true,
  framework,
}).then(() => {})
.catch((error) => {})
```

更多高级特性（强烈推荐）

消息云端搜索

注意：

@tencentcloud/chat-uikit-vue ≥ 2.0.0 默认支持，只需要进入 [控制台 > 插件市场 > 云端搜索](#)，单击[免费试用](#)或[购买插件](#)即可使用消息云端搜索功能。



在线客服

注意：

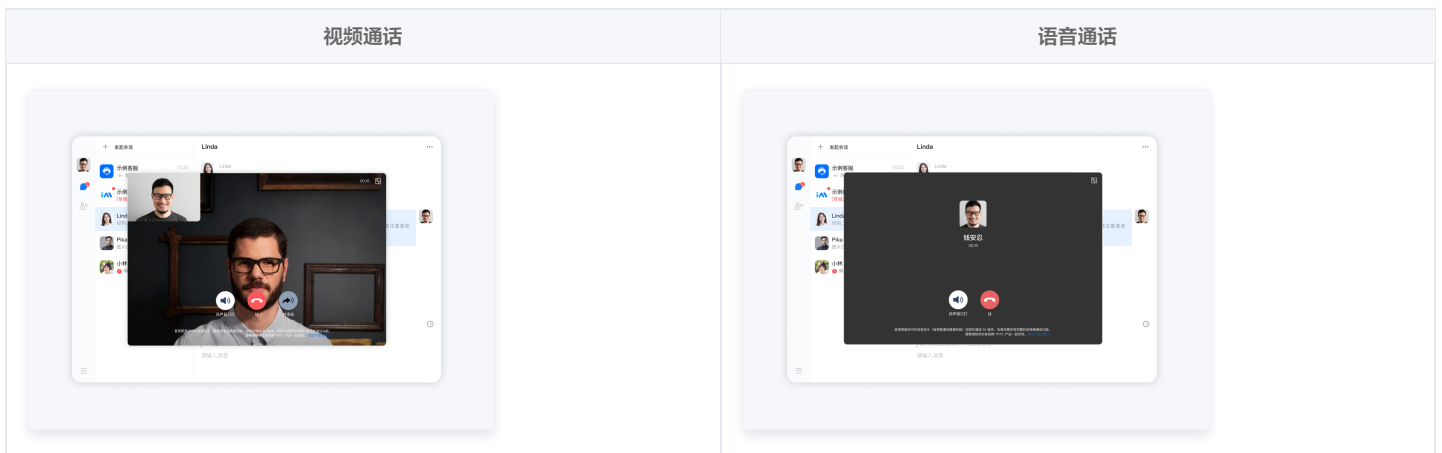
@tencentcloud/chat-uikit-vue ≥ 2.0.0 默认支持，只需要进入 **控制台 > 插件市场 > 客服插件**，单击**免费试用**或**购买插件**即可使用在线客服功能。



音视频通话

注意：

@tencentcloud/chat-uikit-vue ≥ 1.4.0 默认支持，只需要进入 **控制台 > 应用管理 > 应用详情**，单击**免费体验**或**购买正式版**即可使用音视频通话功能。



云端审核

在消息发送、资料修改场景中，很有可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，更很有可能违法并导致业务被监管部门查封。

即时通信 IM 支持内容审核（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。

云端审核功能 在服务端检测由单聊、群聊、资料场景中产生的文本、图片、音频、视频内容，支持针对不同场景的不同内容分别配置审核策略，并对识别出的不安全内容进行拦截。此功能已提供默认预设拦截词库和审核场景，只需在 IM 控制台打开功能开关，即可直接使用。

参见文档步骤 [快速开通云端审核](#) 功能。

常见问题

1. 音视频通话能力包未开通？音视频通话发起失败？

请单击 [音视频通话](#) > [常见问题](#) 查看解决方案。

2. 我只想集成聊天窗口，如何单独集成 TUIChat？如果直接打开指定单聊/群聊会话？

独立集成聊天窗口，请参考 [TUIChat 独立集成方案](#)。

3. js 工程如何接入 TUIKit 组件？

TUIKit 仅支持 ts 环境运行，您可以通过渐进式配置 typescript 来使您项目中已有的 js 代码与 TUIKit 中 ts 代码共存。

vue-cli

请在您 vue-cli 脚手架创建的工程根目录执行：

```
vue add typescript
```

之后按照如下进行配置项进行选择（为了保证能同时支持原有 js 代码与 TUIKit 中 ts 代码，请您务必严格按照以下五个选项进行配置）

```
Run `npm audit` for details.
✓ Successfully installed plugin: @vue/cli-plugin-typescript

? Use class-style component syntax?  Yes
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)?  Yes
? Convert all .js files to .ts?  No
? Allow .js files to be compiled?  Yes
? Skip type checking of all declaration files (recommended for apps)?  Yes

🔥 Invoking generator for @vue/cli-plugin-typescript...
📦 Installing additional dependencies...
```

完成以上步骤后，请重新运行项目。

vite

请在您 vite 创建的工程根目录执行：

```
npm install -D typescript
```

4. 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](#)。

⚠ 注意

本文示例代码采用的获取 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 签发方式请参见上文。

5. 运行时报错："TypeError: Cannot read properties of undefined (reading "getFriendList")"

若按照上述步骤接入后，运行时出现以下错误，请您务必删除 TUIKit 目录下的 node_modules 目录，以保证 TUIKit 的依赖唯一性，避免 TUIKit 多份依赖造成问题。

```

▼ [Vue warn]: Error in v-on handler (Promise/async): "TypeError: Cannot read properties of undefined (reading 'getFriendList')"

found in

---> <Index> at
src/TUIKit/components/TUISearch/index.vue
  <App> at src/App.vue
    <Root>

warn @ vue.runtime.esm.js:4568
    
```

6. 运行时报错: /chat-example/src/TUIKit/components/TUIChat /message-input/message-input-editor.vue .ts (8, 23) TS1005: expected.

```

98% after emitting CopyPlugin
ERROR Failed to compile with 2 errors 16:20:59
error in [redacted] /chat-example/src/TUIKit/components/TUIChat/message-input/message-input-editor.vue.ts
[tsl] ERROR in [redacted] /chat-example/src/TUIKit/components/TUIChat/message-input/message-input-editor.vue.ts(8,23)
TS1005: ',' expected.
    
```

出现以上报错信息，是因为您安装的 @vue/cli 版本过低导致，请您务必保证您的 @vue/cli 版本在 5.0.0 及以上。升级方式如下：

```
npm install -g @vue/cli@5.0.8
```

7. 运行时报错: Failed to resolve loader: sass-loader

```

ERROR Failed to compile with 1 error
Failed to resolve loader: sass-loader
You may need to install it.
No issues found.
    
```

出现以上报错信息，是因为您未安装 `sass` 环境导致，请执行以下命令进行 `sass` 环境安装：

```
npm i -D sass sass-loader@10.1.1
```

8. ESLint 报错: error 'withDefaults' is not defined no-undef? 'defineProps' is not defined no-undef? 以及 ESLint 其他报错?

```

/Users/ashsterchen/Documents/silvia-work/test/test-2.0.7/chat-example-vue2.6/src/TUIKit/plugins/plugin-components/message-call/message-call-group.vue
21:15 error 'withDefaults' is not defined no-undef
21:28 error 'defineProps' is not defined no-undef

/Users/ashsterchen/Documents/silvia-work/test/test-2.0.7/chat-example-vue2.6/src/TUIKit/plugins/plugin-components/message-customer/message-customer-service.vue
10:15 error 'withDefaults' is not defined no-undef
10:28 error 'defineProps' is not defined no-undef

/Users/ashsterchen/Documents/silvia-work/test/test-2.0.7/chat-example-vue2.6/src/TUIKit/plugins/plugin-components/message-plugin-layout.vue
58:15 error 'withDefaults' is not defined no-undef
58:28 error 'defineProps' is not defined no-undef
65:15 error 'defineEmits' is not defined no-undef

/Users/ashsterchen/Documents/silvia-work/test/test-2.0.7/chat-example-vue2.6/src/TUIKit/plugins/plugin-components/message-plugin.vue
60:15 error 'withDefaults' is not defined no-undef
60:28 error 'defineProps' is not defined no-undef
65:15 error 'defineEmits' is not defined no-undef

/Users/ashsterchen/Documents/silvia-work/test/test-2.0.7/chat-example-vue2.6/src/TUIKit/plugins/plugin-components/message-room/message-room-default.vue
19:15 error 'withDefaults' is not defined no-undef
19:28 error 'defineProps' is not defined no-undef

* 467 problems (231 errors, 236 warnings)
18 errors and 0 warnings potentially fixable with the `--fix` option.

webpack compiled with 3 errors
    
```

若 chat-uikit-vue 拷贝到 src 目录汇总与您本地项目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：

```
# .eslintignore
src/TUIKit
```

9. vue/cli 如何关闭 dev 模式下，webpack 全屏 overlay error 报错信息提示?

可以在您项目根目录的 `vue.config.js` 中进行关闭:

webpack4

```
module.exports = defineConfig({
  ...
  devServer: {
    client: {
      overlay: false,
    },
  },
});
```

webpack3

```
module.exports = {
  ...
  devServer: {
    overlay: false,
  },
};
```

10. 出现 Component name "XXXX" should always be multi-word 怎么办?

- IM TUIKit web 所使用的 ESLint 版本为 v6.7.2，对于模块名的驼峰式格式并不进行严格校验。
- 如果您出现此问题，您可以在 `.eslintrc.js` 文件中进行如下配置:

```
module.exports = {
  ...
  rules: {
    ...
    'vue/multi-word-component-names': 'warn',
  },
};
```

11. 出现 ERESOLVE unable to resolve dependency tree 怎么办?

`npm install` 的时候如果出现 `ERESOLVE unable to resolve dependency tree`，表示依赖安装冲突，可采用以下方式进行安装:

```
npm install --legacy-peer-deps
```

12. 运行报错如下'vue packages version mismatch',如何解决?

- 如果您是 `vue2.7` 项目，请在您项目根目录执行:

```
npm i vue@2.7.9 vue-template-compiler@2.7.9
```

- 如果您是 `vue2.6` 项目，请在您项目根目录执行:

```
npm i vue@2.6.14 vue-template-compiler@2.6.14
```

13. vite 项目 npm run build 之后 ts 报错?

```
node_modules/@tencentcloud/chat-ui-kit-vue/dist/customer-service-plugin/components/message-rating/index.vue:35 - error TS2551: Property 'RATING_TEMPLATE_TYPE' does not exist on type '{ $: ComponentInternalInstance; $data: {}; $props: Partial<Pick< & limited; randomId: message>; Record<string, any>; onSendMessage: (...args: any[]) => any } & ... 4 more ... & { ... }'; never; ... 10 more ...; &watcher extends string | (...args: any) => any}<source: T, cb: T extends (...args: any) => infer R ? (arg.., Did you mean 'ratingTemplate'?

v-if="ratingTemplate.type === RATING_TEMPLATE_TYPE_STAR"

src/TUIKit/components/common/FetchMore/index.vue:66:16 - error TS2384: Cannot find name 'uni'.

observer = uni

src/TUIKit/components/common/ImagePreviewer/index.vue:164:7 - error TS2322: Type 'Timeout' is not assignable to type 'number'.

timer = setTimeout(() => {

src/TUIKit/components/common/ImagePreviewer/index.vue:189:5 - error TS2322: Type 'Timeout' is not assignable to type 'number'.

timer = setTimeout(() => {

src/TUIKit/components/common/Transfer/index.vue:97:21 - error TS2303: Operator '>' cannot be applied to types '{ toString: (radius?: number) => string; toFixed: (fractionDigits?: number) => string; toExponential: (fractionDigits?: number) => string; toPrecision: (precision?: number) => string; valueOf: () => number; toLocaleString: (...; }' and 'number'.

v-if="transferTotal > transferList.length"

src/TUIKit/components/TUIChat/chat-header/index.vue:17:39 - error TS2345: Argument of type 'string | number | object | { onClicked?: Function; onLongPressed?: Function; onTouched?: Function; onSwiped?: Function; }' is not assignable to parameter of type 'ExtensionInfo'.
Type 'string' is not assignable to type 'ExtensionInfo'.

@click.stop="handleExtensions(iten)"

src/TUIKit/components/TUIChat/chat-header/index.vue:18:27 - error TS2339: Property 'icon' does not exist on type 'string | number | object | { onClicked?: Function; onLongPressed?: Function; onTouched?: Function; onSwiped?: Function; }'.
Property 'icon' does not exist on type 'string'.

<Icon :file="iten.icon"~/icon

src/TUIKit/components/TUIChat/chat-header/index.vue:41:39 - error TS2345: Argument of type 'undefined[]' is not assignable to parameter of type 'ExtensionInfo'.
Type 'undefined[]' is missing the following properties from type 'ExtensionInfo': weight, text, icon, data, listener

const extensions = ref<ExtensionInfo[]>()
```

- 原因: package.json script 下 "build": "vue-tsc && vite build" 中的 vue-tsc 命令导致。

```
"scripts": {
  "dev": "vite",
  "build": "vue-tsc && vite build",
  "preview": "vite preview"
},
```

- 解决方案: 删除 vue-tsc 即可。"build": "vite build"。

```
"scripts": {
  "dev": "vite",
  "build": "vite build",
  "preview": "vite preview"
},
```

14. 使用 pnpm 下载缺少依赖?

pnpm 安装 @tencentcloud/chat-uikit-vue 后拷贝到 TUIkit 目录使用会存在内部依赖缺失问题, 建议您优先选择 npm / yarn 方式安装。若您确实需要使用 pnpm 安装依赖, 请在进行完上文接入步骤后:

- 将 node_modules/@tencentcloud/chat-uikit-vue/package.json 中的依赖项拷贝到您自身项目根目录 package.json 中。
- 运行 pnpm i, 进行内部缺失依赖安装。

15. H5 页面 iOS 低版本机型出现白屏?

您可以通过配置 polyfill 兼容低机型版本。(什么是 polyfill? polyfill 是一种用于解决浏览器兼容问题的技术。它用于模拟某些浏览器不支持的新特性, 使得这些特性能在不支持的浏览器中正常工作。更多详情请参见 babel-polyfill。)

- 安装 babel-polyfill

```
npm i @babel/polyfill
```

- 入口文件引入 'babel-polyfill'

```
import 'babel-polyfill';
```

- 新增 babel.config.js 配置

```
module.exports = {
  presets: [
    [
      '@vue/cli-plugin-babel/preset',
      {
        useBuiltIns: 'entry',
        corejs: 3,
      },
    ],
  ],
}
```

```
],  
[  
  '@babel/preset-env',  
  {  
    targets: {  
      browsers: ['iOS >= 11'],  
    },  
    useBuiltIns: 'entry',  
    corejs: 3,  
  },  
],  
];  
};
```

16. 运行报错: Uncaught TypeError: marked__WEBPACK_IMPORTED_MODULE_0_Marked is not a constructor

如果您运行过程中出现如下错误, 说明您当前 Vue CLI 构建的项目环境版本较低, 需要降低 uikit 中使用的 marked 版本至 5.1.2。

```
✘ Uncaught TypeError: marked__WEBPACK_IMPORTED_MODULE_0_Marked is not a constructor      marked.ts:22  
    at eval (marked.ts:22:1)  
    at ./src/TUIKit/components/TUIChat/message-list/message-elements/message-stream-markdown/marked.ts (app.js:8381:1)  
    at __webpack_require__ (app.js:849:30)  
    at fn (app.js:151:20)  
    at ./node_modules/cache-loader/dist/cjs.js?!./node_modules/babel-loader/lib/index.js?!./node_modules/ts-  
loader/index.js?!./node_modules/cache-loader/dist/cjs.js?!./node_modules/vue-  
loader/lib/index.js?!./node_modules/unplugin/dist/webpack/loaders/transform.js?unpluginName=unplugin-vue2-script-  
setup!./src/TUIKit/components/TUIChat/message-list/message-elements/message-stream-markdown/index.vue?  
vue&type=script&lang=ts (app.js:1310:1)  
    at __webpack_require__ (app.js:849:30)  
    at fn (app.js:151:20)  
    at eval (index.vue:1:1)  
    at ./src/TUIKit/components/TUIChat/message-list/message-elements/message-stream-markdown/index.vue?  
vue&type=script&lang=ts (app.js:8357:1)  
显示已列入忽略列表的帧
```

请在您项目的 根目录 使用以下脚本降低 marked 版本:

```
npm i marked@5.1.2 --legacy-peer-deps
```

17. 编译报错 node_modules/marked/lib/marked.esm.js: Class private methods are not enabled.

如果您运行过程中出现如下错误, 说明您当前使用的 marked 版本过低, 请升级 marked 版本至 6.0.0。

```
ERROR in ./node_modules/marked/lib/marked.esm.js  
Module build failed (from ./node_modules/babel-loader/lib/index.js):  
SyntaxError: /Users/ashsterchen/Documents/silvia-work/code/test-release/v2.3.6/chat-example-2/node_modules/marked/lib/marked.esm.js: Class private  
methods are not enabled. Please add '@babel/plugin-transform-private-methods' to your configuration.  
   2716 |   }  
   2717 | }  
> 2718 | #parseMarkdown(lexer, parser) {  
       | ^  
   2719 |   return (src, opt, callback) => {  
   2720 |     if (typeof opt === 'function') {  
   2721 |       callback = opt;  
   at File.buildCodeFrameError (/Users/ashsterchen/Documents/silvia-work/code/test-release/v2.3.6/chat-example-2/node_modules/@babel/core/lib/tra  
nsformation/file/file.js:195:12)
```

请在您项目的 根目录 使用以下脚本降低 marked 版本:

```
npm i marked@6.0.0 --legacy-peer-deps
```

交流与反馈

[点此进入 IM 社群](#), 享有专业工程师的支持, 解决您的难题。

相关文档

Vue2 & Vue3 Uikit 相关:

- [集成音视频通话](#)
- [chat-uikit-vue npm](#)
- [Vue2 Demo源码及跑通示例](#)
- [Vue3 Demo源码及跑通示例](#)

Vue2 & Vue3 Uikit 逻辑层: engine 相关

- [chat-uikit-engine npm 仓库](#)
- [chat-uikit-engine 接口文档](#)

小程序

最近更新时间：2025-06-12 10:52:42

TUIkit 介绍

TUIkit 是基于腾讯云 IM SDK 的一款 UI 组件库，它提供了一些通用的 UI 组件，包含会话、聊天、群组、音视频通话等功能。

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。

TUIkit 中的组件在实现 UI 功能的同时，会调用 IM SDK 相应的接口实现 IM 相关逻辑和数据的处理，因而开发者在使用 TUIkit 时只需关注自身业务或个性化扩展即可。

TUIkit 主要功能介绍

TUIkit 主要分为 TUIConversation、TUIChat、TUICallKit、和TUIGroup 几个 UI 子组件，每个 UI 组件负责展示不同的内容。

界面效果如下图所示：



在线客服功能界面效果如下图所示：



注意：

1. 如果您在集成 TUIKit 之后还需要客服功能，请参见 [客服场景](#)。
2. 如果您在集成 TUIKit 之后还需要支持单人/多人音视频通话功能，请参见 [使用音视频通话](#)。

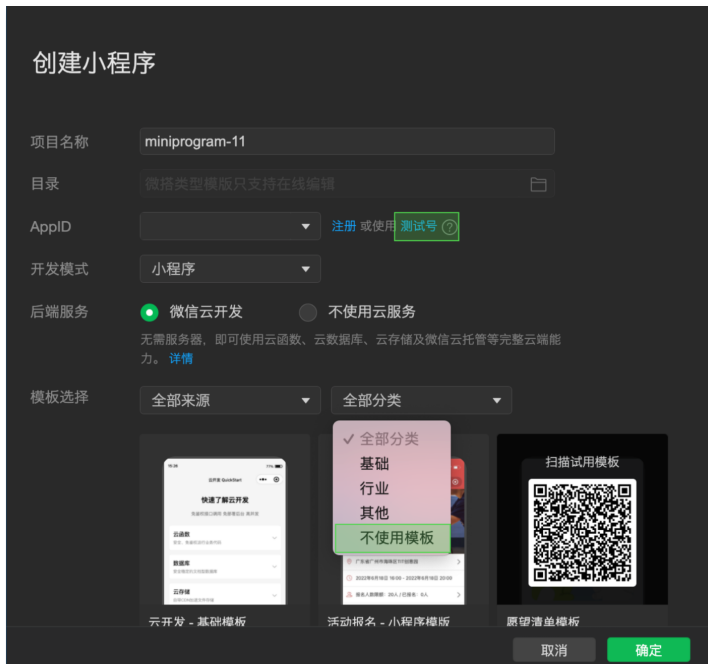
开发环境要求

- 微信开发者工具
- JavaScript
- node (12.13.0 ≤ node版本 ≤ 17.0.0, 推荐使用 Node.js 官方 LTS 版本 16.17.0)
- npm (版本请与 node 版本匹配)

TUIKit 源码集成

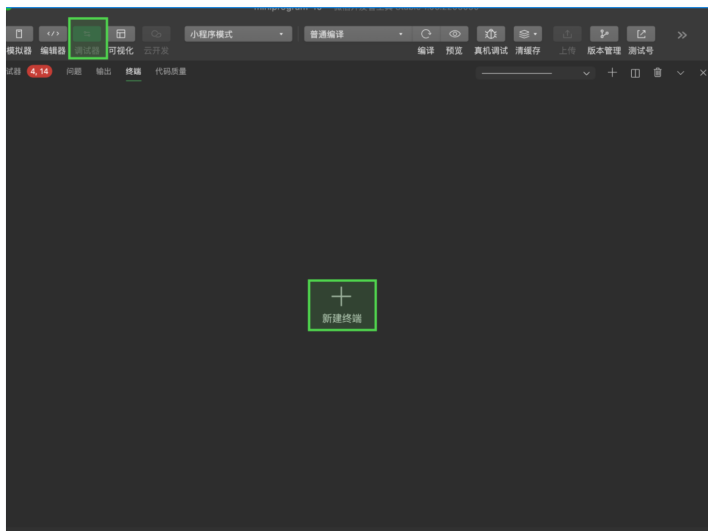
步骤1: 创建项目

在微信开发者工具上创建一个小程序项目，选择不使用模板。



步骤2: 下载 TUIKit 组件

微信开发者工具创建的小程序不会默认创建 package.json 文件，因此您需要先创建 package.json 文件。新建终端，如下：



输入：

```
npm init
```

然后通过 npm 方式下载 TUIKit 组件，为了方便您后续的拓展，建议您将 TUIKit 组件复制到自己的小程序目录下：

macOS 端

```
npm i @tencentcloud/chat-uikit-wechat
```

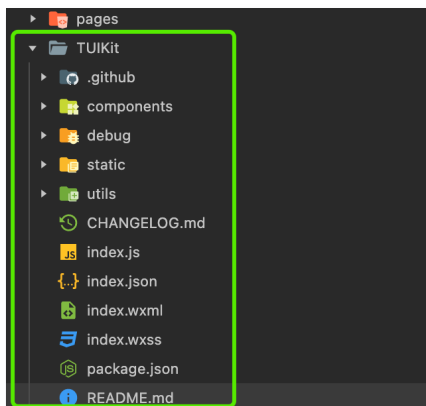
```
mkdir -p ./TUIKit && cp -r node_modules/@tencentcloud/chat-uikit-wechat/ ./TUIKit
```

Windows 端

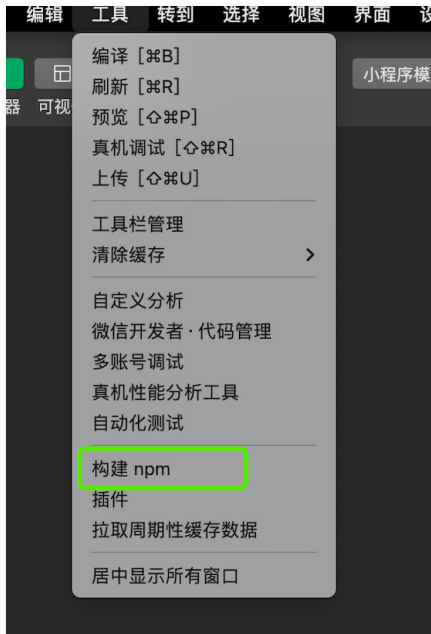
```
npm i @tencentcloud/chat-uikit-wechat
```

```
xcopy node_modules\@tencentcloud\chat-uikit-wechat .\TUIKit /i /e
```

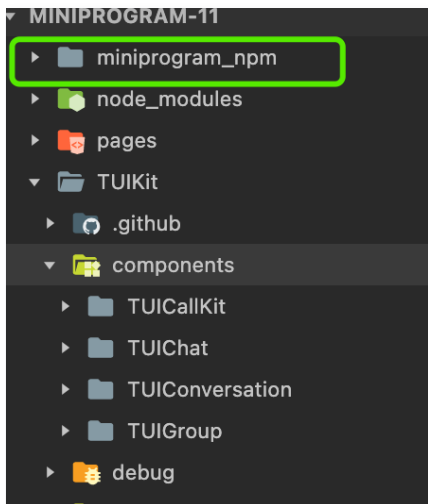
成功后目录结构如图所示：



构建 npm：微信开发者工具-工具-选择构建 npm：



构建 npm 后目录如下 (新增 miniprogram_npm 文件夹) :



注意:

您构建 npm 时, 若出现如下图所示提示, 请单击确定, 该提示信息不会影响到组件的正常使用。



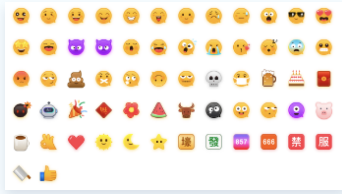
步骤3: 引入 TUIKit 组件

说明:

- 小程序分包有如下好处:
 - 规避所有逻辑代码放主包, 导致主包文件体积超限问题。
 - 支持按需载入, 降低小程序载入耗时和页面渲染耗时。

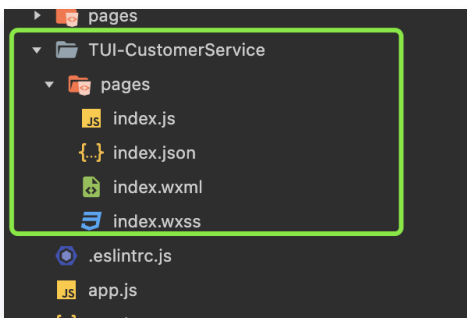
○ 支持更加复杂的功能。

- 为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



分包引入 (推荐)

1. 在自己项目里创建分包，本文以 TUI-CustomerService 为例。和 pages 同级创建 TUI-CustomerService 文件夹，并在文件夹内部创建 pages 文件夹并且其下创建 index 页面。创建后的目录结构：



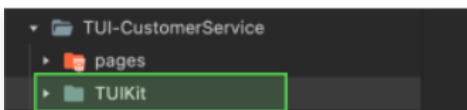
2. 在 app.json 文件注册分包。

```

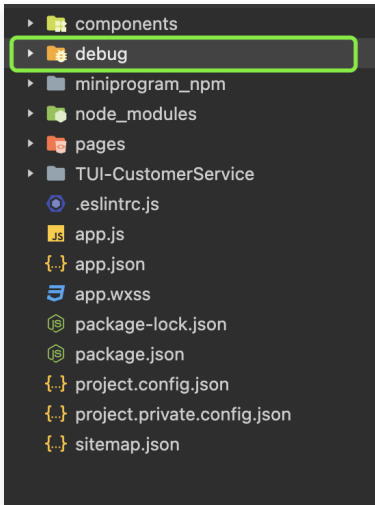
{
  "pages": [
    "pages/index/index"
  ],
  "subPackages": [
    {
      "root": "TUI-CustomerService",
      "name": "TUI-CustomerService",
      "pages": [
        "pages/index"
      ],
      "independent": false
    }
  ],
  "window": {
    "backgroundTextStyle": "light",
    "navigationBarBackgroundColor": "#fff",
    "navigationBarTitleText": "Weixin",
    "navigationBarTextStyle": "black"
  },
  "style": "v2",
  "sitemapLocation": "sitemap.json"
}

```

3. 将 TUIkit 文件夹复制到分包目录下。成功后的目录结构：



4. 将 TUIkit 文件夹下的 debug 文件夹复制到主包 (app.js 同级目录下)，成功后的目录结构：



5. 在分包内引用 TUIkit组件，为此需要分别修改分包内部 index.wxml 、 index.js 、 index.json 文件，以及 app.js 文件。

wxml 文件



```
<view>
  <TUIKit config="{{config}}" id="TUIKit"></TUIKit>
</view>
```

config 的参数如下表所示：

参数	类型	是否必填	含义
userID	String	是	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（_）。
SDKAPPID	Number	是	云通信应用的 SDKAppID。
SECRETKEY	String	是	密钥信息，详情可参见 步骤4 。
EXPIRETIME	Number	否	userSig 过期时间。

js 文件



```
Page({

  // 其他代码

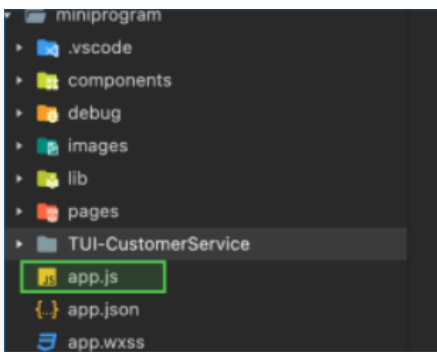
  onLoad() {
    const TUIKit = this.selectComponent('#TUIKit');
    TUIKit.init();
  },
});
```

json 文件



```
{
  "usingComponents": {
    "TUIKit": "../TUIKit/index"
  },
  "navigationStyle": "custom"
}
```

app.js 文件



```
import TencentCloudChat from '@tencentcloud/chat';
import TIMUploadPlugin from 'tim-upload-plugin';
import TIMProfanityFilterPlugin from 'tim-profanity-filter-plugin';
import { genTestUserSig } from './debug/GenerateTestUserSig';
App({
  onLaunch: function () {
    wx.$TUIKit = TencentCloudChat.create({
      SDKAppID: this.globalData.config.SDKAPPID,
    });
    const userSig = genTestUserSig(this.globalData.config).userSig;
    wx.$chat_SDKAppID = this.globalData.config.SDKAPPID;
    wx.TencentCloudChat = TencentCloudChat;
    wx.$chat_userID = this.globalData.config.userID;
```

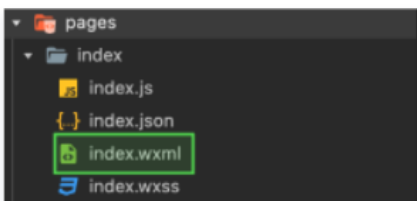
```

wx.$chat_userSig = userSig;
wx.$TUIKit.registerPlugin({ 'tim-upload-plugin': TIMUploadPlugin });
wx.$TUIKit.registerPlugin({ 'tim-profanity-filter-plugin': TIMProfanityFilterPlugin });
wx.$TUIKit.login({
  userID: this.globalData.config.userID,
  userSig
});
// 监听系统级事件
wx.$TUIKit.on(wx.TencentCloudChat.EVENT.SDK_READY, this.onSDKReady,this);
},
onUnload() {
  wx.$TUIKit.off(wx.TencentCloudChat.EVENT.SDK_READY, this.onSDKReady,this);
},
globalData: {
  config: {
    userID: '', // User ID
    SECRETKEY: '', // Your secretKey
    SDKAPPID: 0, // Your SDKAppID
    EXPIRETIME: 604800,
  },
},
onSDKReady(event) {
  // 监听到此事件后可调用 SDK 发送消息等 API, 使用 SDK 的各项功能。
}
});

```

6. 按需载入分包, 您需要修改主包 pages 下的 index.wxml、index.js。

wxml 文件

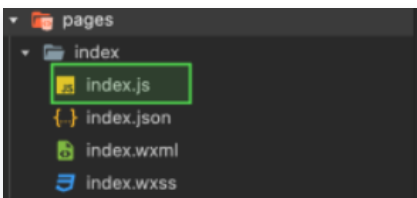


