

# 即时通信 IM 快速入门





【版权声明】

#### ©2013-2024 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】

## 🕗 腾讯云

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

快速入门 快速入门(Android) 快速入门(iOS) 快速入门(Web & H5 Vue2/Vue3) 快速入门(Web & H5 Vue2/Vue3) 快速入门(Unitapp vue2/vue3) 快速入门(React) 快速入门(React) 快速入门(React Native) 快速入门(React Native) 快速入门(Electron) 快速入门(Electron) 快速入门(Unity) 快速入门(Unity) 快速入门(Unity) 快速入门(UE) 快速入门(Flutter) 快速入门(Flutter) 快速入门(APICloud) 快速入门(HarmonyOS)



## 快速入门 快速入门(Android)

```
最近更新时间: 2024-04-09 15:26:11
```

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。



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。

接入	▼
接入	•
标签值	×
	标签值

## 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。

- 正 应用管理	创建新应用	申请开发版					请输入 SDKApp	ID 或 应用名称 或 标签	Q
管理与配置	应用名称 SDK	(AppID 应用版本 ④	服务状态	数据中心 ▼	创建时间	到期时间 🕄	标签 🕄	操作	
品 1% 见	1400	0 体验版	使用中 🛈	中国	2023-08-16	-	-	应用详情 版本对比 查看密钥 标签管理	
器 群组管理	1400	0 体验版	使用中	中国	2023-04-24	-	-	应用详情 版本对比 查看密钥 标签管理	
號 功能配置 ∨	1.获取 SD	KAppID 信息	使用中	中国	2023-03-20	-	-	应 2. 点击【查看密钥】	
③ 回调配置		in the level							

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。



查看密钥	4. 复制显示后的密钥信息 ×
() 密钥信	息为敏感信息,请注意保密,不要泄露。
密钥	······· <sub>显示密钥</sub> ●── <mark>3.点击【显示密钥】</mark>

▲ 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。

① 说明 为尊重表情设计版权,下载的 Demo 工程中不包含大表情元素切图,您可以使用自己本地表情包来配置代码。未授权使用 IM Demo 中的表情包可能 会构成设计侵权。

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,霍麦普换为您自己账号下的 SDKAppId。 *		
<ul> <li>・ 进入攝訊云云通信[12制台](https://console.cloud.tencent.com/avc ) 创稿应用, 即可叠到 5DKAppId,</li> <li>* 它是顯訊云用于区分案户的唯一标识。</li> </ul>	基本信息	
public static final int SDKAPPID = 0;	SDKAppID	1400
	状态	启用 ⊘ 停用
	应用名称	test
* 金名辺邦时间,建议不要设置的辺辺 *	应用类型	视频
* 时间单位: 秒		
* 法认识问: 7 x 24 x 60 x 60 = 604800 = 7 大 */		
private static final int EXPIRETIME = 604800;		
/** * 21/28/28/->100/04-0-0000000000000000000000000000000		
。 计具金名用的加密密码,获取步骤观下: *		
* step1. 进入腾讯云云通信[控制台](https://console.cloud.tencent.com/avc ) ,如果还没有应用就创建一个,		
* step2. 举击"应用赋监"进入基础赋监反曲,开进一步视到"恢告体系集队"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,请将其拷贝并复制到如下的变量中		
	应用名称	test
<ul> <li>注意: 该方案に200月丁頃にUemo, 正式上线削損待 UserS1g 计算代码和密钥定移到2019后首册方容上, 以避免加密密用電路等数的端重盈用。</li> <li>文档: https://cloud.tencent.com/document/product/269/32688#Server</li> </ul>	应用类型	视频
*/	欧阳	72c9
proce source ration derang sector recti		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。



## 快速入门(iOS)

最近更新时间: 2024-04-09 15:26:11

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。
  - 🕛 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**。

即时通信 IM	应用管理 Ⅲ	技术服务交流群 🖸 🛛 IM	出海交流专区
- ご 応用管理	创建新应用	申请开发版	
管理与配置	应用名称	SDKAppID 🛈	应用版本 🛈

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。

应用名称 🔹	请输入应用名称	
数据中心 う *	<b>中国</b> 数据存储在中国,支持全球接入	•
标签	标签键	ă ×
	+ 添加 💿 键值粘贴板	

#### 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。

- 空田管理	创建新应用 申请开发服	Ŕ				请输入 SDKAp	piD 或 应用名称 或 标签	Q
管理与配置	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	
铝 概览								
△ 账号管理	1400	体验版 使用中 ()	中国	2023-08-16	-	-	应用详情 版本对比 查看密钥 标签管理	
品 群组管理	1400	体验版使用中	中国	2023-04-24	-	-	应用详情 版本对比 查看密钥 标签管理	
□□ 功能配置 ~		<b>D / 白 </b> 使用中	中国	2023-03-20	-	-	刻 o 上士 [太手索伊]	
③ 回调配置	1.374X SDKAppi	し旧思					2.	

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。



查看密钥	4. 复制显示后的密钥信息 ×
() 密钥信	息为敏感信息,请注意保密,不要泄露。
密钥	
	显示密钥 ●── 3.点击【显示密钥】

▲ 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

为尊重表情设计版权,下载的 Demo 工程中不包含大表情元素切图,您可以使用自己本地表情包来配置代码。未授权使用 IM Demo 中的表情包可能 会构成设计侵权。

- 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。

* 公有云 SDKAppld,需要替换为您自己账号下的 SDKAppld。		
* · 法入器讯云云通信抢慰局(https://console.cloud.tencent.com/avc)创雄应用,即可看到SDKAppld, * 它是测讯云用于区分客户的唯一标识。	应用资料	
*/ ####################################	SDKAppID 应用名称 应用类型 应用简介	1-milion m 可 Millest -
<sup>a</sup> BAUGDED, A A A BA ADD SOURCE F X static const int EXPIRETIME = 0; /		
+ 计算算系用如加密图码, 联联步增加下: * Mun1. 通从预算正式通信器转送的*tgs://console.cloud.tencent.com/avg), 如果还没有应用数值建一个, * Mun2. 是从原则正常见人基础就是面面, 并是一步找到*张母在美国点"部分。 * Mun2. 点点"重量都好"按照, 就可以指置计算 Lundig 使用的加密的密闭了, 请用其得具并重制到如下的交量中		
> 注意: 该方案仅适用于调试Demo, 正式上线前请将 UserSig 计算代码和密闭过移到您的后台摄务径上,以避免加密密钥泄漏导致的流量溢 < 文档: https://cloud.tencent.com/document/product/269/32688#Server	基础信息	
*/ static NSString * const public_SECRETKEY = 0**;	密明	C 隐藏密钥 密钥信息为敏感信息,请注意保密,不要泄露。
	创建时间 最近修改时间	2021-07-08 2021-07-13

#### ▲ 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。

// 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 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。



## 快速入门(Web & H5 Vue2/Vue3)

最近更新时间: 2024-04-17 10:03:51

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:插件市场 > 开通指引。
- 使用表情回应功能: 请参见 购买指引 购买旗舰版,并且升级 chat-uikit-vue 版本 ≥ 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:







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:



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
使用 vite 方式创建项目,按照下图选项配置 Vue + TypeScript 。
npm create vite@latest
<pre>     Project name: chat-example     Select a framework: &gt; Vue     Select a variant: &gt; TypeScript </pre>
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:



#### 步骤3:引入 TUIKit 组件

.