```

<view class="container" bindtap="handleJump">
  载入腾讯云 IM 分包
</view>

```

js 文件



```

Page({
  handleJump() {
    wx.navigateTo({

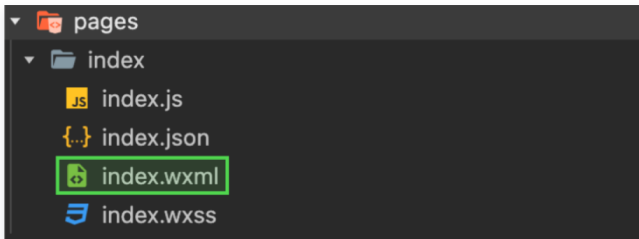
```

```
url: '../TUI-CustomerService/pages/index',
  })
}
})
```

主包引入（适用于业务逻辑简单的小程序）

在 page 页面引用 TUIKit 组件，为此您需要分别修改 index.wxml、index.js 和 index.json。

wxml 文件

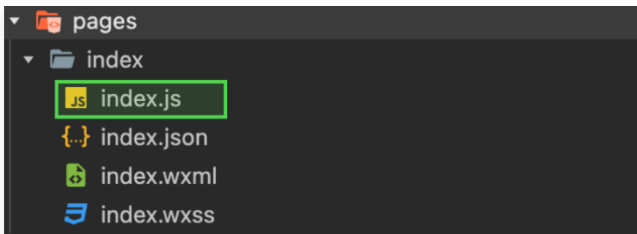


```
<view>
  <TUIKit config="{{config}}" id="TUIKit"></TUIKit>
</view>
```

config 的参数如下表所示：

参数	类型	是否必填	含义
userID	String	是	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（_）
SDKAPPID	Number	是	云通信应用的 SDKAppID
SECRETKEY	String	是	密钥信息，详情可参考 步骤4
EXPIRETIME	Number	否	userSig 过期时间

js 文件



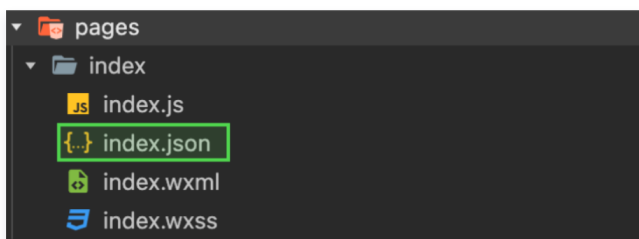
```
import TencentCloudChat from '@tencentcloud/chat';
import TIMUploadPlugin from 'tim-upload-plugin';
```

```
import TIMProfanityFilterPlugin from 'tim-profanity-filter-plugin';
import { genTestUserSig } from '../../TUIKit/debug/GenerateTestUserSig';

Page({
  data: {
    config: {
      userID: '', //User ID
      SDKAPPID: 0, // Your SDKAppID
      SECRETKEY: '', // Your secretKey
      EXPIRETIME: 604800,
    }
  },

  onLoad() {
    const userSig = genTestUserSig(this.data.config).userSig
    wx.$TUIKit = TencentCloudChat.create({
      SDKAppID: this.data.config.SDKAPPID
    })
    wx.$chat_SDKAppID = this.data.config.SDKAPPID;
    wx.$chat_userID = this.data.config.userID;
    wx.$chat_userSig = userSig;
    wx.TencentCloudChat = TencentCloudChat;
    wx.$TUIKit.registerPlugin({ 'tim-upload-plugin': TIMUploadPlugin });
    wx.$TUIKit.registerPlugin({ 'tim-profanity-filter-plugin': TIMProfanityFilterPlugin });
    wx.$TUIKit.login({
      userID: this.data.config.userID,
      userSig
    });
    wx.setStorage({
      key: 'currentUserID',
      data: [],
    });
    wx.$TUIKit.on(wx.TencentCloudChat.EVENT.SDK_READY, this.onSDKReady, this);
  },
  onUnload() {
    wx.$TUIKit.off(wx.TencentCloudChat.EVENT.SDK_READY, this.onSDKReady, this);
  },
  onSDKReady() {
    const TUIKit = this.selectComponent('#TUIKit');
    TUIKit.init();
  }
});
```

json 文件

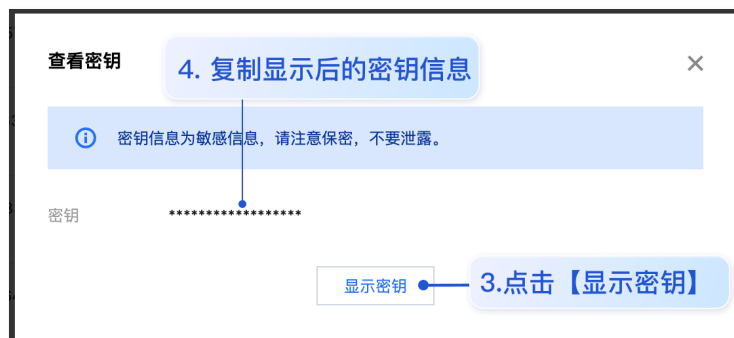


```
{
  "usingComponents": {
    "TUIKit": "../../TUIKit/index"
  },
}
```

```
"navigationStyle": "custom"
}
```

步骤4：获取 SDKAppID、SECRETKEY 与 userID

通过 [即时通信 IM 控制台 > 账号管理](#) 获取 SDKAPPID、SECRETKEY，并设置 [步骤3](#) 示例代码中的 SDKAPPID、SECRETKEY。



通过 [即时通信 IM 控制台 > 账号管理](#) 创建账号并获取 userID，并设置 [步骤3](#) 示例代码中的 userID。



步骤5：编译小程序

1. 请在本地设置里面勾选上“不校验合法域名、web-view (业务域名)、TLS 版本以及 HTTPS 证书”。



2. 单击清缓存 > 全部清除。避免开发者的缓存造成渲染异常。



3. 单击编译图标。



步骤6：发送您的第一条消息



注意:

如果您在集成 TUIkit 之后还需要支持单人/多人音视频通话功能, 详情请参见 [使用音视频通话](#)。

更多高级特性 (强烈推荐)

在线客服

注意:

@tencentcloud/chat-uikit-wechat ≥ 2.0.0 默认支持, 只需要进入 [控制台 > 插件市场 > 客服插件](#), 单击[免费试用](#)或[购买插件](#)即可使用在线客服功能。



云端审核

在消息发送、资料修改场景中, 很有可能会扩散不合适的内容, 特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容, 不仅严重损害了用户们的身心健康, 更很有可能违法并导致业务被监管部门查封。

即时通信 IM 支持内容审核 (反垃圾信息) 功能, 可针对不安全、不适宜的内容进行自动识别、处理, 为您的产品体验和业务安全保驾护航。可以通过 [云端审核](#) 方式来实现。

常见问题

1. 什么是 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。

2. 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](#)。

⚠ 注意:

本文示例代码采用的获取 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 签发方式请参见上文。

3. 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 > 开发管理 > 开发设置 > 服务器域名 中进行域名配置，域名配置方法请参见: [小程序 socket 合法域名](#)。

交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

uniapp vue2/vue3

最近更新时间: 2025-04-30 10:26:42

! 体验反馈:

您在文档接入过程中是否遇到困难? 接入过程是否显得过于复杂? 无论是建议、疑惑还是吐槽, 都欢迎在这里向我们反馈。希望各位开发者能在闲暇时花费1分钟填写 [TUIKit for Uniapp 体验反馈](#)。我们将根据您的反馈优化TUIKit, 以提升您的使用体验。

TUIKit 新手接入指引视频

对于 TUIKit v2.2.0 及之后的版本, 如果您是首次接入, 强烈建议您参考视频指引完成 TUIKit 的接入。

[观看视频](#)

chat-uikit-uniapp 介绍

! 说明:

- TUIKit v2.2.0 起请按照本文所述步骤接入。
- TUIKit v2.1.3 起支持通过 cli 脚手架工具创建的项目集成, 集成指引请点击 [uniapp vue2/vue3 \(cli 脚手架\)](#) 查看。

chat-uikit-uniapp (vue2/vue3) 是一款基于腾讯云 Chat SDK 的 uniapp UI 组件库, 提供了一些通用的 UI 组件, 包含会话、聊天、群组等功能。基于这些精心设计的 UI 组件, 您可以快速构建优雅的、可靠的、可扩展的 Chat 应用。chat-uikit-uniapp 界面效果如下图所示:



打包 App 集成 [uni-app 腾讯云推送服务 \(Push\)](#) 支持在线推送和离线推送, 开发 app 必备。效果如下:



消息云端搜索功能界面效果如下图所示：



在线客服功能界面效果如下图所示：



支持平台

- Android
- iOS
- 微信小程序
- H5

开发环境要求

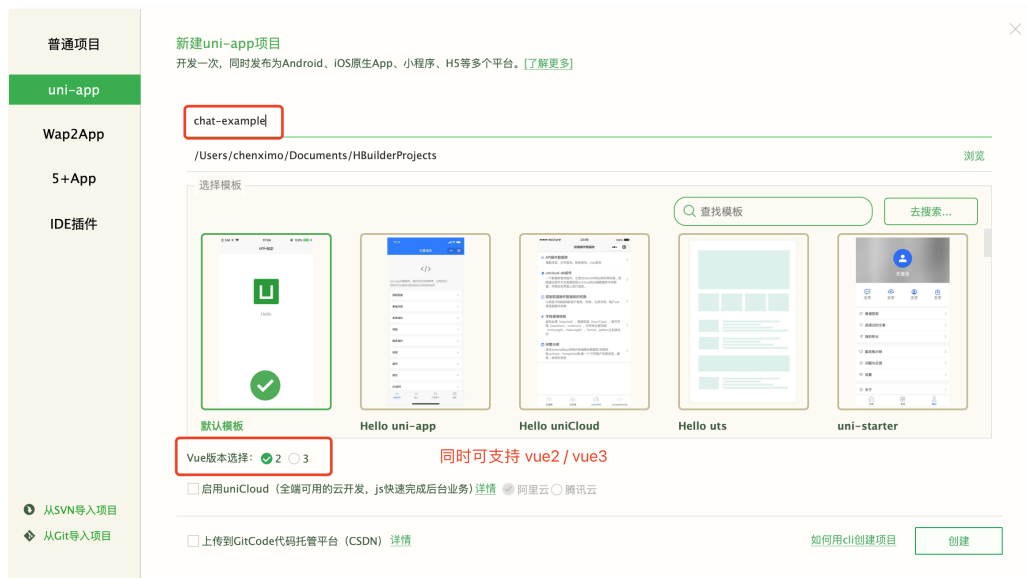
- HBuilderX 升级到新版本
- TypeScript / JavaScript (TUIKit 使用 ts 语言开发, 支持在 js 或者 ts 项目中集成)
- Vue2 / Vue3
- sass (sass-loader 版本 ≤ 10.1.1)
- node (12.13.0 ≤ node , 推荐使用 Node.js 官方 LTS 版本 16.17.0)
- npm (版本请与 node 版本匹配)

TUIKit 源码集成

完成以下步骤即可发送您的第一条消息。

步骤1: 创建项目 (已有项目可忽略)

打开 HbuilderX, 在菜单栏中选择 “文件-新建-项目”, 创建一个名为 chat-example 的 uni-app 项目。



步骤2: 下载 TUIKit 组件

- HBuilderX 创建项目时默认不会创建 package.json 文件, 请在项目根目录下执行以下命令创建 package.json 文件:

```
npm init -y
```

- 下载 TUIKit 并拷贝源码至项目中:

macOS 端

通过 NPM 方式下载 TUIKit 组件:

```
npm i @tencentcloud/chat-uikit-uniapp unplugin-vue2-script-setup
```

为了方便您对 UI 进行扩展, 请在项目的根目录下执行以下命令, 将 TUIKit 组件复制到项目中:

```
mkdir -p ./TUIKit && rsync -av --exclude={'node_modules','package.json','excluded-list.txt'} ./node_modules/@tencentcloud/chat-uikit-uniapp/ ./TUIKit
```

TUIKit 默认支持 客服插件, 需要将客服插件源码复制到项目中 (请不要跳过本步骤)。

```
mkdir -p ./TUIKit/tui-customer-service-plugin && rsync -av ./node_modules/@tencentcloud/tui-customer-service-plugin/ ./TUIKit/tui-customer-service-plugin
```

Windows 端

通过 [NPM](#) 方式下载 TUIKit 组件:

```
npm i @tencentcloud/chat-uikit-uniapp unplugin-vue2-script-setup
```

为了方便您对 UI 进行扩展,请在项目的根目录下执行以下命令,将 TUIKit 组件复制到项目中:

```
xcopy .\node_modules\tencentcloud\chat-uikit-uniapp .\TUIKit /i /e  
/exclude:.\node_modules\tencentcloud\chat-uikit-uniapp\excluded-list.txt
```

TUIKit 默认支持 [客服插件](#),需要将客服插件源码复制到项目中(请不要跳过本步骤)。

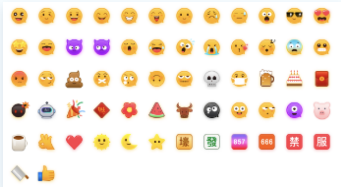
```
xcopy .\node_modules\tencentcloud\tui-customer-service-plugin .\TUIKit\tui-customer-service-plugin /i  
/e
```

步骤3: 引入 TUIKit 组件

1. 工程配置

ⓘ 说明:

为了尊重版权,IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前,请您替换为自己设计或拥有版权的其他表情包。请注意,下图所示的默认小黄脸表情包版权属于腾讯云,您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



manifest.json 文件

在 `manifest.json` 文件的源码视图中开启小程序分包 `subPackages` 和关闭 H5 `treeShaking` 选项。

```
// weixin miniProgram  
"mp-weixin" : {  
  "appid" : "",  
  "optimization" : {  
    "subPackages" : true  
  }  
},  
// H5: close treeShaking to solve the problem of uni[method]() is not a function  
"h5" : {  
  "optimization" : {  
    "treeShaking" : {  
      "enable" : false  
    }  
  }  
}
```

```
},
```

⚠ 注意:

小程序默认使用分包集成，打包小程序时 `manifest.json` 不要配置 `lazyCodeLoading` 选项。

vue.config.js (Vue3 项目可忽略)

Vue2 项目必须在根目录下创建或修改 `vue.config.js` 。

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

module.exports = {
  parallel: false,
  configureWebpack: {
    plugins: [
      ScriptSetup({
        /* options */
      }),
    ],
  },
  chainWebpack(config) {
    // disable type check and let `vue-tsc` handles it
    config.plugins.delete('fork-ts-checker');
  },
};
```

2. 集成 TUIKit

请将以下内容复制到项目对应的文件中。

App.vue 文件

```
<script lang="ts">
import { TUILogin } from '@tencentcloud/tui-core';
// #ifdef APP-PLUS || H5
import { TUIChatKit } from './TUIKit';
TUIChatKit.init();
// #endif

let vueVersion = 2;
// #ifdef VUE3
vueVersion = 3;
// #endif

// Required information
// You can get userSig from TencentCloud chat console for Testing TUIKit.
// Deploy production environment please get it from your server.
// View https://cloud.tencent.com/document/product/269/32688
uni.$SDKAppID = 0; // Your SDKAppID
uni.$userID = ''; // Your userID
uni.$userSig = ''; // Your userSig

export default {
  onLaunch: function () {
```



```
TUILogin.login({
  SDKAppID: uni.$SDKAppID,
  userID: uni.$userID,
  userSig: uni.$userSig,
  useUploadPlugin: true, // If you need to send rich media messages, please set to true.
  framework: `vue${vueVersion}` // framework used vue2 / vue3
}).catch(() => {});
}
};
</script>
<style>
/* common css for page */
uni-page-body,html,body,page {
  width: 100% !important;
  height: 100% !important;
  overflow: hidden;
}
</style>
```

pages.json 文件

```
{
  "pages": [
    {
      "path": "pages/index/index" // 您的项目首页
    }
  ],
  "subPackages": [
    {
      "root": "TUIKit",
      "pages": [
        {
          "path": "components/TUIConversation/index",
          "style": {
            "navigationBarTitleText": "腾讯云 IM"
          }
        },
        {
          "path": "components/TUIChat/index",
          "style": {
            "navigationBarTitleText": "腾讯云 IM",
            "app-plus": {
              "softinputMode": "adjustResize",
              "titleNView": {
                "buttons": [
                  {
                    "type": "menu"
                  }
                ]
              }
            }
          },
          "h5": {
            "titleNView": {
              "buttons": []
            }
          }
        }
      ]
    }
  ],
}
```

```

// 集成 chat 组件, 必须配置该路径: 视频播放
{
  "path": "components/TUIChat/video-play",
  "style": {
    "navigationBarTitleText": "腾讯云 IM"
  }
},
{
  "path": "components/TUIChat/web-view",
  "style": {
    "navigationBarTitleText": "腾讯云 IM"
  }
},
{
  "path": "components/TUIContact/index",
  "style": {
    "navigationBarTitleText": "腾讯云 IM"
  }
},
{
  "path": "components/TUIGroup/index",
  "style": {
    "navigationBarTitleText": "腾讯云 IM"
  }
},
{
  "path": "components/TUISearch/index",
  "style": {
    "navigationBarTitleText": "聊天记录"
  }
}
]
},
"preloadRule": {
  "pages/index/index": {
    "network": "all",
    "packages": ["TUIKit"]
  }
},
"globalStyle": {
  "navigationBarTextStyle": "black",
  "navigationBarTitleText": "uni-app",
  "navigationBarBackgroundColor": "#F8F8F8",
  "backgroundColor": "#F8F8F8"
}
}

```

main.js (Vue3 项目可忽略)

如果您是 Vue2 项目, 请在 main.js 中引入组合式API, 防止环境变量 `isPC` 等无法使用。

```

// #ifndef VUE3
import VueCompositionAPI from '@vue/composition-api';
Vue.use(VueCompositionAPI);
// #endif

```

3. 在项目中配置 TUIConversation 和 TUIContact 的入口

请将以下内容复制到 pages/index/index.vue 文件中。

```
<template>
  <div>
    <button @click="openConversationList">打开会话列表</button>
    <button @click="openContact">打开联系人</button>
  </div>
</template>
<script>
export default {
  methods: {
    // 打开会话列表
    openConversationList() {
      uni.navigateTo({ url: '/TUIKit/components/TUIConversation/index' });
    },
    // 打开联系人
    openContact() {
      uni.navigateTo({ url: '/TUIKit/components/TUIContact/index' });
    },
  },
};
</script>
```

步骤4: 获取 SDKAppID、userID、userSig

获取 SDKAppID、userID、userSig 信息后填写到 App.vue 中对应的字段上。

```
uni.$SDKAppID = 0; // Your SDKAppID
uni.$userID = ''; // Your userID
uni.$userSig = ''; // Your userSig
```

- SDKAppID 信息，可通过 [即时通信 IM 控制台](#) 获取：



- userID 信息，可单击 [即时通信 IM 控制台 > 账号管理](#)，切换至目标应用所在账号，创建 2 个 userID 方便后续体验聊天功能。

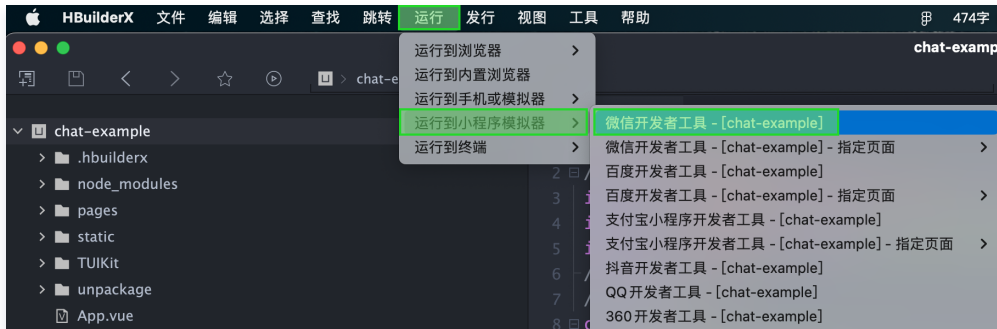


- userSig 信息，可单击 [即时通信 IM 控制台 > 开发工具 > UserSig生成&校验](#)，填写创建的 userID，即可生成 userSig。



步骤5: 启动项目

1. 使用 HBuilderX 启动该项目，单击 **运行 > 运行到小程序模拟器 > 微信开发者工具**。



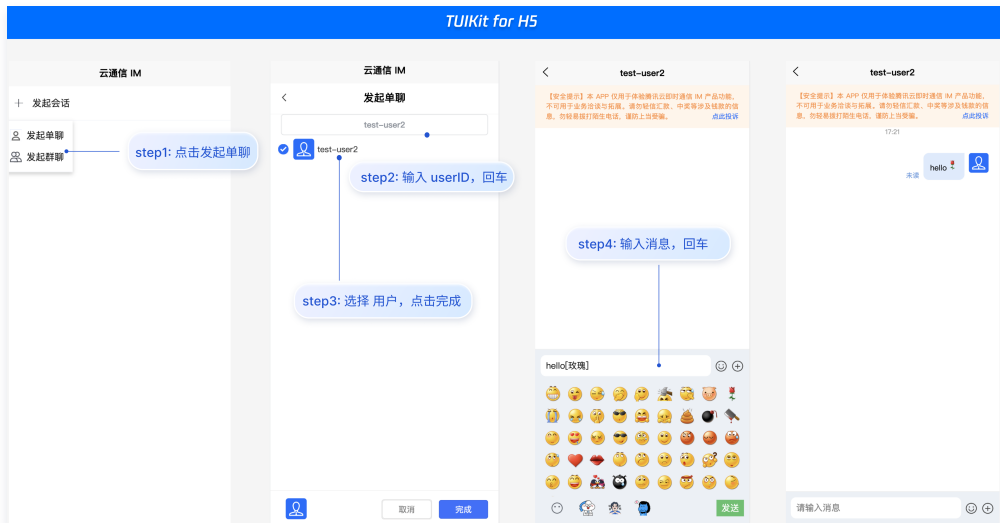
2. 如果 HBuilderX 没有自动拉起微信开发者工具，请使用微信开发者工具手动打开编译后的项目。

使用微信开发者工具打开项目根目录下的 `unpackage/dist/dev/mp-weixin` 即可。

3. 打开项目后，在微信开发者工具 **详情 > 本地设置** 中勾选 **不校验合法域名、web-view (业务域名)、TLS版本以及 HTTPS 证书**。

步骤6: 发送您的第一条消息

1. 单击 **打开会话列表** 按钮，在搜索栏输入 **步骤4** 中创建的 `userID`，选中后单击 **【完成】**，发送您的第一条消息。



步骤7: 发布前准备 & 微信小程序体积优化

发布前请删除 Debug 脚本

出于体积和安全双重因素考虑，请在发布前删除项目目录下 `/TUIKit/debug` 文件夹。在开发阶段为了方便开发，项目提供生成本地 UserSig 的脚本文件存放于 `TUIKit/debug` 文件夹中，但这并不安全，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 Demo 和功能调试。因此，请在项目发布前删除 Debug 脚本，使用后台生成 UserSig，具体原因和操作步骤请参考文档：[生成 UserSig](#)。

微信小程序体积优化

⚠ 注意：

按照标准集成流程——全量集成，TUIKit 通过 Uniapp 编译到 Vue2 和 Vue3 分别会占用不同空间（具体数值请以实际情况为准）：

- Vue2: 主包约 750KB，分包约 1.4MB。
- Vue3: 主包约 800KB，分包约 0.75MB。

微信小程序官方限制了发布时小程序包体积的大小：

- 整个小程序所有分包大小不超过 20M（开通虚拟支付后的小游戏不超过30M）
- 单个分包/主包大小不能超过 2M

因此在结合业务代码之后有可能造成小程序体积超过限制的问题，导致无法发布。

方案一：删除本地 Debug 脚本

发布前删除项目目录下 `/TUIKit/debug` 文件夹，使用后台生成 UserSig，具体原因和操作步骤请参见文档：[生成 UserSig](#)。

方案二：使用分包策略对小程序进行拆包

使用分包策略对程序进行拆包，参考微信小程序官方分包文档。需要说明的是本文档默认采取分包策略，在按步骤集成时已完成分包，主包包含 chat SDK 和 TUICore，分包包含 chat-UIKit-engine 和 TUIKIT 的各个组件，具体分包详情请参考步骤3中的 `page.json` 文件。

方案三：小程序发布前使用 HBuilderX 的发行功能进行打包

小程序发布前使用 HBuilderX 的发行功能进行打包。使用发行方式进行打包会删除代码中的 Source Map 等信息，打包体积最小。在 HBuilderX 的工具栏中依次单击发行 > 小程序-微信（仅适用于 uni-app）。



然后单击 发行，会在项目目录中的 `unpackage/dist/build/mp-weixin` 文件夹中生成微信小程序源码，再使用微信开发者工具打开该文件夹即可。



更多高级特性（强烈推荐）

仅集成聊天

如果您按照本文步骤集成 TUIKit 后，需要单独使用 TUIChat 能力，按照以下示例代码用 `uni.navigateTo` 进行跳转即可。

```
// lv1 chat: conversationID = `C2C${userID}`
// group chat: conversationID = `GROUP${groupID}`
const conversationID = '';
uni.navigateTo({ url: `/TUIKit/components/TUIChat/index?conversationID=${conversationID}` });
```

消息推送

说明:

如果您的 app 需要推送功能，推荐您接入 [uni-app 腾讯云推送服务 \(Push\)](#)，支持 iOS 和 Android 推送，同时适配各大厂商推送。对 FCM 也有完善的支持，**出海必备**。



消息云端搜索

注意:

@tencentcloud/chat-uikit-uniapp \geq 2.0.6 默认支持，只需要进入 [控制台 > 插件市场 > 云端搜索](#)，单击[免费试用](#)或[购买插件](#)即可使用消息云端搜索功能。



在线客服

注意:

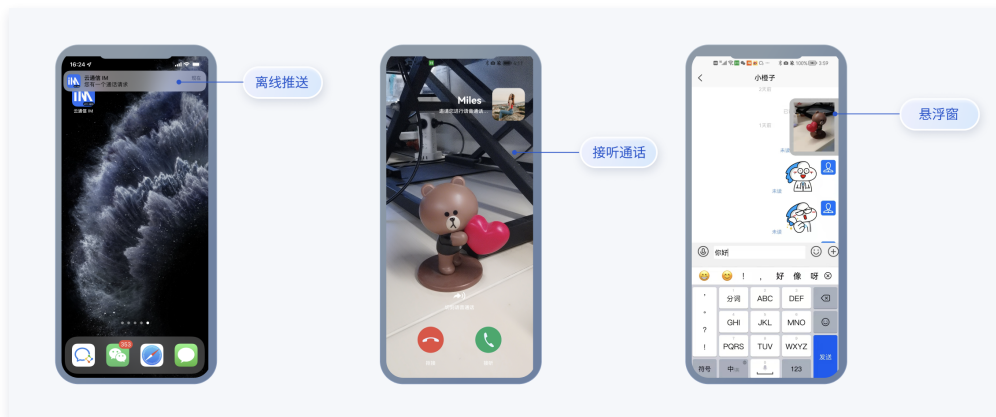
@tencentcloud/chat-uikit-uniapp \geq 2.0.0 默认支持，只需要进入 [控制台 > 插件市场 > 客服插件](#)，单击[免费试用](#)或[购买插件](#)即可使用在线客服功能。



音视频通话 TUICallKit 插件

- 说明：**
 TUICallKit 中默认没有集成 TUICallKit 音视频组件，TUICallKit 主要负责语音、视频通话。
 如果您需要集成通话功能，可参见以下文档实现。
- 打包到 APP 请参见：[音视频通话（客户端）](#)。
 - 打包到小程序请参见：[音视频通话（小程序）](#)。
 - 打包到 H5 请参见：[音视频通话（H5）](#)。

• 客户端通话示意图：



• 小程序通话示意图：



• H5 通话示意图：



云端审核

在消息发送、资料修改场景中，很有可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，更很有可能违法并导致业务被监管部门查封。

即时通信 IM 支持云端审核（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。

云端审核：在服务端检测由单聊、群聊、资料场景中产生的文本、图片、音频、视频内容，支持针对不同场景的不同内容分别配置审核策略，并对识别出的不安全内容进行拦截。此功能已提供默认预设拦截词库和审核场景，只需在 IM 控制台打开功能开关，即可直接使用。

交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

常见问题

1. 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](#) > [开发管理](#) > [开发设置](#) > [服务器域名](#) 中进行域名配置，域名配置方法请参见：[小程序 socket 合法域名](#)。

2. 更多问题请参见：[uni-app 相关问题](#)。

参考文档

TUIKit github 源码：

- [chat-uikit-uniapp \(vue2/vue3\) github 源码](#)

实现更多功能，请参见 [ChatEngine API 文档](#)：

- [ChatEngine API 手册](#)
- [ChatEngine npm](#)

React

最近更新时间: 2025-04-30 10:26:42

TUIKit 介绍

TUIKit 是基于腾讯云 IM SDK 的一款 UI 组件库，它提供了一些通用的 UI 组件，包含会话、聊天、关系链、群组、音视频通话等功能。

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。

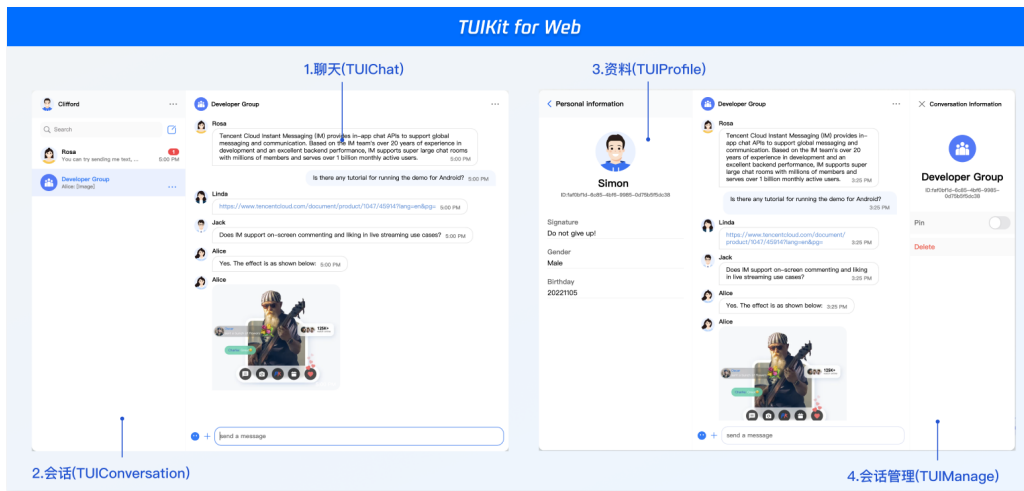
TUIKit 中的组件在实现 UI 功能的同时，会调用 IM SDK 相应的接口实现 IM 相关逻辑和数据的处理，因而开发者在使用 TUIKit 时只需关注自身业务或个性化扩展即可。

基于 React 开发的 TUIKit 界面风格更契合境外客户的使用习惯，而且支持国际化，如果您的业务有出海的需求，欢迎接入。

Example App

我们构建了可供体验的在线 [Demo](#)，并将代码开源到了 github，[chat-uikit-react](#)。

Web 端界面效果如下图所示：



开发环境要求

- React \geq v18.0
- TypeScript
- node ($12.13.0 \leq$ node 版本 \leq 17.0.0, 推荐使用 Node.js 官方 LTS 版本 16.17.0)
- npm (版本请与 node 版本匹配)

跑通demo

步骤1: 下载源码

```
# Run the code in CLI
$ git clone https://github.com/TencentCloud/chat-uikit-react
# Go to the project
$ cd chat-uikit-react/examples/sample-chat
# Install dependencies of the demo and build chat-uikit-react
$ npm install
```