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):



## 🕗 腾讯云

#### ▲ 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/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:

<b></b>	创建新应用	申请开发版	<u>x</u>		请输入 SDKAppID 或 应用名称 或 标签		Q			
管理与配置	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	
<b>器 概</b> 览										
🔔 账号管理	_	1400	体验版	使用中()	中国	2023-08-16	-	-	应用详情 版本对比 查看密钥 标签管理	
晶 群组管理		1400	体验版	使用中	中国	2023-04-24	-	-	应用详情 版本对比 查看密钥 标签管理	
□□ 功能配置 ~	1.获取	SDKAppIE	)信息	使用中	中国	2023-03-20	-	-	应用详情 版本对比 查看密钥 标签管理	
◎ 回调配置										

● 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。



即时通信 IM	账号管理 140082(	• 3. 切换至目	I标应用账号 <sup>選</sup>	R		产品体验,你说了算
三: 应用管理	新建账号 批量删除	<b>批量导入</b> 批量导出				搜索用户名(UserID Q 🗘
管理与配置	用户名(UserID)	昵称(NickName)	账号类型 ▼	头像(Image)	创建时间	操作
器 概览			答理员		2023-08-16	已中 伯姆 预谐等理员
<b>之</b> 账号管理	4. 点击【新建账号】		百姓以		16:20:45	守山 编码 取消目任贝
<b>點</b> 群组管理	1111		普通账号		2023-08-16 16:21:48	导出 编辑 删除
□□ 功能配置 ~						
◎ 回 2. 点击【账号	号管理】		普通账号		2023-09-19 18:54:48	导出 编辑 删除
● 数据统计 插件服务	2222		普通账号		2023-10-19 11:07:24	导出 编辑 删除

• 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。

即时通信 IM	UserSig生成&校验         16000         当前数据中心:中国①         IM 技术服务交流群 ピ         IM 出)
管理与配置	
<b>器 概览</b>	签名 (UserSig) 生成工具 7. 填写步骤4中创建的账号 <sup>登录鉴权介绍 Ⅳ</sup>
	此工具可以快速生成签名(UserSig),用于中和超速 Demo 以及功能调试。
品 群组管理	用户名 (UserID) 26825
いい 功能配置 🌱 👋	密钥 44
◎ 回调配置	
③ 数据统计	
插件服务	
▶ 推送 ~	
⑤ 内容审核 ~	如有需要,请到应用基础信息中复制密钥
⊙ 智能机器人	
◎ 客服插件	当前生成签名 (11
, 5. 只击【开友】	-#/
⑦ 实时监控	
④ 开发工具 6. 点击	i【UserSig生成&校验】
• 离线推送自查	
・ UserSig生成&校 验	
<ul> <li>自助排障日志</li> </ul>	

## 步骤5: 启动项目

执行以下命令启动项目:

vue-cli	
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:

十 发起单聊 黑 发起群脚	1111
	192287) + xxx (18718888250)283 xx 2458. 77871833058, 8028258, 978925830583, 5282678243, 821598. A8257
1.点击"发起单聊"	2.输入您在"步骤4"中创建的userID, 约下"+"母讲与相类
	3.选择目标用户,点击"完成"发起聊天
	<ul> <li>② E I D 回 回</li> <li>4.在输入框中输入消息并点击"发送"</li> <li>4.</li> </ul>

H5 端 "发送您的第一条消息" 具体步骤示例:

十 发起会话		<	发起单聊		<	1111	
△ 发起单聯●	6 分钟的	1111	Ţ		【安全提示】本 APP 用于业务治该与拓展。	仅用于体验腾讯云即时避 请勿轻信汇款、中奖等过	值 IM 产品功能,不可 反抵款的信息,勿轻易
<ol> <li>久尼有明</li> <li>2222</li> </ol>	11 分钟前	o 🏹 111			班打哈生电话, 编历上	22 <b>0</b> .	nitikw
1.点击"发起单聊"	0.1				- Aria adm		\$\$\$F 🔀
	2.7	制八心住 亚镰4 •	中回建的usenD,按下(	enter 確近1:	「技系		
					4.在转	俞入框中输入	消息并点击"发
		3.选择目标用户,	点击"完成"发起聊天				• <b></b>
					0 5		
		Q 1	取消 奈林		0 🗠		면 (의

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。



Q 你好	〇 十 示例群聊					Q报索	+	示例	洋聊		
全部         文本         文件         其           选择时间:         全部・         今天	(他 近三天 近7天					全部 🥺 未读 12	10:25	iM	MB的手 你好哦,腾讯云即时通讯M的开发者。	搜索会话内容 文本 文件 其他	
文本 観河進梦机 3009条相关文本	15:26	10条与"你好"相关的文本 2010年 你好哇	进入聊天 > 15:26		8=	★ 朝天11:30的部门会 ▲ 朝天11:30的部门会 ▲ 新天11:30的部门会 ▲ 新天11:30時部 ● 日本	议, 你来 10:32 一个用户	iM	■大幅並行消息支送測试路- M助手 等儲倍告诉你几个宝藏链接: の 生物系令口の空間	Q 冠索 选择时间:金翻• 今天 近三天 近7天	
<ul> <li>記水不牛油</li> <li>10条相关文本</li> <li>小熊出击</li> <li>3009条相关文本</li> </ul>	15:28	で 世本不牛油 你好你好	15:28 定位到聊天位置			<ul> <li>         Einda 好的,多街     </li> <li>         Pika 田片已発送,注意査収     </li> </ul>	昨天 <b>〕</b>		<ol> <li>(1) 林道史SHA Lamos</li> <li>(2) 下数中心 (SDK&amp;Demo源码) »</li> <li>(3) 含UH快速集成 »</li> <li>(4) 无UI常规集成 »</li> </ol>	2023年5月 響 浩辰 の好七月.key	
月亮不会告白           3009条相美文本           デ宙航行日记           3009条相美文本	2023/7/21	● 肥水不牛油 你好搞笑	2023/07/26			小林     ● 麻烦告知一下,在該	昨天 等悠回复啊		⑥ 探討特別新活动>	38.2MB 28.2MB 二日 二日 二日 二日 二日 二日 二日 二日 二日 二日	
<b>前任三秒</b> 3009条相关文本	2022/12/12 请输入消息	你好到哪了?		O				〇 [ 道输入	9 - 20 C & - 9	28.2MB	

## 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。

			TUIKit for web 客服插件
	客	服(	Chat 客服 ContactList
Image: Section of the section of t	trans     t		AT     - 201/000
1     1	3 P(0) 223-11-03 223-11-03 223-10-30	0 E	

### 音视频通话

## 

视频通话	语音通话





#### 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。

<mark>云端审核功能</mark> 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 )



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。



6. 运行时报错: /chat-example/src/TUIKit/components/TUIChat /message-input/message-inputeditor.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



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-gro
up.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-cust
omer-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/asnsterchen/ubcuments/sitvia-work/test/test-z.0.//cnat-example-vuez.0/src/lulkit/plugins/plugin-components/message-plugin.vue
6015 error withDefaults is not defined no-under
60:28 error define rops is not defined no-under
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-def
ault.vue
19:15 error 'withDefaults' is not defined no-undef
19:28 error 'defineProps' is not defined no-undef
* 40/ problems (231 errors, 230 warnings) 10 errors and A warnings patentially fizikle with the `fix` ention
To errors and o warnings potentiatly lixable with the
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
<pre>module.exports = defineConfig({      devServer: {       client: {          overlay: false,       },     }, });</pre>
webpack3
<pre>module.exports = {     devServer: {         overlay: false,         },     };</pre>

#### 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:

m	odule.exports = {
r. r	 ules: {
	 '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?