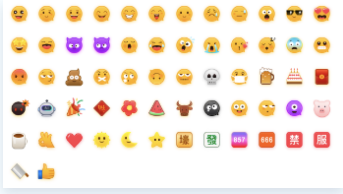
⚠ 注意:

项目 `examples/sample-chat` 下依赖的 `@tencentcloud/chat-uikit-react` 为本地包，因此需要在 `chat-uikit-react` 根目录下执行 `npm run build` 或者 `npm run start`，后者会启动 `npm run rollup -c -w`，`examples/sample-chat` 项目会实时加载修改后的组件库，建议在需要自己开发修改组件库时使用。

步骤2: 配置 demo

说明:

为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



1. 打开 `examples/sample-chat` 项目，通过路径 `./examples/sample-chat/src/debug/GenerateTestUserSig.js` 找到 `GenerateTestUserSig.js` 文件。
2. 在 `GenerateTestUserSig.js` 文件中设置 `SDKAPPID` 和 `SECRETKEY`，其值可以在 [即时通信 IM 控制台](#) 中获取。点击目标应用卡片，进入应用的基础配置页面，例如：



3. 在图中所示区域，点击 **复制**，替换 `GenerateTestUserSig.js` 文件原有的 `SDKAPPID` 和 `SECRETKEY`。

注意:

- 本文提到的生成 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 Demo 和功能调试。
- 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](#)。

4. 进入应用的账号管理页面，创建账号并获取 userID，用于当作后续发送消息的测试用户。

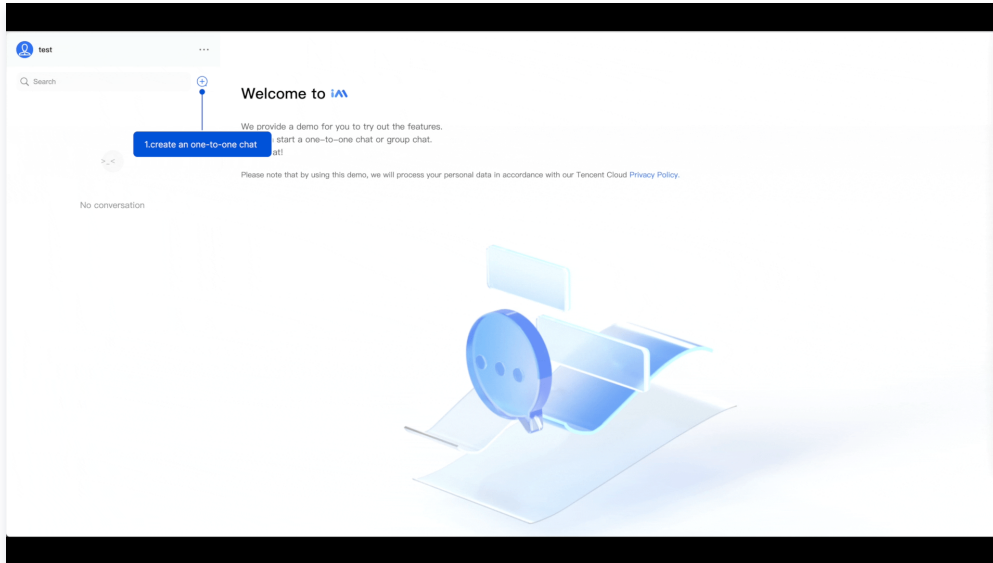


步骤3：启动项目

```
# Launch the project
$ npm run start
```

步骤4：发送您的第一条消息

1. 项目启动成功后，单击“+”图标，创建会话。
2. 在输入框中搜索另一个用户的 userID（参考：[步骤2.4](#)）。
3. 单击用户头像发起会话。
4. 在输入框输入消息，按下“enter”键发送。



步骤5：开通云端审核功能

在消息发送、资料修改等场景中，可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，还有可能违法并导致业务被监管部门查封。

即时通信 IM 支持 [云端审核](#)（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。您可直接前往 [云端审核控制台](#) 开通免费试用，每个应用可免费试用7天（含10000次免费抵扣量）。

交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

常见问题

什么是 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。

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](#)。

⚠ 注意:

本文示例代码采用的获取 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 签发方式请参见上文。

相关文档**参考文档****UIKit 相关:**

- [chat-uikit-react npm](#)
- [UIKit 快速集成文档](#)

实现更多功能，请参考 ChatEngine API 文档:

- [chat-uikit-engine API 手册](#)
- [chat-uikit-engine npm](#)

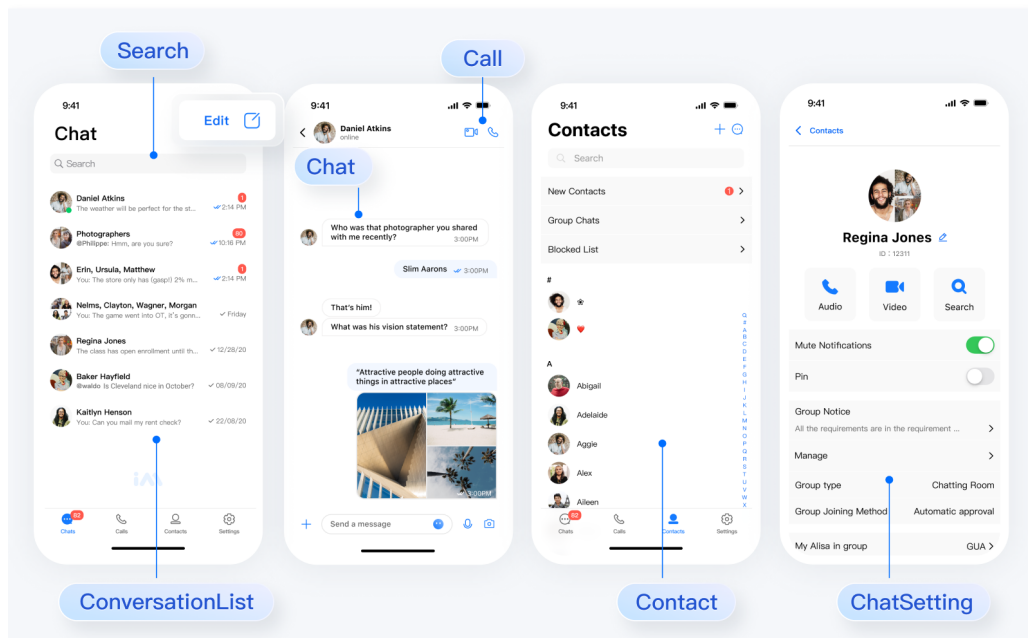
React Native

最近更新时间：2025-04-30 10:26:42

chat-uikit-react-native 介绍

chat-uikit-react-native 是一款基于腾讯云 Chat SDK 的 React Native UI 组件库，提供了一些通用的 UI 组件，包含会话、聊天、群组等功能。基于这些精心设计的 UI 组件，您可以快速构建优雅的、可靠的、可扩展的 Chat 应用。基于 React Native 开发的 UIKit 界面风格更契合境外客户的使用习惯，而且支持国际化，如果您的业务有出海的需求，欢迎接入。具体可参见 [开源代码](#)。

chat-uikit-react-native 界面效果如下图所示：



环境要求

- React Native 0.75.0
- Node.js version 18+
- JDK 17
- Xcode 版本 14.0 或更高版本
- Android Studio

配置开发环境

如果您是首次开发 React Native 项目，请参见 React Native 官网步骤 [set-up-your-environment](#) 配置开发环境。

在创建项目或编译项目过程中如果遇到环境问题，可以运行 `npx react-native doctor` 进行环境诊断。

跑通 Demo

步骤1: 下载源码

```
git clone https://github.com/TencentCloud/chat-demo-react-native
```

```
cd chat-demo-react-native/Demo
```

使用 yarn 安装 (推荐)：

```
yarn install
```

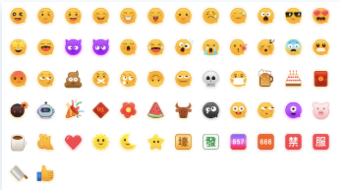
或者使用 npm 安装：

```
npm i --legacy-peer-deps
```

步骤2: 配置 Demo

说明:

为了尊重版权, IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前, 请您替换为自己设计或拥有版权的其他表情包。请注意, 下图所示的默认小黄脸表情包版权属于腾讯云, 您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



1. 打开 Demo 项目, ./debug 目录下的 GenerateTestUserSig.js 文件。
2. 在 GenerateTestUserSig.js 文件中设置 SDKAPPID 和 SECRETKEY, 其值可以在 [即时通信 IM 控制台](#) 中获取。点击目标应用卡片, 进入应用的基础配置页面, 例如:



3. 在图中所示区域, 点击复制, 替换 GenerateTestUserSig.js 文件原有的 SDKAPPID 和 SECRETKEY。

注意:

- 本文提到的生成 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 Demo 和功能调试。
- 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](#)。

4. 进入应用的账号管理页面, 创建账号并获取 userID, 用于当作后续发送消息的测试用户。



步骤3: 启动项目

- 编译运行项目您需要使用真机或模拟器, 推荐使用真机运行。您可以参见 React Native 官网 [running-on-device](#) 连接真机进行调试。

Android

1. 手机开启开发者模式, 打开 USB 调试开关。
2. 用 USB 连接手机, 推荐选择 传输文件 选项, 不要选择 仅充电 选项。
3. 确认手机连接成功后, 执行 `npm run android` 编译运行项目。

```
npm run android
```

iOS

1. 用 USB 连接手机，用 Xcode 打开工程的 ios 目录。
2. 按照 React Native 官网 [running-on-device](#) 配置签名信息。
3. 进入 ios 目录下安装依赖。

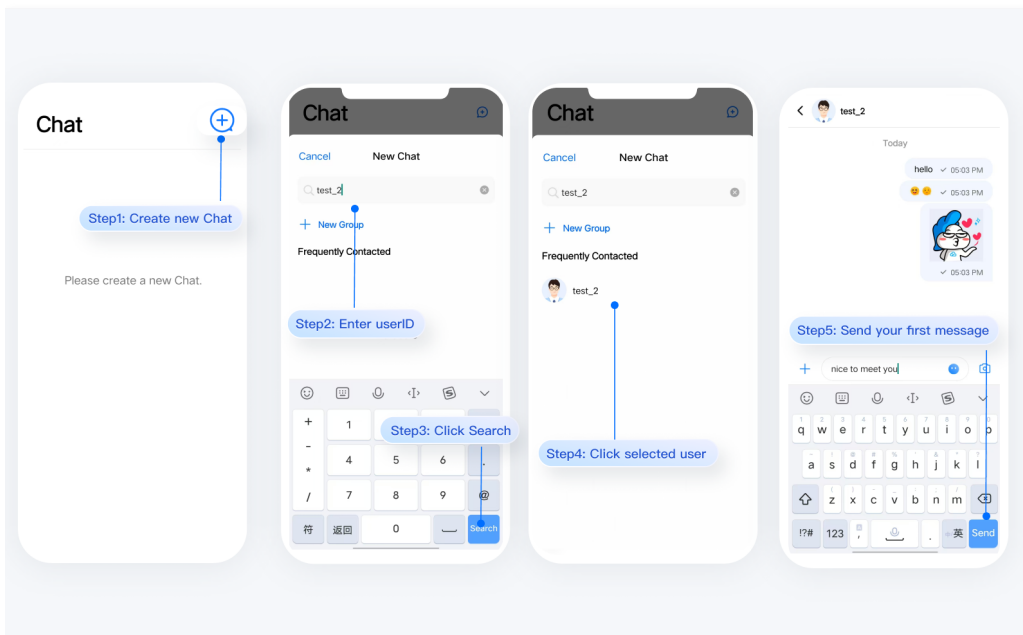
```
cd ios  
pod install
```

4. 回退到根目录，执行 npm run ios 编译运行项目。

```
cd ../  
npm run ios
```

步骤4：发送您的第一条消息

1. 项目启动之后单击左上角发起会话。
2. 进入发起会话窗口。在搜索栏输入 步骤2 中创建的 userID (test_2)，选中后打开会话。
3. 在输入框中输入消息并点击发送。



更多高级特性（强烈推荐）

消息推送

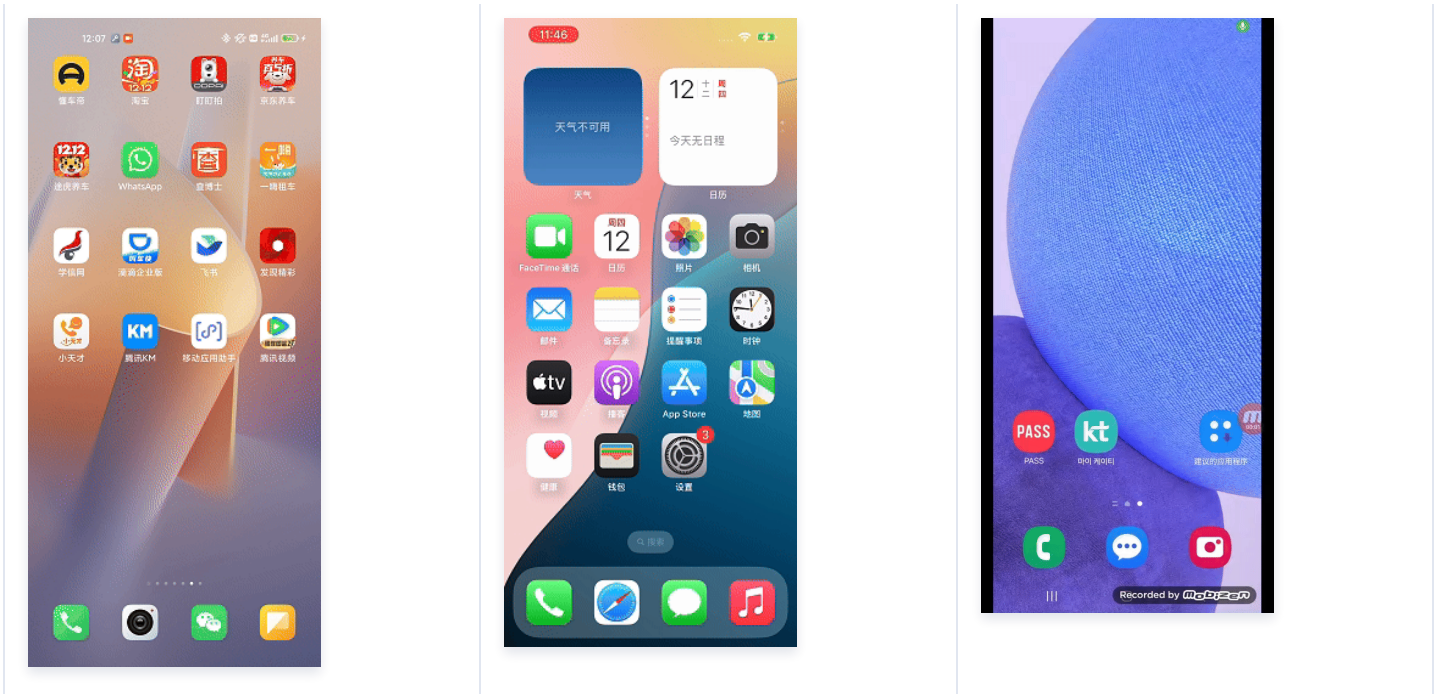
说明：

如果您的 app 需要推送功能，推荐您接入 [react native 腾讯云推送服务（Push）](#)，支持 iOS 和 Android 推送，同时适配各大厂商推送。对 FCM 也有完善的支持，出海必备。

小米 11 Pro

iPhone 13

三星 Galaxy A23 海外版（Google FCM 推送）



交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

常见问题

1. 运行 `npm run android` 提示如图所示错误时，请在项目根目录下重新设置环境变量。

```
BUILD FAILED in 2s
error Failed to install the app. Command failed with exit code 1: ./gradlew app:installDebug -
PreactNativeDevServerPort=8081 FAILURE: Build failed with an exception. * What went wrong:
Could not determine the dependencies of task ':app:compileDebugJavaWithJavac'.
> [SDK location not found.] Define a valid SDK location with an ANDROID_HOME environment variabl
```

```
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=$PATH:$ANDROID_HOME/emulator
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

2. 在 Xcode 中执行 Build 命令如果提示 node 环境变量问题，请进行如下操作：