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



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



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



## 快速入门(小程序)

最近更新时间: 2024-04-08 18:08:11

## 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:



		TUIKit for wec	hat 客服插件			
客服	限Chat			客服 Ca	ontactList	
10:20	<b>::!! 🗢 44</b> 5			10:20		:::! 🗢 443
	上商城 ••• 💿			<	客服号	•• 0
【安全提示】本 APP 仅用于体】 用于业务洽谈与拓展,请勿轻信 拨打陌生电话,谨防上当受骗。	验腾讯云即时通信 M 产品功能,不可 汇款、中奖等涉及钱款等信息,勿轻易 <u>点此投诉</u>			线上商城		
未回复,本次服务	务已自动结束,感谢			🧖 线上医疗问	\$	
文迎咨询线上商 <del>1</del>	汤,很高兴为您服务~					
请选择您需要的	的服务:					
告前咨询 # C 昭 名						
查 告后服务 投诉与建议						
超时自动结束提示 未回复,本次服务 您的支持。	示语:由于您长时间 务已自动结束,感谢					
效迎咨询线上商步	汤,很高兴为您服务~					
■ 请选择您需要的	的服务:					
告前咨询						
售后服务						
投诉与建议						
常用语  发送订单	服务评价					
9)	÷ (+)					
						-

#### ▲ 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。



创建小程)	序		
项目名称	miniprogram-11		
目录			
ApplD	•	· 注册 或使用 测试号 ?	
开发模式	小程序		
后端服务	● 微信云开发 无需服务器,即可使用云函数、 力。详情	》 <b>不使用云服务</b> 云数据库、云存储及微信云托管	
模板选择	全部来源	全部分类	
	またののでは、 またのでは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののででは、 本ののでででは、 本ののでででは、 本ののでででです。 本ののででででででででででででででででででででででででででででででででででで	✓ 全部分类 基础 行业 其他 不使用模板 ● arctenate ====================================	扫描试用模板 中分の内容20 中分の中分の日本 中分の中分の日本 中分の日本 中分の中分の日本 中分の中分の日本 中分の日本 中分の中分の日本 中分の日本 中分の日本 中分の日本 中分の中 中分の中分の日本 日本 中分の日本 日本 日本 日本 日本 日本 日本 日本 日本 日本

### 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:





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:

编辑	工具转	到 选择	视图	界面	访
品 田 田 田	编译 [#B] 刷新 [#R] 预览 [公器 真机调试 [	P] [ <b>ፚ</b> ፞፞፞፞ቘR]		小程序	5模:
	工具栏管理清除缓存	2	>		
	自定义分析 微信开发者 多账号调证 真机性能分 自动化测证	f f・代码管理 t t f f f	!		
	构建 npm 插件 拉取周期性 居中显示所	缓存数据			

构建 npm 后目录如下 (新增 miniprogram\_npm 文件夹):



#### MINIPROGRAM-11

▶ 🖿 miniprogram_npm
node_modules
🕨 🌄 pages
🝷 💼 TUIKit
• 👩 .github
🝷 📻 components
🕨 🖿 TUICallKit
🕨 🖿 TUIChat
TUIConversation
► 🖿 TUIGroup
▶ 📑 debug
<b>—</b>

#### ▲ 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。

完成构建。耗时 592 毫秒。
▲
取消 确定

### 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。

#### 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/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。

👻 💼 TUIKit			
	<ul> <li>In .github</li> </ul>		
	۰	components	
	<b>۲</b>	debug	
	<u>۲</u>	static	
	۰ 📭	utils	
	•	.gitignore	
	ū	.npmignore	
	ন্থ	) CHANGELOG.md	
	St.	index.js	
	<b>{</b>	} index.json	
		index.wxml	
	₹	index.wxss	

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。





#### 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
SECRETKE Y	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

<ul> <li>TUI-CustomerService</li> </ul>	•
🝷 💼 pages	•
🗾 index.js	
🔒 index.wxml	м
ᄏ index.wxss	
Page({	
// 其他代码	
onLoad() {	
const TUIKit = this.selectComp	onent('#TUIk

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



JS011 ×1+	
<ul> <li>TUI-CustomerService</li> <li>pages</li> <li>index.js</li> <li>index.yson</li> <li>index.wxml</li> <li>index.wxss</li> </ul>	м М
{ "usingComponents": { "TUIKit": "/TUIKit/index" }, "navigationStyle": "custom" }	
app.js 文件	



<ul> <li>miniprogram</li> <li>vscode</li> <li>components</li> <li>debug</li> <li>mages</li> <li>minages</li> <li>lib</li> <li>pages</li> <li>TUI-CustomerService</li> <li>app.js</li> <li>app.json</li> </ul>
import TencentCloudChat from '@tencentcloud/chat';
import TIMUploadPlugin from 'tim-upload-plugin';
import TimProtanityFilterPlugin from 'tim-protanity-filter-plugin';
App({
onLaunch: function () {
wx.\$TUIKit = TencentCloudChat.create( {
SDKAppID: this.globalData.config.SDKAPPID,
}); const userSig — genTestHserSig(this globalData config) userSig
wx.\$chat_SDKAppID = this.globalData.config.SDKAPPID:
wx.TencentCloudChat = TencentCloudChat;
wx. <mark>\$chat_userlD =</mark> this.globalData.config.userlD;
wx.\$chat_userSig = userSig;
wx.\$TUIKit.registerPlugin({ 'tim-upload-plugin': TIMUploadPlugin });
wx.\$TUIKit.registerPlugin({ 'tim-profanity-filter-plugin': TIMProfanityFilterPlugin });
WX.\$I UIKIT.IOgIn({ 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.\$TOIRIC.ON(WX. IERCENCOULCHILLEVENT.SDK_READT, UNS.ONSDKREady, UNS),
globalData: {
config: {
userlD: '', // User ID
SECRETKEY: ", // Your secretKey
SDKAPPID: 0, // Your SDKAppiD Εχριρετιμέν δολαδο
EXTINETIME. 004000, }.
},
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



<ul> <li>● pages</li> <li>● index.js</li> <li>● index.ison</li> <li>● index.wxml</li> <li>● index.wxss</li> </ul>
js 文件
<ul> <li>▶ mages</li> <li>▶ index.js</li> <li>↓ index.json</li> <li>ⓑ index.wxml</li> <li>♥ index.wxss</li> </ul>
<pre>Page({     handleJump() {         wx.navigateTo({         url: '/.TUI-CustomerService/pages/index',         })     } }</pre>
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
<ul> <li>▼ pages</li> <li>▼ index.js</li> <li>{} index.json</li> <li>Index.wxml</li> <li>Index.wxss</li> </ul>
<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	Numbe r	是	云通信应用的 SDKAppID
SECRETKE Y	String	是	密钥信息,详情可参考 步骤4
EXPIRETIME	Numbe r	否	userSig 过期时间