```
[Warning] You need to configure your node_path in the `".xcode.env" file` environment. You can
set it up quickly by running: `echo export NODE_BINARY=$(command -v node) > .xcode.env` in
the ios folder. This is needed by React Native to work correctly. We fallback to the DEPRECA
TED behavior of finding `node`. This will be REMOVED in a future version. You can read more a
bout this here: https://reactnative.dev/docs/environment-setup#optional-configuring-your-envir
onment
```

```
cd ios
echo export NODE_BINARY=$(command -v node) > .xcode.env
```

参考文档

UIKit 相关：

- [chat-uikit-react-native npm](#)
- [UIKit 快速集成文档](#)

实现更多功能，请参考 ChatEngine API 文档：

- [chat-uikit-engine API 手册](#)
- [chat-uikit-engine npm](#)

Unity

最近更新时间：2025-04-30 10:26:42

通过阅读本文，您可以了解集成 Unity SDK 的方法。

环境要求

环境	版本
Unity	2019.4.15f1 及以上版本。
Android	Android Studio 3.5及以上版本，App 要求 Android 4.1及以上版本设备。
iOS	Xcode 11.0及以上版本，请确保您的项目已设置有效的开发者签名。

支持平台

我们致力于打造一套支持 Unity 全平台的即时通信 IM SDK，帮助您一套代码，全平台运行。

平台	IM SDK
iOS	支持
Android	支持
macOS	支持
Windows	支持
Web	支持，1.8.1+ 版本起

说明：
Web 平台需要简单的几步额外引入，详情请查看本文 [第五部分](#)。

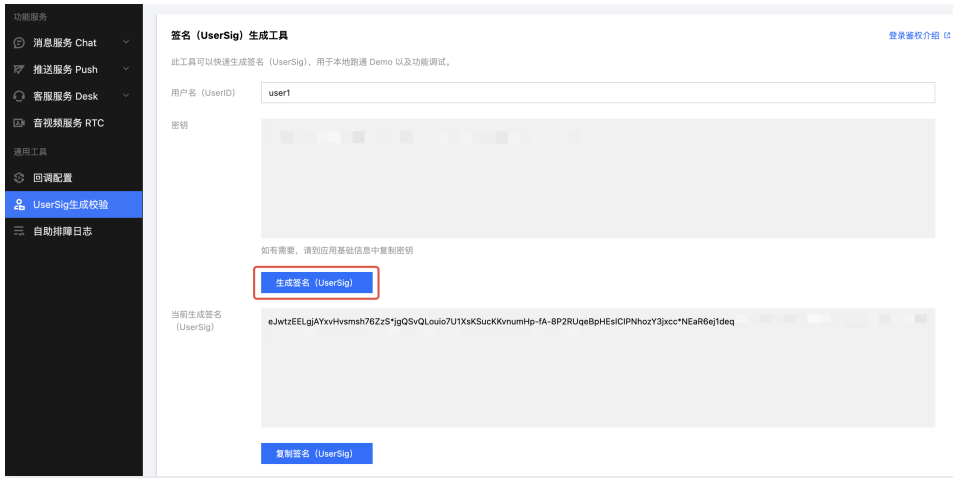
前提条件

1. 您已 [注册腾讯云](#) 账号，并完成 [实名认证](#)。
2. 参照 [创建并升级应用](#) 创建应用，并记录好 SDKAppID。

第一部分：创建测试用户

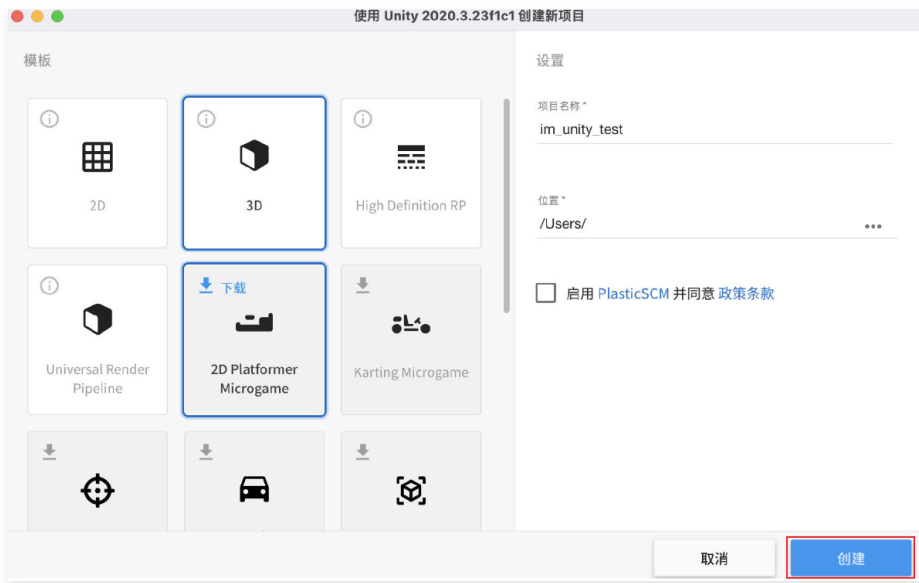
在 [IM 控制台](#) 选择您的应用，在左侧导航栏依次单击 [辅助工具](#) > [UserSig 生成&校验](#)，创建两个 UserID 及其对应的 UserSig，复制 UserID、签名 (Key)、UserSig 这三个，后续登录时会用到。

说明：
该账户仅限开发测试使用。应用上线前，正确的 UserSig 签发方式是由服务器端生成，并提供面向 App 的接口，在需要 UserSig 时由 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](#)。



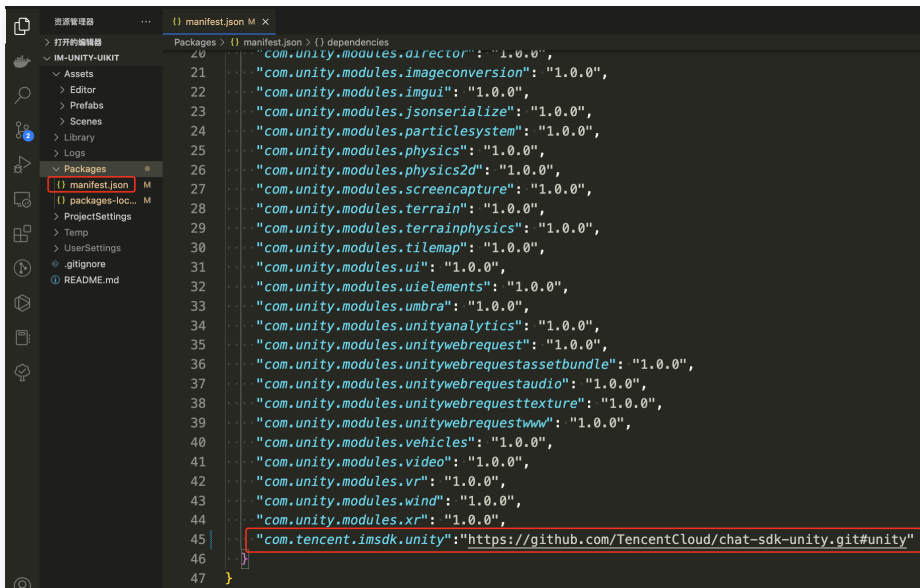
第二部分：集成 IM SDK 进您的 Unity 项目

1. 通过 Unity，创建一个 Unity 项目，并记住项目所在的位置。



或打开一个已有 Unity 项目。

2. 通过 IDE（如：Visual Studio Code）打开项目：



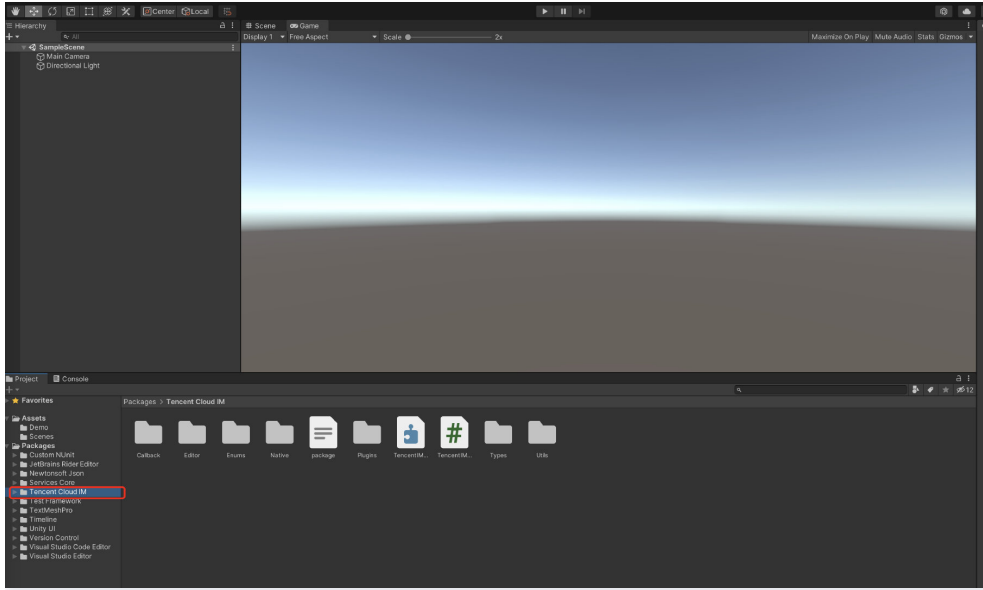
3. 根据目录，找到 Packages/manifest.json，并修改依赖如下：

```
{
  "dependencies":{
    "com.tencent.imsdk.unity":"https://github.com/TencentCloud/chat-sdk-unity.git#unity"
  }
}
```

为帮助您更好的理解 IM SDK 的各 API, 我们还提供了 [API Example](#), 演示各 API 的调用及监听的触发。

第三部分：加载依赖

在 Unity Editor 中打开项目, 等候依赖加载完毕, 确认Tencent Cloud IM 已经加载完成。



第四部分：自实现 UI 集成

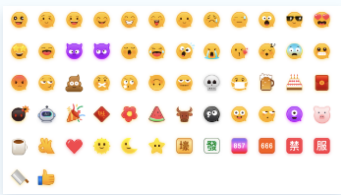
前提条件

您已经完成创建 Unity 项目, 或有可以基于的 Unity 项目, 并加载了 Tencent Cloud IM SDK。

完成 SDK 初始化

说明:

为了尊重版权, IM Demo/TUIKit 工程中默认不包含大表情元素贴图。在正式上线商用前, 请您替换为自己设计或拥有版权的其他表情包。请注意, 下图所示的默认小黄脸表情包版权属于腾讯云, 您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



本节详细文档

调用 `TencentIMSDK.Init()`, 完成 SDK 初始化。

将您的 `SDKAppID` 传入。

```
public static void Init() {
    int SDKAppID = 0; // 从即时通信 IM 控制台获取应用 SDKAppID。
    SdkConfig sdkConfig = new SdkConfig();

    sdkConfig.sdk_config_config_file_path = Application.persistentDataPath + "/TIM-Config";
}
```

```
sdkConfig.sdk_config_log_file_path = Application.persistentDataPath + "/TIM-Log";

TIMResult res = TencentIMSDK.Init(long.Parse(SDKAppID), sdkConfig);
}
```

在 `Init` 后，您可以针对 IM SDK 挂载一些监听，主要包括网络状态及用户信息变更等，详情可参见 [该文档](#)。

登录测试账户

本节详细文档

此时，您可以使用最开始的时候，在控制台生成的测试账户，完成登录验证。

调用 `TencentIMSDK.Login` 方法，登录一个测试账户。

当返回值 `res.code` 为0时，登录成功。

```
public static void Login() {
    if (userid == "" || user_sig == "")
    {
        return;
    }
    TIMResult res = TencentIMSDK.Login(userid, user_sig, (int code, string desc, string json_param, string
user_data)=>{
        // 处理登录回调逻辑
    });
}
```

说明：

该账户仅限开发测试使用。应用上线前，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](#)。

发送消息

本节详细文档

此处以发送文本消息举例

代码示例：

```
public static void MsgSendMessage() {
    string conv_id = ""; // c2c 消息会话 ID 为 userID, 群消息会话 ID 为 groupID
    Message message = new Message
    {
        message_conv_id = conv_id,
        message_conv_type = TIMConvType.kTIMConv_C2C, // 群消息为TIMConvType.kTIMConv_Group
        message_elem_array = new List<Elem>
        {
            new Elem
            {
                elem_type = TIMElemType.kTIMElem_Text,
                text_elem_content = "这是一个普通文本消息"
            }
        }
    };
    StringBuilder messageId = new StringBuilder(128);

    TIMResult res = TencentIMSDK.MsgSendMessage(conv_id, TIMConvType.kTIMConv_C2C, message, messageId,
(int code, string desc, string json_param, string user_data)=>{
        // 消息发送异步结果
    });
    // 消息发送同步返回的消息ID messageId
}
```

说明：

如果发送失败，可能是由于您的 SDKAppID 不支持陌生人发送消息，您可至控制台开启，用于测试。

[请单击此链接](#)，关闭好友关系链检查。

获取会话列表

本节详细文档

在上一个步骤中，完成发送测试消息，现在可登录另一个测试账户，拉取会话列表。

获取会话列表的方式有两种：

1. 监听长连接回调，实时更新会话列表。
2. 请求 API，根据分页一次性获取会话列表。

常见应用场景为：

在启动应用程序后立即获取会话列表，然后监听长连接以实时更新会话列表的变化。

一次性请求会话列表

```
TIMResult res = TencentIMSDK.ConvGetConvList((int code, string desc, List<ConvInfo> info_list, string user_data)=>{
    // 处理异步逻辑
});
```

此时，您可以看到您在上一步中，使用另一个测试账号，发来说话的会话。

监听长连接实时获取会话列表

您在此步骤中，需要先在 SDK 上挂载监听，然后处理回调事件，更新 UI。

1. 挂载监听。

```
TencentIMSDK.SetConvEventCallback((TIMConvEvent conv_event, List<ConvInfo> conv_list, string user_data)=>
{
    // 处理回调逻辑
});
```

2. 处理回调事件，将最新的会话列表展示在界面上。

接收消息

本节详细文档

通过腾讯云 IM SDK 接收消息有两种方式：

1. 监听长连接回调，实时获取消息变化，更新渲染历史消息列表。
2. 请求 API，根据分页一次性获取历史消息。

常见应用场景为：

1. 界面进入新的会话后，首先一次性请求一定数量的历史消息，用于展示历史消息列表。
2. 监听长连接，实时接收新的消息，将其添加进历史消息列表中。

一次性请求历史消息列表

每页拉取的消息数量不能太大，否则会影响拉取速度。建议此处设置为20左右。

您应该动态记录当前页数，用于下一轮请求。

示例代码如下：

```
// 拉取单聊历史消息
// 首次拉取，msg_getmsglist_param_last_msg 设置为 null
// 再次拉取时，msg_getmsglist_param_last_msg 可以使用返回的消息列表中的最后一条消息
var get_message_list_param = new MsgGetMsgListParam
{
    msg_getmsglist_param_last_msg = LastMessage
};
```

```
TIMResult res = TencentIMSDK.MsgGetMsgList(conv_id, TIMConvType.kTIMConv_C2C, get_message_list_param, (int code, string desc, string user_data) => {  
    // 处理回调逻辑  
});
```

```
// 拉取群聊历史消息  
// 首次拉取, msg_getmsglist_param_last_msg 设置为 null  
// 再次拉取时, msg_getmsglist_param_last_msg 可以使用返回的消息列表中的最后一条消息  
var get_message_list_param = new MsgGetMsgListParam  
{  
    msg_getmsglist_param_last_msg = LastMessage  
};  
TIMResult res = TencentIMSDK.MsgGetMsgList(conv_id, TIMConvType.kTIMConv_Group, get_message_list_param,  
(int code, string desc, string user_data) => {  
    // 处理回调逻辑  
});
```

监听长链接实时获取新消息

历史消息列表初始化后, 新消息来自长链接 `TencentIMSDK.AddRecvNewMsgCallback`。

`AddRecvNewMsgCallback` 回调被触发后, 您可以按需将新消息添加进历史消息列表中。

绑定监听器示例代码如下:

```
TencentIMSDK.AddRecvNewMsgCallback((List<Message> message, string user_data) => {  
    // 处理新消息  
});
```

此时, 您已基本完成 IM 模块开发, 可以发送接收消息, 也可以进入不同的会话。

您可以继续完成 [群组](#), [用户资料](#), [关系链](#), [本地搜索](#) 等相关功能开发。

详情可查看 [自实现 UI 集成 SDK 文档](#)。

第五部分: #Unity for WebGL 支持

Tencent Cloud IM SDK (Unity 版本) 自 `1.8.1` 版本起支持构建 WebGL。

相比 Android 和 iOS 端, 需要一些额外步骤。如下:

引入 JS

从 [Npm 下载](#) 下方三个 JS 文件 (需要安装 nodejs, 可参考 [nodejs 官网](#)), 放置于项目构建 WebGL 产物的文件夹内。

- `index.js`
- `modules/group-module.js`
- `modules/relationship-module.js`
- `modules/signaling-module.js`
- 打开 `index.html`, 并引入这四个 JS 文件。如下:

```
<script src="./index.js"></script>  
<script src="./modules/group-module.js"></script>  
<script src="./modules/relationship-module.js"></script>  
<script src="./modules/signaling-module.js"></script>
```

可选操作: 开通内容审核功能

在消息发送、资料修改场景中, 很有可能会扩散不合适的内容, 特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容, 不仅严重损害了用户们的身心健康, 更很有可能违法并导致业务被监管部门查封。

即时通信 IM 支持内容审核 (反垃圾信息) 功能, 可针对不安全、不适宜的内容进行自动识别、处理, 为您的产品体验和业务安全保驾护航。可以通过 [云端审核](#) 方式来实现。

常见问题

支持哪些平台？

目前支持 iOS、Android、Windows、Mac 和 WebGL。

Android 单击 Build And Run 报错找不到可用设备？

确保设备没被其他资源占用，或单击 Build 生成 apk 包，再拖动进模拟器里运行。

iOS 第一次运行报错？

按照上面的 Demo 运行配置后，如果报错，可以单击 **Product**>**Clean**，清除产物后重新 Build，或者关闭 Xcode 重新打开再次 Build。

2019.04版 Unity, iOS 平台报错？

Library/PackageCache/com.unity.collab-proxy@1.3.9/Editor/UserInterface/Bootstrap.cs(23,20): error CS0117: 'Collab' does not contain a definition for 'ShowChangesWindow'

在 Editor 工具栏单击 **Window** > **Package Manager**，将 Unity Collaborate 降级到 1.2.16。

2019.04版 Unity, iOS 平台报错？

Library/PackageCache/com.unity.textmeshpro@3.0.1/Scripts/Editor/TMP_PackageUtilities.cs(453,84): error CS0103: The name 'VersionControlSettings' does not exist in the current context

打开源码，把 `|| VersionControlSettings.mode != "Visible Meta Files"` 这部分代码删除即可。

这是 C# 接口吗？如何脱离 unity 使用？

Unity SDK 是使用 C# 的 SDK，但由于 Unity SDK 包含 Unity 特性，不能直接在纯 C# 的环境下使用。

若需要能在 C# 环境下使用，我们提供单独的 [C# SDK](#) nuget 包。使用方法与 Unity SDK 相同，可直接参考 Unity SDK 文档使用。

其中，纯 C# SDK 只支持 PC 端，unity SDK 支持移动端。

有可以直接使用的 UI 吗？

现在暂不提供 unity SDK 和 C# SDK 相应的 UIKit。

错误码如何查询？

IM SDK 的 API 层面错误码，请查看 [该文档](#)。

联系我们

[点此进入IM社群](#)，享有专业工程师的支持，解决您的难题。

UE

最近更新时间：2025-04-30 10:26:42

本文旨在介绍如何将腾讯云即时通信 IM 服务集成进 Unreal Engine 5 (虚幻引擎 5, 简称 UE5) 平台的应用或游戏中。

步骤1: 创建应用

1. 登录 [即时通信 IM 控制台](#)。

说明:

- 如果您已有应用, 请记录其 SDKAppID 和密钥信息, 直接跳到 [步骤2: 安装 UE5 并创建新项目](#)。
- 单个腾讯云账号最多可创建300个应用。若已有300个应用, 您可以先 [停用并删除](#) 不再使用的应用后, 再创建新的应用。应用删除后, 该 SDKAppID 对应的所有数据和服务不可恢复, 请谨慎操作。

2. 在 [应用管理](#) 页面, 单击 [创建新应用](#), 在弹出的对话框中输入应用名称, 选择合适的数据中心, 单击 [确定](#)。



3. 创建完成后, 可在 [应用管理](#) 页面查看、搜索、管理应用。请记录 [SDKAppID](#) 和 [密钥信息](#), 在后续使用 [初始化](#) 和 [登录](#) 接口时将会用到。

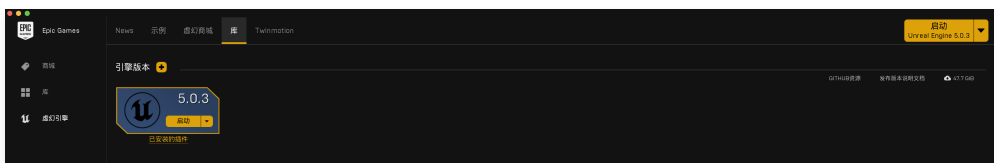
应用名称	SDKAppID	应用版本	服务状态	数据中心	创建时间	到期时间	标签	操作
test1	1600XXXXXX	体验版	使用中	中国	2023-12-28	2024-01-28	-	应用详情 版本对比 查看密钥 标签管理
test2	1600XXXXXX	体验版	使用中	中国	2023-12-15	-	-	应用详情 版本对比 查看密钥 标签管理
test3	1600XXXXXX	体验版	使用中	中国	2023-12-15	-	-	应用详情 版本对比 查看密钥 标签管理

注意:

密钥信息为敏感信息, 为防止他人盗用, 请妥善保管, 谨防泄露。

步骤2: 安装 UE5 并创建新项目

1. 安装 [Epic Games Launcher](#), 并且通过 Epic 安装 Unreal Engine 5.0.0+版本, 安装好如图所示:



2. 启动 UE5, 创建 UE5 应用, 如图所示:



其中项目默认设置中蓝图和 C++、目标平台的配置，可根据业务方自己的需求进行配置，腾讯云 IM UE5 Plugin 对此没有限制。

3. 创建好应用后，项目目录如图所示：

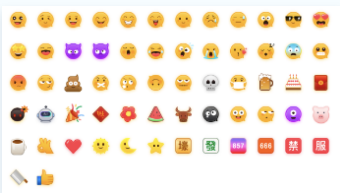


步骤3: 下载 Plugin

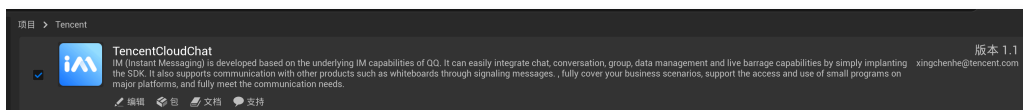
```
// 将插件从GitHub下来
git clone https://github.com/TencentCloud/chat-plugin-ue5.git
```

步骤4: Plugin 引入

说明：
为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 **IM 企业版套餐** 免费使用该表情包。



在项目根目录创建插件目录，命名叫 Plugins，将步骤三中下载的插件包（Plugins/下的 TencentCoudChat 文件夹）引入其中，引入后可在 UE5 Editor 的 **编辑 > 插件模块查看** 到如图所示插件安装：



插件引入后，在项目主模块中的 Build.cs 文件中，引入插件模块，如图所示：

```

Source > UE5_Plugin_Mobile > UE5_Plugin_Mobile.Build.cs
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 using UnrealBuildTool;
4
5 public class UE5_Plugin_Mobile : ModuleRules
6 {
7     public UE5_Plugin_Mobile(ReadOnlyTargetRules Target) : base(Target)
8     {
9         PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;
10
11         PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "HeadMountedDisplay", "EnhancedInput", "TencentCloudChat" });
12     }
13 }
14

```

步骤5: 头文件引入

在业务 CPP 文件中引入 TencentCloudChat.h 后，开始使用腾讯云 IM UE5 Plugin 提供的能力。

步骤6: 测试验证

验证代码如图所示：

```

void AUE5_Plugin_MobileCharacter::BeginPlay()
{
    // Call the base class
    Super::BeginPlay();

    V2IMString version = TencentCloudChat::GetVersion();

    const FString text = FString::FromStr(static_cast<FString>(version.CString()));

    FMessageDialog::Open(EAppMsgType::Ok, text);
    // you can use other Tencent cloud chat sdk here

    // V2IMSDKConfig timConfig;

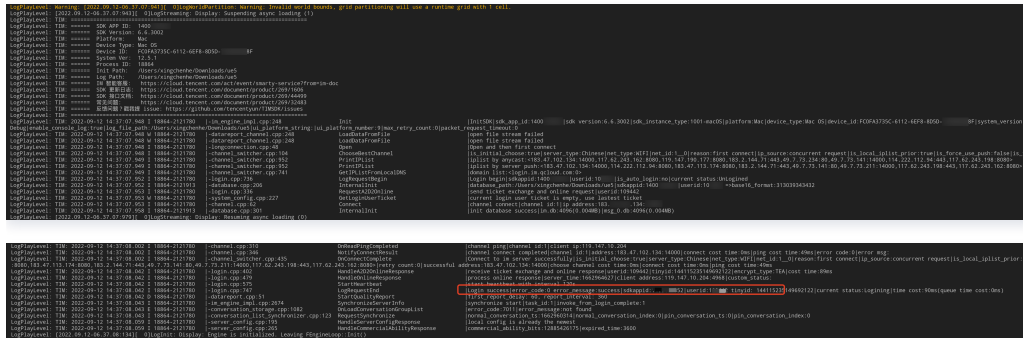
    // timConfig.InitPath = static_cast<V2IMString>("");
    // timConfig.LogPath = static_cast<V2IMString>("");

    // TencentCloudChat::InitSDK(0,timConfig);

    //Add Input Mapping Context
    if (APlayerController* PlayerController = Cast<APlayerController>(Controller))
    {
        if (UEnhancedInputLocalPlayerSubsystem* Subsystem = ULocalPlayer::GetSubsystem<UEnhancedInputLocalPlayerSubsystem>(PlayerController->GetLocalPlayer()))
        {
            Subsystem->AddMappingContext(DefaultMappingContext, 0);
        }
    }
}

```

其中初始化用到的 SDKAppId 在步骤一中获得，登录使用到的 UserSig，可在 控制台辅助工具 生成。在生产环境，UserSig 由业务侧后台生成。本篇入门教程只演示了初始化、登录、注册消息监听事件的 API 调用，完整的 API 文档可以查看 此处。UE5 各个平台均集成的是 C++ SDK，且所有接口一致。在 UE5 控制台中见到以下日志，代表 plugin 集成成功：



可选操作: 开通内容审核功能

在消息发送、资料修改场景中，很有可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，更很有可能违法并导致业务被监管部门查封。

即时通信 IM 支持内容审核（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。可以通过 云端审核 方式来实现。

相关文档

- API 文档
- UE5 官方文档

常见问题

1. 在 mac 调试环境，如遇到提示文件已经损坏，可执行如下命令修复：

```
sudo xattr -r -d com.apple.quarantine $pathToFrameWork
```

2. 初始化传入的日志和 db 目录，需要应用有读写权限。
3. 插件暂不支持 win32 和 Mac M1 进行调试和编译。

联系我们

[点此进入 IM 社群](#)，享有专业工程师的支持，帮助解决您的难题。

Flutter

最近更新时间：2025-05-27 16:43:02



欢迎使用腾讯云 IM Flutter 相关集成产品，通过阅读本文，您可以了解集成 Flutter UIKit 组件库的方法。

环境要求

- Android Studio Dolphin | 2021.3.1 及以上版本，或参考官网[环境配置](#)。
- Flutter 3.24.0 及以上

操作步骤

步骤1: 创建应用

1. 登录 [即时通信 IM 控制台](#)。如果您已有应用，请记录其 SDKAppID 信息，并直接跳到 [步骤2: 获取 SDKAppID 和密钥信息](#) 参见后续步骤进行操作。
2. 在应用管理页面，单击创建新应用。



3. 输入应用名称、选择合适的数据中心，单击确定即可完成应用的创建。

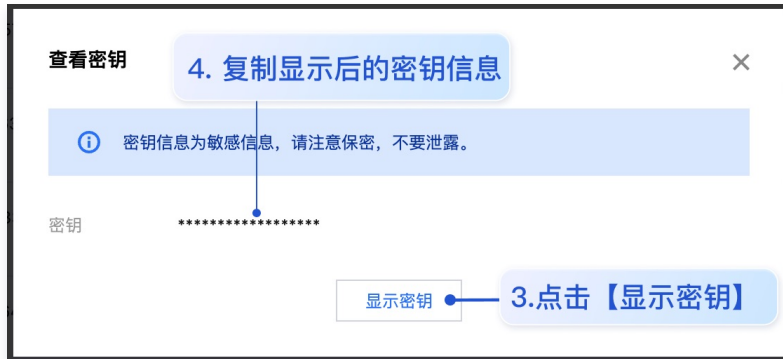


步骤2: 获取 SDKAppID 和密钥信息

1. 在应用管理页面的 SDKAppID 列获取 SDKAppID 信息。



2. 在操作列单击查看密钥，随后在弹出的对话框中，单击显示密钥，复制显示后的密钥信息。



注意：

密钥信息为敏感信息，为防止他人盗用，请妥善保管，谨防泄露。

步骤3：下载并配置 Demo 源码

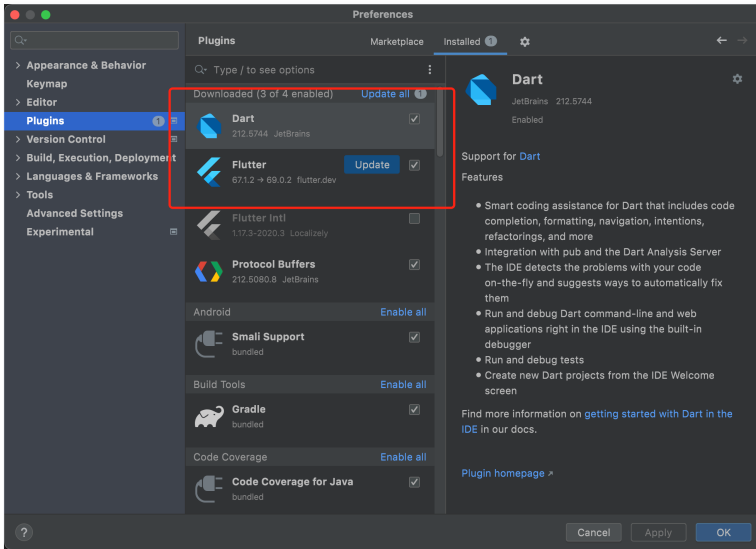
1. 下载即时通信 Flutter IM Demo 工程，具体下载地址请参见 [经典风格源码地址](#) 或者 [简约风格源码地址](#)。
2. 打开终端目录的工程，找到对应的 `config` 文件，路径为 `lib/config.dart`。
3. 设置 `config` 文件中的相关参数：
 - `sdkAppID`: 请设置为 **步骤1** 中获取的实际应用 SDKAppID。
 - `key`: 请设置为 **步骤2** 中获取的实际密钥信息。

注意：

本文提到的获取 UserSig 的方案是在客户端代码中配置 `key` 并由 `GenerateUserSig.dart` 计算得出，该方法中 `key` 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 Demo 和功能调试。
 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：编译运行

用 Android Studio 导入工程，并安装 Flutter 和 Dart 插件。



在项目根目录执行如下命令安装依赖后，编译运行即可。

```
flutter pub get
```

步骤5: 开通内容审核功能

在消息发送、资料修改等场景中，可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，还有可能违法并导致业务被监管部门查封。

即时通信 IM 支持 **云端审核**（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。您可直接前往 [云端审核控制台](#) 开通免费试用，每个应用可免费试用7天（含10000次免费抵扣量）。

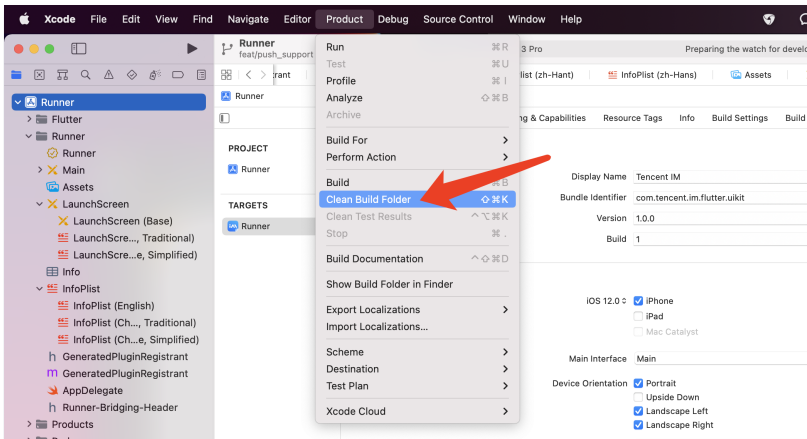
常见问题

1. iOS 端 Pods 依赖无法安装成功

方案一：手动删除 `ios/Pods` 文件夹，及 `ios/Podfile.lock` 文件，并执行如下命令，重新安装依赖。

```
cd ios
sudo gem install ffi
pod install --repo-update
```

方案二：配置运行后，如果报错，可以单击 **Product > Clean Build Folder**，清除产物后重新运行 `pod install` 或 `flutter run`



2. 佩戴 Apple Watch 时，真机调试 iOS 报错

```
Ineligible destinations for the "Runner" scheme:
{ platform:iOS, id:00008020-000A61AA268002E, name:王润霖的 iPhone, error:Device is busy (Preparing the watch for development via 王润霖的 iPhone) }
{ platform:iOS, id:dvtdevice-DVTiPhonePlaceholder-iphonesimulator:placeholder, name:Any iOS Device }
{ platform:iOS Simulator, id:dvtdevice-DVTiOSDeviceSimulatorPlaceholder-iphonesimulator:placeholder, name:Any iOS Simulator Device }
```

请将您的 Apple Watch 调整至飞行模式，并将 iPhone 的蓝牙功能通过 **设置 => 蓝牙** 彻底关闭。

重新启动 Xcode (若打开), 并重新运行 `flutter run` 即可。

3. Flutter 环境问题

如您需得知 Flutter 的环境是否存在问题, 请运行 Flutter doctor 检测 Flutter 环境是否装好。

4. 使用 Flutter 自动生成的项目, 引入TUIKit后, 运行 Android 端报错

```
Flutter Fix
The plugin camera requires a higher Android SDK version.
Fix this issue by adding the following to the file D:\IM\IMEST\flutter_application_1\android\app\build.gradle:
android {
  defaultConfig {
    minSdkVersion 21
  }
}
```

1. 打开 `android\app\src\main\AndroidManifest.xml`, 根据如下, 补充 `xmlns:tools="http://schemas.android.com/tools"` / `android:label="@string/android_label"` 及 `tools:replace="android:label"`。

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="替换成您的 Android 端包名"
  xmlns:tools="http://schemas.android.com/tools">
  <application
    android:label="@string/android_label"
    tools:replace="android:label"
    android:icon="@mipmap/ic_launcher" // 指定一个 icon 路径
    android:usesCleartextTraffic="true"
    android:requestLegacyExternalStorage="true">
```

2. 打开 `android\app\build.gradle`, 补充 `defaultConfig` 中 `minSdkVersion` 及 `targetSdkVersion`。

```
defaultConfig {
  applicationId " " // 替换成您的Android端包名
  minSdkVersion 21
  targetSdkVersion 30
}
```

5. Windows 11 运行 Demo 报错怎么办?

报错一:

```
Error waiting for a debug connection: The log reader stopped unexpectedly, or never started.
Error launching application on Windows.
```

1. 首先, 将项目名称中的横线改为下划线。例如, 将 "im-flutter-uikit-full-platform" 改为 "im_flutter_uikit_full_platform"。
2. 删除根目录下的 "windows" 文件夹。
3. 在命令行中运行以下命令: `flutter create . --org=com.tencent.im.flutter`。这将创建一个新的 Flutter 项目, 并将其组织名称设置为 "com.tencent.im.flutter"。
4. 运行 `flutter clean` 命令以清除缓存。
5. 运行 `flutter pub get` 命令以获取项目所需的依赖项。
6. 最后, 运行 `flutter run -d windows` 命令以在 Windows 平台上运行应用程序。

通过按照上述步骤操作, 您可以尝试在新项目中运行应用程序, 以确定问题是否出在您的项目中。如果新项目可以正常工作, 则可以尝试将新项目中的 `pubspec.yaml` 文件复制到原来不工作的项目中, 并保存以便 Flutter 可以运行新的 `pubspec.yaml` 代码。然后, 您可以返回到旧的 `pubspec.yaml` 文件, 这样它就可以正常工作了。

报错二:

```
Nuget.exe not found, trying to download or use cached version.
```

这个提示表明您的系统中没有找到 `nuget.exe`, 因此程序尝试下载或使用缓存的版本。`nuget.exe` 是 NuGet 包管理器的命令行工具, 用于在 .NET 项目中管理依赖项。

要解决此问题, 您可以手动下载并安装 `nuget.exe`。以下是操作步骤:

1. 访问 NuGet 官方网站的下载页面: [页面链接](#)。
2. 在 "Command-line" 部分, 找到 "Windows x86 Commandline", 点击 "Download" 下载 `nuget.exe`。

3. 将下载的 `nuget.exe` 文件保存到一个合适的位置，例如 `C:\NuGet`。
4. 将 `nuget.exe` 的路径添加到系统的环境变量 `PATH` 中。这样，您就可以在命令行中全局访问 `nuget.exe`。
6. 错误码如何查询？
 - IM SDK 的 API 层面错误码，请查看 [错误码](#)。
 - TUIKit 的场景码，用于界面弹窗提示，通过 [onTUIKitCallbackListener 监听](#) 获得。

联系我们

[点此进入IM社群](#)，享有专业工程师的支持，解决您的难题。

HarmonyOS

最近更新时间：2025-06-12 10:52:42

本文主要介绍如何快速运行腾讯云即时通信 IM Demo（HarmonyOS）。

环境要求

- DevEco Studio 4.0 Release 及以上。
- HarmonyOS SDK API 10 及以上。

操作步骤

步骤 1：创建应用

1. 登录 [即时通信 IM 控制台](#)。

① 说明：

- 如果您已有应用，请记录其 SDKAppID 信息，直接跳到 [步骤2：获取 SDKAppID 和密钥信息](#)。
- 单个腾讯云账号最多可创建 300 个应用。若已有 300 个应用，您可以先 [停用并删除](#) 不再使用的应用后，再创建新的应用。应用删除后，该 SDKAppID 对应的所有数据和服务不可恢复，请谨慎操作。

2. 在应用管理页面，单击创建新应用。



3. 输入应用名称、选择合适的数据中心，单击确定即可完成应用的创建。

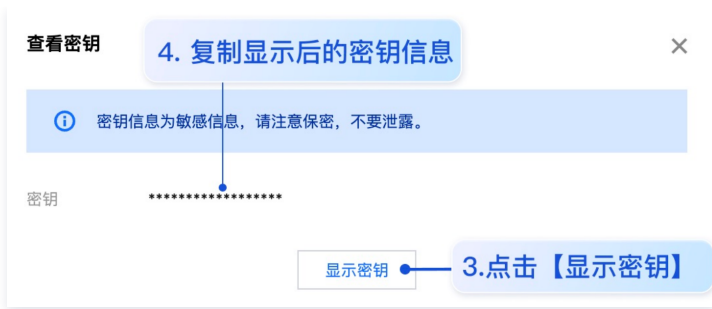


步骤 2：获取 SDKAppID 和密钥信息

1. 在应用管理页面的 SDKAppID 列获取 SDKAppID 信息。



2. 在操作列单击查看密钥，随后在弹出的对话框中，单击显示密钥，复制显示后的密钥信息。

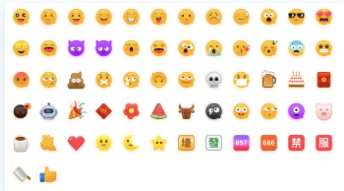


注意：
密钥信息为敏感信息，为防止他人盗用，请妥善保管，谨防泄露。

步骤 3：下载并配置 Demo 源码

1. 下载即时通信 IM Demo 工程，具体下载地址请参见 [SDK 下载](#)。

说明：
为了尊重版权，IM Demo/TUIKit 工程中默认不包含大表情元素切图。在正式上线商用前，请您替换为自己设计或拥有版权的其他表情包。请注意，下图所示的默认小黄脸表情包版权属于腾讯云，您可以通过升级至 [IM 企业版套餐](#) 免费使用该表情包。



2. 打开所属终端目录的工程，找到对应的 `GenerateTestUserSig.ts` 文件（路径：`TIMSDK/HarmonyOS/Demo/entry/src/main/ets/tools/GenerateTestUserSig.ts`）。

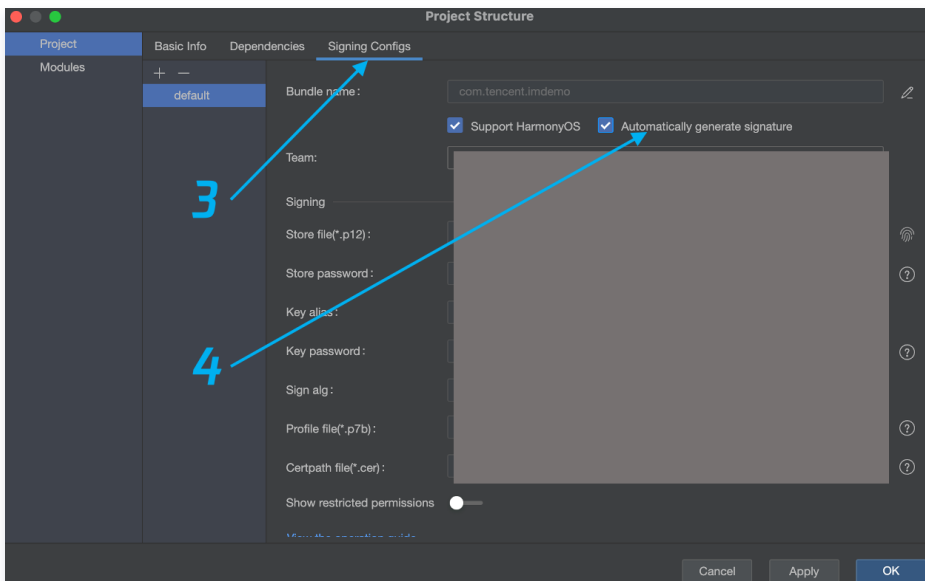
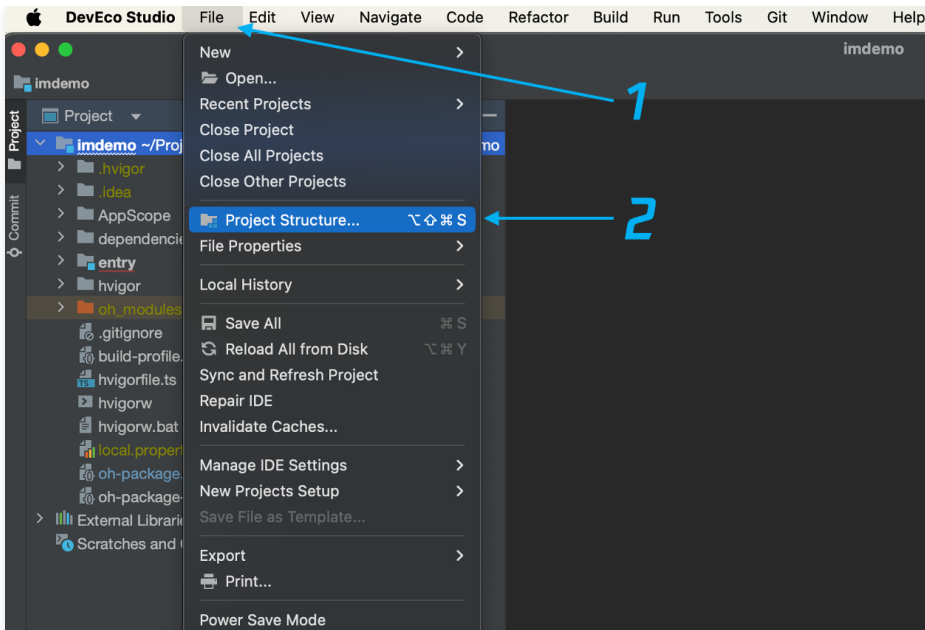
3. 设置 `GenerateTestUserSig.ts` 文件中的相关参数：
- SDKAPPID: 请设置为 [步骤1](#) 中获取的实际应用 SDKAppID。
 - SECRETKEY: 请设置为 [步骤2](#) 中获取的实际密钥信息。



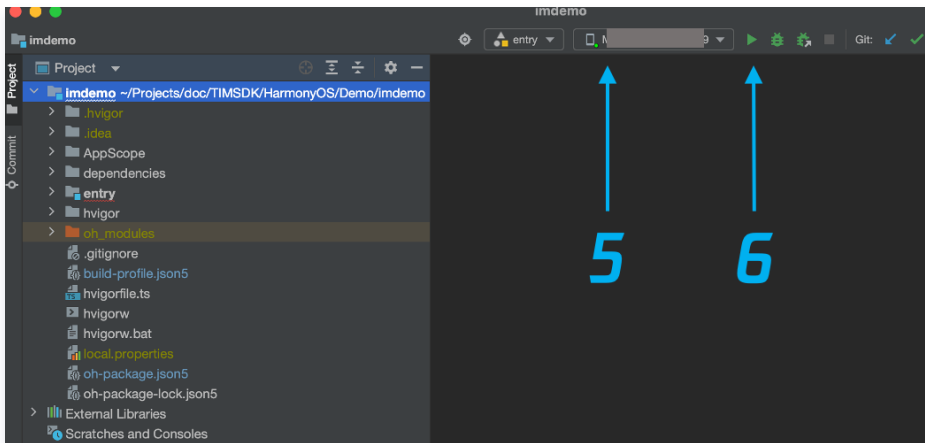
注意：
本文提到的获取 UserSig 的方案是在客户端代码中配置 SECRETKEY，该方法中 SECRETKEY 很容易被反编译逆向破解，一旦您的密钥泄露，攻击者就可以盗用您的腾讯云流量，因此该方法仅适合本地跑通 Demo 和功能调试。正确的 UserSig 签发方式是将 UserSig 的计算代码集成到您的服务端，并提供面向 App 的接口，在需要 UserSig 时由您的 App 向业务服务器发起请求获取动态 UserSig。更多详情请参见 [服务端生成 UserSig](#)。

步骤 4：编译运行

1. 打开 imdemo（路径：`TIMSDK/HarmonyOS/Demo`），配置签名：



2. 连接真机，编译运行：



步骤5：开通云端审核功能

在消息发送、资料修改等场景中，很有可能会扩散不合适的内容，特别是与敏感事件/人物相关、黄色不良内容等令人反感的内容，不仅严重损害了用户们的身心健康，更很有可能违法并导致业务被监管部门查封。

即时通信 IM 支持 [云端审核](#)（反垃圾信息）功能，可针对不安全、不适宜的内容进行自动识别、处理，为您的产品体验和业务安全保驾护航。您可直接前往 [云端审核控制台](#) 开通免费试用，每个应用可免费试用7天（含10000次免费抵扣量）。

交流与反馈

[点此进入 IM 社群](#)，享有专业工程师的支持，解决您的难题。

相关链接

- [集成 IM SDK \(HarmonyOS\)](#)
- [API 文档 \(HarmonyOS\)](#)