#### js 文件

🕶 🛅 pages							
- 🖬	index						
[	<mark>s</mark> index.js						
	.} index.json						
	👌 index.wxml						
	ヲ index.wxss						

```
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.$TUIKit.login( {
       userID: this.data.config.userID,
      userSig
     wx.setStorage( {
       data: [],
     wx.$TUIKit.on(wx.TencentCloudChat.EVENT.SDK_READY, this.onSDKReady,this);
```



<pre>}, onUnload() {     wx.\$TUIKit.off(wx.TencentCloudChat.EVENT.SDK_READY, this.onSDKReady,this); }, onSDKReady() {     const TUIKit = this.selectComponent('#TUIKit');     TUIKit.init(); }; </pre>
json 文件
<ul> <li>register pages</li> <li>register index</li> <li>us index.js</li> <li>findex.json</li> <li>index.wxml</li> <li>index.wxms</li> </ul>
{     "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 -- 【应用管理】获取 SDKAPPID、SECRETKEY ,并设置 步骤3 示例代码中的 SDKAPPID、SECRETKEY。

式 应用管理		创建新应用	申请开发版	i					请输入 SDKApp	ID 或 应用名称 或 标签	Q
管理与配置	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 (1)	操作	
出 概览 2、账号管理			1400	体验版	使用中 🛈	中国	2023-08-16	-	-	应用详情 版本对比 查看密钥 标签管理	
品 群组管理			1400	体验版	使用中	中国	2023-04-24	-	-	应用详情 版本对比 查看密钥 标签管理	
□□ 功能配置 ③ 回调配置	ř I	1.获取	SDKAppl	D 信息	使用中	中国	2023-03-20	-	-	应 2. 点击【查看密钥】	
查看密钥	4. 复制	削显示	后的密钥	月信息			×				
() 密钥信息	<b>』为敏感信</b> 』	息,请注意	银密,不要)	世露。							
, 密钥 *	********	*****									
			显示密钥	• 3.	.点击	【显示密钥					
■ 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。											


即时通信 IM	账号管理 1400826	● 6. 切换至日	目标应用账号 群	R		产品体验,你说了算
⋮ 应用管理	新建账号 批量删除	批量导入批量导出				搜索用户名(UserID Q 🗘
管理与配置	用户名 (UserID)	昵称(NickName)	账号类型 ▼	头像(Image)	创建时间	操作
<b>器 概览</b>			節理品		2023-08-16	已山 护城 取消等期日
2 账号管理	7.点击【新建账号】		官理贝		16:20:45	守山 编辑 取消管理贝
晶 群组管理	1111		普通账号		2023-08-16	导出 编辑 删除
號 功能配置 ∨					10.21.40	
<sup>◎ 回调</sup> 5. 点击【账 <sup>4</sup>	弓管理】 <sup>34</sup>		普通账号		2023-09-19 18:54:48	导出 編辑 删除
<ul> <li>数据统计</li> <li>插件服务</li> </ul>	2222		普通账号		2023-10-19 11:07:24	导出 编辑 删除

# 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。避免开发者工具的缓存造成渲染异常。



普通编译	C	0	<u>ب</u>	<b>⊗</b> ∙		
	编译	预览	真机调试	清除模拟器缓存	>	
				清除编译缓存 		

## 3. 单击编译图标。



# 步骤6:发送您的第一条消息

1925 at ♥ ■ ○ 82553 ● 0	1638 输入你在步骤4中创 建的userID,点击				16:31 く 【学生現示】本 APP 校 用于自然空谈与后来,1 展行团主先生。国际上生	<b>152758</b> 用于体道接讯云即打通话 物段做汇款,中实等涉及 过资源。	
Запоров         лла           Полоров         лла	完成进行搜索		<ul> <li>Revolution</li> </ul>	选择目标用户并点 击"确认邀请"发起会 话 在输入框输入消	(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)		🕒 xiz
2002 USA	点击发起会话 ## ## ## ## ## ## ## ## ## ## ## ## ##	1 2 3 G 4 5 6 25 7 8 9 14 1 0 25	<b>8.4</b>	息,并点击"发送"	找你哈哈 123?! epv OHI ABC PORS 通道時間	好哈哈哈 ABC DI JKL MI S TUY W3 F 空格	你们 i 丫 EF ② NO 쓴 YYZ XX

## ▲ 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。



			TUIKit for wea	chat 客服插件			
	客服Chat				客服(	ContactList	
10:3	20 ::!!	<b>≈</b> 44+			10:20		::!! 🗢 445
	线上商城	•• •			<	客服号	••• 0
【安全提示 用于业务洽 裁打陌生电	3)本 APP 仅用于体验期讯云即时通信 M 产品 该与拓展。请勿轻信汇款、中奖等涉及钱款等付 活,谨防上当受骗。	品功能,不可 宫息,勿轻易 点此投诉			线上商城		
	未回复,本次服务已自动结束,感谢 您的支持。				👳 线上医疗问	可诊	
Ť	欢迎咨询线上商场,很高兴为您服务~				1.07		
	请选择您需要的服务:	_					
	售前咨询	_					
	售后服务	_					
Ť	投诉与 建议 超时自动结束提示语:由于您长时间 未回复,本次服务已自动结束,感谢 您的支持。						
*	欢迎咨询线上商场,很高兴为您服务~						
_	请选择您需要的服务:						
-	售前咨询 售后服务						
	投诉与建议						
()) ()	语 发送订单 服务评价 •	• +					
					-		_

## 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。



# 快速入门(uniapp vue2/vue3)

最近更新时间: 2024-04-17 10:03:51

# chat-uikit-uniapp 介绍

chat-uikit-uniapp(vue2 / vue3)是基于腾讯云 Chat SDK 的一款 uniapp UI 组件库,它提供了一些通用的 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 后,集成 TencentCloud-TIMPush,可支持消息推送,效果如下:

941	普通消息推送				
		9:41	Patton		
09:34 6月28日 星期三		A	稍后会议联系♥	占土则结议	自毋而
補后会议联系 Nicole 図の分析前 国片消費を自め设计稿、対比者下研				<b>黒山郎</b> 村7月	あって回
James [刊版编述话]					
收到离线推送消息					
		٩		⊕ ©	





#### 消息云端搜索功能界面效果如下图所示:

● 席席       ● 作用       ● 作       ●        ● 作       ●        ● 作       ●        ●        ●        ●        ●        ●        ●        ●        ●        ● ●	9:41 消息	= \$ III. ⊕	9:41 🕈 🖛	9:41 。 1 探索	I <b>?</b> ■
With FLB         Comparing Lage         Mit FLB         Comparing Lage         Comparing Lage <thcomparing lage<="" th="">         Comparing Lage</thcomparing>	Q 搜索		全部 聊天记录 联系人 群聊	全部 文本 文件 图片/视频 其他	
Instantiality       Cp.Bt/S7       Documentality         Image: Stant Add Product Add Prodot Add Product Add Product Add Product Add Pr	6 陈志杰	20分钟前	选择时间:全部• 今天 近三天 近7天	选择时间:全部 • 今天 近三天 近7天	
	请问您的问题是?	点击"搜索"	全局搜索	2023年5月	- 1
MABB       MRL::::::::::::::::::::::::::::::::::::	▲ 这个是什么?	进入全局搜索	3009条相关聊天记录	<b>經</b> 你脸红什么啊	5/9
学育育       202/00/01         (正規))       202/00/01         (二)       202/00/01         (二)       202/00/01         (二)       10006/用見規定法法         (二)       10006/用見規定法         (二)       10006/用見規定法         (1)       10006/用見規定法         (1)       10006/用見         (1)       10006/用見         (1)       10006/用目         (1)       10006/用目         (1)       10006/用目         (1)       10006/用目         (1)       10006/用目         (1)	<b>林敬海</b> [132条]好的	星順二 12:02 段	<b>肥水不牛油</b> 3条相关聊天记录	の好七月.key 38.2MB	
78.4x         0.021/00/01         2020/00/01         2020/00/01         12.55A/54           <	李育菁 [草稿]哈哈哈哈哈哈	2021/09/01	小熊出击	🧱 你脸红什么啊	5/4
取消違梦机 (正音)         2021/00/01         「「」」のの参相長期天法法         「」」のの参相長期天法法         「」」のの参相長期天法法         「」」の参相長期天法法         「」」の参相長期天法法         「」」の参相長期天法法         「」」の参相長期天法法         「」」の参相長期天法法         「」」の参加         「」」の参加         「」」の         「」」」の         「」」の         「」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」の         「」」」」の         「」」」」の         「」」」」の <td><b>罗德杰</b> ④ 这个是什么?</td> <td>2021/09/01</td> <td>3009条相关聊天记录</td> <td><b>十万个为什么.key</b> 38.2MB</td> <td>- 1</td>	<b>罗德杰</b> ④ 这个是什么?	2021/09/01	3009条相关聊天记录	<b>十万个为什么.key</b> 38.2MB	- 1
(面)         (面)         (面)         (一)         (1) <td>银河造梦机</td> <td>2021/09/01</td> <td>3009条相关聊天记录</td> <td>(你脸红什么啊</td> <td>5/1</td>	银河造梦机	2021/09/01	3009条相关聊天记录	(你脸红什么啊	5/1
成プルマト油         2010001         第 3008年損失期天记法         2023年3月           ① 1/2 (F1 new 5.txt         2021/00/01         前 任三秒 3009条相关期天记法         2023年3月         3/9           ② 1/2 (F1 new 5.txt         2021/00/01         2020/01/01         2023年3月         3/9           ② 有務常約夏日 1/2 (大作) new 5.txt         2021/00/01         2020/01/01         2023年3月         3/9           ② 有務常約夏日 1/2 (大作) new 5.txt         2021/00/01         2020/01/01         2023年3月         3/9	[语音]	2021/00/01	宇宙航行日记	P 学会沟通.key 38.2MB	
小瓶出击         202/(0/01)         前任三秒         1000000000000000000000000000000000000	(二) 肥水小牛油   文件] new 5.txt	2021/08/01	3009条相关聊天记录	2023年3日	
有薄雾約夏日     202/(0//01     202/(0//01     百事可夏     日常可夏     日常可夏     300の条相关期天记录       [文件] new 5.txt     第二分相节     第二分相节     第二分相节     第二分相节     第二分相节     第二分相节	<b>小熊出击</b> [文件] new 5.txt	2021/09/01	<b>前任三秒</b> 3009条相关聊天记录	《 你脸红什么啊	3/9
- 一〇田戸 - 〇田戸 - 〇日 - 〇	<b>有薄雾的夏日</b> [文件] new 5.txt	2021/09/01	百事可爱 3009条相关聊天记录	金字塔原理.key 38.2MB	
			来一份甜豆	(你脸红什么啊	3/9

在线客服功能界面效果如下图所示:



TUIKit fo	pr uniapp 客服插件
rang object	
各版Chat	各版 ContactList
く 线上商城	+ 添加好友/群聊
【安全提示】本 APP 仅用于体验算讯云即时通信 M 产品功能,不可 用于业务治该与托展。请勿轻信汇款、中零等涉及钱款的信息。勿轻易	> 新的联系人
<b>拔打陌生电话,谨防上当受骗。</b>	>黒名単
	> 我的詳聊
"管" 选择需要退货的商品:	> 我的好友
	◇客服号
手机 >	22 At Lands
电脑	4X M0 4X
	👷 线上医疗问诊
本次服务已结束,感谢您的支持。	
(m)的商品已经安排调整。	
Point Studient Physics 6 14, 56756 5	
请选择您需要的服务:	
售前咨询 <b>&gt;</b>	
告后服务 >	
投诉与建议 >	
(mist) 2 (mist)	
	HAR REAL LATIN

# 支持平台

- Android
- iOS
- 微信小程序
- H5

# 开发环境要求

- HBuilderX (HBuilderX 版本 >= 3.8.4.20230531) 或者升级到新版本
- Vue2 / Vue3
- sass (sass-loader 版本 ≤ 10.1.1)
- 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

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。



普通项目	新建uni-app项目 开发一次,同时发布为Android、i	OS原生App、小程序、H5等多个	平台。[了解更多]		;	×
uni-app						
Wap2App	chat-example	ts/HRuilderProjects			3311 Vár	
5+App	选择模板	(s) induitive in ojects			MEN	
IDE插件		Image: state		Q 查找模板	去搜索	
	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	neno uni-app 同时可	支持 vue2 / vue3	Hello uts	un-statter	
<ul> <li>从SVN导入项目</li> <li>4.(in=) 项目</li> </ul>	启用uniCloud(全端可用的	云开发,js快速完成后台业务) <u>详</u>	膏 ☞ 阿里云 ○ 腾讯云			
♥ 从UIT等入坝目	上传到GitCode代码托管平台	(CSDN) <u>详情</u>			如何用cli创建项目    创建	

## 步骤2: 下载 TUIKit 组件

• HBuilderX 不会默认创建 package.json 文件,因此您需要先创建 package.json 文件。请在项目根目录下执行以下命令:

	npm init -y
•	下载 TUIKit 并拷贝至源码中:
r	nacOS 端

通过 npm 方式下载 TUIKit 组件:

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

为了方便您后续的拓展,建议您将 TUIKit 组件复制到自己工程的 pages 目录下,请在自己的项目根目录下执行以下命令:

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

从 v2.0.0 版本起,uikit 默认支持 客服插件,客服插件需要进行源码集成,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

为了方便您后续的拓展,建议您将 TUIKit 组件复制到自己工程的 pages 目录下,请在自己的项目根目录下执行以下命令:



xcopy .\node\_modules\@tencentcloud\chat-uikit-uniapp .\TUIKit /i /e /exclude:.\node\_modules\@tencentcloud\chat-uikituniapp\excluded-list.txt

从 v2.0.0 版本起,uikit 默认支持 客服插件,客服插件需要进行源码集成,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

• 在根目录下创建 vue.config.js( vue3 项目请忽略此步骤 )。



• 在 manifest.json 文件的源码视图中开启分包配置。



⚠ 注意:

打包小程序时 manifest.json 不要配置 lazyCodeLoading 选项。

## 2. 集成 TUIKIt

```
⚠ 注意:
进行集成时,请严格按照以下四个步骤进行集成。如果您希望打包小程序,请不要跳过"小程序分包首页"的配置。
```



#### main.js 文件

请注意,Vue2环境下要使用 Vue.use(VueCompositionAPI) ,防止环境变量 isPC 等无法使用。

// 引入主包依赖 import <b>TencentCloudChat</b> from "@tencentcloud/chat"; import <b>TUICore</b> from "@tencentcloud/tui-core";	
import <b>App</b> from './App';	
<pre>// #ifndef VUE3 import Vue from 'vue'; import './uni.promisify.adaptor'; import VueCompositionAPI from "@vue/composition-api"; Vue.use(VueCompositionAPI); Vue.config.productionTip = false; App.mpType = 'app'; const app = new Vue({    App,     });     app.\$mount(); // #endif</pre>	
<pre>// #ifdef VUE3 import { createSSRApp } from 'vue'; export function createApp() {    const app = createSSRApp(App);    return {      app,    };    }    // #endif</pre>	

#### pages.json 文件

```
{
    "pages": [{
    "pages": [{
    "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"
        }
        },
        // 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"
        }
    },
    // 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"
        // 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"
        // 集成 chat 细件,
        // 集成 chat 细件,
        // 集成 chat 细件,
        // 集成 chat 细件,
        /// 集成 chat 细件,
        // 集成 chat 细 chat muth(
        // 生)
        /
```



#### , },

```
"navigationBarTitleText": "腾讯云 IM"
"navigationBarTitleText": "腾讯云 IM"
"navigationBarTitleText": "腾讯云 IM"
"navigationBarTitleText": "聊天记录"
}
```

# App.vue 文件

```
<script lang="ts">
// #ifdef APP-PLUS || H5
import { TUIChatKit, genTestUserSig } from "./TUIKit";
import { vueVersion } from "./TUIKit/adapter-vue";
import { TUILogin } from "@tencentcloud/tui-core";
// #endif
// 必填信息
const config = {
    userID: "test-user1", // User ID
    SDKAppID: 0, // Your SDKAppID
    secretKey: "", // Your secretKey
};
uni.$chat_userID = config.userID;
uni.$chat_SDKAppID = config.SDKAppID;
uni.$chat_secretKey = config.secretKey;
```



```
uni.$chat_userSig = genTestUserSig(config).userSig;
TUIChatKit.init();
 onLaunch: function () {
 TUILogin.login({
   SDKAppID: uni.$chat_SDKAppID,
   userID: uni.$chat_userID,
   // UserSig 是用户登录即时通信 IM 的密码,其本质是对 UserID 等信息加密后得到的密文。
   userSig: uni.$chat_userSig,
   useUploadPlugin: true,
   // 本地审核可识别、处理不安全、不适宜的内容,为您的产品体验和业务安全保驾护航
   // 此功能为增值服务,请参考: https://cloud.tencent.com/document/product/269/79139
   useProfanityFilterPlugin: false,
   framework: `vue${vueVersion}` // 当前开发使用框架 vue2 / vue3
onShow: function() {
   console.log('App Show')
onHide: function() {
   console.log('App Hide')
</script>
<style>
uni-page-body,
html,
body
page {
 width: 100% !important;
 height: 100% !important;
 overflow: hidden;
}
</style>
```

#### 小程序分包首页

## ▲ 注意:

- 小程序默认分包集成,需要在 TUIKit 首启动页面完成 login。
- 如果您不需要打包小程序(例如仅构建 H5),可忽略小程序分包首页的配置内容。

#### 示例: TUIKit 分包首屏启动页面为 TUIConversation 页面。

1. 在 TUIKit/components/TUIConversation 文件夹下创建 subPackage-init.ts 文件。

import { TUIChatKit, genTestUserSig } from "../../index.ts"; import { vueVersion, onMounted } from "../../adapter-vue"; import { TUILogin } from "@tencentcloud/tui-core";

// TUIChatKit 初始化





2. 在 TUIKit/components/TUIConversation/index.vue 中导入。



# 3. 在项目主包首页中配置 TUIConversation 和 TUIContact 的入口

在 pages/index 文件夹下创建 index.vue 文件。





```
methods: {
  </script>
  <style lang="scss" scoped>
  .index {
   height: 100%;
   display: flex;
   flex-direction: column;
   align-items: center;
   &-button {
    width: 180px;
    padding: 10px 40px;
    color: #fff;
    background-color: #006eff;
    font-size: 16px;
    margin-top: 65px;
    border-radius: 30px;
    text-align: center;
  </style>
步骤4: 获取 SDKAppID、secretKey、userID
```

配置根目录下 App.vue 文件中 config 对象的 SDKAppID、secretKey 以及 userID。其中 SDKAppID、secretKey 可通过 即时通信 IM 控制台 获取, userID 可在 即时通信 IM 控制台 中创建账户时获取。



#### 获取 SDKAppID、secretKey

在即时通信 IM 控制台中的应用管理页面下,可以看到您创建的应用,第二列即是 SDKAppID,然后单击操作中的查看密钥。

式 应用管理	创建新应用	申请开发版						请输入 SDKApp	ID 或 应用名称 或 标签	Q
管理与配置	应用名称	SDKAppID (j)	应用版本	服务状态	数据中心 ▼	创建时间	到期时间 🛈	标签 🛈	操作	
A 账号管理		1400	体验版	使用中 🛈	中国	2023-08-16	-	-	应用详情 版本对比 查看密钥 标签管理	
<b>晶</b> 群组管理		1400	体验版	使用中	中国	2023-04-24	-	-	应用详情 版本对比 查看密钥 标签管理	
□□ 功能配置 ~	1.获取	SDKAppl	D信息	使用中	中国	2023-03-20	-	-	应 2 点击【杳看密钥】	
◎ 回调配置		OBIG								

网站会弹出查看密钥的对话框,然后单击显示密钥,即可查看密钥。





#### 创建 userID 为 test-user1 的账户

单击控制台左侧的**账号管理**,如果您有多个应用,请注意切换至当前应用,然后在当前应用下单击新建账号,创建一个 userID 为 test-user1 的账号。

(	说明:				
	创建账户的步骤可以跳过,	因为 TUIKit 进行登录时,	若配置的 userID 不存在,	会自动创建账户,	此处仅展示如何获取 userID。

即时通信 IM	账号管理 1400820	● 6. 切换至	目标应用账号	群区		产品体验,你说了算
弐 应用管理	新建账号 批量删除	批量导入批量导出				搜索用户名(UserID Q 🗘
管理与配置	用户名(UserID)	昵称(NickName)	账号类型 ▼	头像(Image)	创建时间	操作
<b>器 概</b> 览			餘田日		2023-08-16	
2 账号管理	7.点击【新建账号】		官埋贝		16:20:45	导出 骊铒 取消管理页
品 群组管理	1111		普通账号		2023-08-16	导出 编辑 删除
□□ 功能配置 >					10.21.40	
<sup>◎ 回调</sup> 5. 点击【账 <sup>4</sup>	号管理】34		普通账号		2023-09-19 18:54:48	导出 编辑 删除
<ul> <li>数据统计</li> <li>插件服务</li> </ul>	2222		普通账号		2023-10-19 11:07:24	导出 编辑 删除

#### 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。

É	HB	uilderX	文件	编辑	选择	查找	跳转	运行	发行	视图	工具	具 帮助 段 474	字 6
••	•							运行到	浏览器		>	chat-exa	mpl
孠				ជ			chat-e	运行到	内置浏	览器			
								运行到	手机或机	莫拟器			
~ <b>D</b>	chat	-exam	ole					运行到	小程序	莫拟器	>	微信开发者工具-[chat-example]	
		hbuilde	rx					运行到	终端		>	微信开发者工具 - [chat-example] - 指定页面	>
			a alcolara							2	2 🗆 /	百度开发者工具 - [chat-example]	
>	_	10de_m	odules								1	百度开发者工具 - [chat-example] - 指定页面	>
>		bages										支付宝小程序开发者工具 - [chat-example]	
>	<b>•</b>	static										支付宝小程序开发者工具 - [chat-example] - 指定页面	>
>		ΓUIKit										抖音开发者工具 - [chat-example]	
>		unpacka	ige								, (	QQ开发者工具 - [chat-example]	
	1	App.vue									$\downarrow$	360开发者工具 - [chat-example]	

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. 通过 即时通信 IM 控制台 创建一个 User 账号。

从左侧边栏进入账号管理页面,单击新建账号并创建一个普通账号 userID: test-user2。



即时通信 IM	账号管理	▼ 当前数据中心:中国	④ IM 技术服务交流	記群 ピ		产品体验,你说了算
≕ 应用管理	新建账号 批量删除	批量导入批量导出				搜索用户名(UserID Q 🗘
管理与配置	用户名(UserID)	昵称(NickName)	账号类型 ▼	头像(Image)	创建时间	操作
器 概览			等理品		2023-08-16	昆山 伯姆 预谐等理员
<b>之</b> 账号管理	<b>点</b> 击【新 <b>建</b> 账号】		自柱以		16:20:45	守山 潮林 秋府昌庄贝
品 群组管理	test-user2	test-user2	普通账号		2023-08-16 16:21:48	导出 编辑 删除
□□ 功能配置 ~						
⇒ 回调配置	1234		普通账号		2023-09-19 18:54:48	导出 编辑 删除
② 数据统计					2023-10-19	
插件服务	2222		普通账号		11:07:24	导出 编辑 删除

#### 2. 运行项目并发起会话。

单击打开 TUIKit 会话,搜索用户 userID: test-user2,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

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)**。

É	HBuile	derX	文件	编辑	选择	查找	跳转	运行	发行	视图	工具	帮助	
) () 7]	•						size-te	st–vue	原生	App-云打 App-查看	包云打包料	犬态	
>	🖿 nod	e_moo	dules						原:  	App-本地 pp-制作	打包 应用wg	t包	>
>	🖿 pag	es							小程	序-微信 (亻	又适用于	uni-app)	
>	🖿 stat	ic							小程	序-百度 (f	又适用于	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。

微信小程序发行	<u>cli程序化部署教程</u>	×
size-test		
输入微信小程序AppId		
生成sourcemap(可用于uni统计的错误分析) 详情		
发行为混合分包 详情		
自动上传到微信平台(不会打开微信开发者工具) 详情		
欢迎开通 <u>uniAD</u> 微信小程序版广告, <u>申请指南</u> 丨 <u>开发文档</u> 高级	发行	

# 更多高级特性(强烈推荐)

# 消息推送

🕛 说明:

TUIKit 中默认没有集成 TencentCloud-TIMPush 推送插件。TencentCloud-TIMPush 是腾讯云即时通信 IM Push 插件。目前推送支持小米、 华为、荣耀、OPPO、vivo、魅族、APNs、一加、realme、iQOO 和 苹果等厂商通道。 如果您需要在 APP 中集成离线推送能力,请参见 uni-app 推送 实现。



	普通消息推送		
9:41 09:34 6月28日 星期三		9:41 < ③ Patton	= \$ la. 
Patton         5 分钟的           稍后会议联系         0           Nicole         20 分钟前           IM         Nicole           IM         Nicole           IM         Nicole           IM         Nicole           IM         利田雪田           IM         James           IM         Immediate	•		点击跳转消息界面
收到离线推送消息			
		٩	ⓒ ⊕





# 消息云端搜索

# ▲ 注意:

@tencentcloud/chat-uikit-uniapp ≥ 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 ≥ 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。

TUIKit for uniapp 客服插件	
客服Chat	客服 ContactList
〈 线上商城	+ 添加好友/群聯
【安全课示】本 APP (风开于体结器讯云即时通信 M 产品功能,不可 用于业务运动与规制、请导经保证数,中奖等涉及钱数的信息,为经易	> 新的联系人
	〉黑名单
这择需要退货的商品:	> 我的群期
	> 找的好友
<u>軽 /</u> 手机 >	
	2. 23_16940
本次服务已结束,感谢您的支持。	💮 线上医疗问诊
er 🛪 🙎	
您的商品已经安排退货。	
请选择您需要的服务:	
告前咨询 入	
雪石服务 > 投诉与建议 >	
文理咨询线上商场,很高兴为您服务-	
请输入消息 发送	0 b 1
	消息 遺筑表 个人中心

# 音视频通话 TUICallKit 插件

() 访	<b>徒明:</b>			
т	UIKit 中默认没有集成 TUICallKi	<b>it 音视频组件,</b> TUICallKit 主要负责语	<b>音、视频通话</b> 。	
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: 音视频通道	舌 ( 小程序 )		
	• 打包到 H5 请参见: 音视频通话	(H5)		

• 客户端通话示意图:





## 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

- 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,请参见 TUIChat 独立集成方案。
- 2. 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 。

3. 更多问题请参见 uni-app 相关问题。

# 参考文档

UIKit Github 源码:

• chat-uikit-uniapp (vue2/vue3) github 源码

实现更多功能,请参考 ChatEngine API 文档:

- ChatEngine API 手册
- ChatEngine npm



# 快速入门(React)

最近更新时间: 2024-04-22 11:03:21

# 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 ≥ v18.0
- TypeScript
- 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)

# 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

- 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:

即时通信 IM	← 基本配置 ▼ IM 技术服务交流群	
手 基本配置		
<b>日</b> 功能配置 ~	应用套餐包	
晶 群组管理	状态 使用中	
☆ 回调配置		
凹 插件市场 BETA	升级套餐包 更多操作 ▼	
♂ 监控仪表盘 ◆		
<ul> <li>   ④ 辅助工具 ~  </li> </ul>	应用资料	扁辑
◎ 在线客服	SDKAppID	
	应用名称 <b>tim-demo</b>	
	应用类型 游戏	
	应用简介 -	
	基础信息	
	密钥	
	隐藏密钥 密钥信息为敏感信息,请注意保密,不要泄露。	
	创建时间 2022-06-24	
	最近修改时间 2022-06-24	

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。

即时通信 IM		▼ IM 技术服务交流群			产品体验。你说了算
·	新建成号 北岸形脉 批量导入	批量导出			搜索用户名(UserID) Q 🗘
III 功能配置 ·	用户名 (UserID)	kName)	半像 (Image)	éri sur pet na	操作
<b>名</b> 账号管理		maxim,	, (m. (	00702-07-0	2011
品 群组管理	1111	普通帐号		2022-09-01 10:56:48	导出 编辑 删除
☆ 回调配置				2022-09-01	
凹 插件市场 BETA	administrator	管理员		10:53:16	导出 編輯 删除
⑦ 监控仪表盘 →	共 2 条			10 ▼ 条/页 H	4 1 /1页 ▶ H

## 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,点击"+"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. 点击用户头像发起会话。
- 4. 在输入框输入消息,按下"enter"键发送。

😡 test	
Q Search (+) I.create an one-t	Welcome to in We provide a demo for you to try out the features. brone chat Please note that by using this demo, we will process your personal data in accordance with our Tencent Cloud Privacy Policy.
No conversation	

### 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



# 快速入门(React Native)

最近更新时间: 2024-04-09 15:26:11

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

# 环境要求

	版本
React Native	0.63.4 版本以上
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。

## 前提条件

您已 注册腾讯云 账号,并完成 实名认证。

# 第一部分: 创建测试用户

在 IM 控制台 选择您的应用,在左侧导航栏依次点击 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 。

於 購訊云 总統	
即时通信 IM	← UserSig生成&校验 ▼
⋮ 基本配置	
11 功能配置	签名(UserSig)生成工具 登录鉴权介绍 🗹
晶 群组管理	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。
② 回调配置	请输入用户名 (UserID)
🕣 安全打击	user1
数据监控器	密钥 (Key)
@ 辅助工具	
• 离线推送自查	
・ UserSig生成&校 验	生成签名 (UserSig)
-	当前生成的签名(UserSig)是:
	复制签名 (UserSig)

# 第二部分:集成 React Native SDK

前提条件



您已经完成创建 React Native 项目,或有可以基于的 React Native 项目。

# 接入步骤

## 安装 IM SDK

使用如下命令,安装 React Native IM SDK 最新版本。在命令行执行:

// npm npm install react-native-tim-js // varn

yarn add react-native-tim-js

## 完成 SDK 初始化

调用 initSDK ,完成 SDK 初始化。将您的 sdkAppID 传入。

```
import { TencentImSDKPlugin, LogLevelEnum } from 'react-native-tim-js';
TencentImSDKPlugin.v2TIMManager.initSDK(
    sdkAppID: 0, // Replace 0 with the SDKAppID of your IM application when integrating
    loglevel: LogLevelEnum.V2TIM_LOG_DEBUG, // Log
    listener: V2TimSDKListener(),
);
```

在本步骤,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

- 1. 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。
- 2. 调用 TencentImSDKPlugin.v2TIMManager.login 方法,登录一个测试账户。当返回值 res.code 为0时,登录成功。

```
import { TencentImSDKPlugin } from 'react-native-tim-js';
const userID = "";
const userSig = "";
const res = await TencentImSDKPlugin.v2TIMManager.login(
userID,
userSig,
);
```

() 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

此处以发送文本消息举例,其流程为:

- 1. 调用 createTextMessage(String) 创建一个文本消息。
- 2. 根据其返回值,拿到消息 ID。

3. 调用 sendMessage() 发送该 ID 的消息。 receiver 可填入您此前创建的另一个测试账户 ID。发送单聊消息无需填入 groupID 。

代码示例:

import { TencentImSDKPlugin } from 'react-native-tim-js';

```
const createMessage =
   await TencentImSDKPlugin.v2TIMManager
   .getMessageManager()
   .createTextMessage("The text to create");
```



#### const id = createMessage.data!.id!; // The message creation ID

```
const res = await TencentImSDKPlugin.v2TIMManager
.getMessageManager()
.sendMessage(
    id: id, // Pass in the message creation ID to
    receiver: "The userID of the destination user",
    groupID: "The groupID of the destination group",
    );
```

#### 🕛 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

```
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



#### 为了获取会话列表,需要维护 nextSeq ,记录当前位置。

```
import { useState } from "react";
import { TencentImSDKPlugin } from "react-native-tim-js";
const [nextSeq, setNextSeq] = useState<string>("0");
const getConversationList = async () => {
  const count = 10;
  const res = await TencentImSDKPlugin.v2TIMManager
  .getConversationManager()
  .getConversationList(count, nextSeq);
  setNextSeq(res.data?.nextSeq ?? "0");
};
```

此时,您可以看到您在上一步中,使用另一个测试账号,发来消息的会话。

#### 监听长链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。
```

```
import { TencentImSDKPlugin } from "react-native-tim-js";
const addConversationListener = () => {
TencentImSDKPlugin.v2TIMManager
.getConversationManager()
.addConversationListener({
onNewConversationListener({
onNewConversation: (conversationList) => {
// new conversation created callback
_onConversationListChanged(conversationList);
},
onConversationChanged: (conversationList) => {
// conversationChanged callback
_onConversationListChanged(conversationList);
},
};
};
```

2. 处理回调事件,将最新的会话列表展示在界面上。

```
const _onConversationListChanged = (list) => {
    // you can use conversation list to update UI
};
```

## 接收消息

通过腾讯云 IM React Native 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. 监听长链接,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:
```

import { TencentImSDKPlugin } from "react-native-tim-js";



```
const getGroupHistoryMessageList = async () => {
  const groupID = "";
  const count = 20;
  const lastMsgID = "";
  const res = await TencentImSDKPlugin.v2TIMManager
  .getMessageManager()
  .getGroupHistoryMessageList(groupID, count, lastMsgID);
  const msgList = res.data ?? [];
  // here you can use msgList to render your message list
};
```

#### 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 V2TimAdvancedMsgListener.onRecvNewMessage 。 onRecvNewMessage 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:

```
import { TencentImSDKPlugin } from "react-native-tim-js";
const adVancesMsgListener = {
    onRecvNewMessage: (newMsg) => {
    _onReceiveNewMsg(newMsg);
    /// ... other listeners related to message
    },
};
const addAdvancedMsgListener = () => {
    TencentImSDKPlugin.v2TIMManager
    .getMessageManager()
    .addAdvancedMsgListener(adVancesMsgListener);
};
```

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,离线推送,本地搜索等相关功能开发,详情可参见 SDK API 文档 。

#### 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)。

## 常见问题

运行 demo 时出现 Undefined symbols for architecture x86\_64 [duplicate] 如何解决?

请参见文档。

运行 demo 时出现 Failed to resolve: react-native-0.71.0-rc.0-debug 如何解决?

请参见 <mark>文档</mark> 。

## 联系我们

如果您在接入使用过程中有任何疑问,请扫码加入微信群,或加入QQ群: 437955475 咨询。



有问题,群里问!

扫码加入 腾讯云IM React Native 微信/QQ 交流群







**QQ群** 群号: 437955475



# 快速入门 (Electron)

最近更新时间:2024-03-14 10:54:32

本文主要介绍如何快速运行腾讯云即时通信 IM Demo(Electron)并了解集成 Electron SDK 的方法。

# 环境要求

平台	版本
Electron	13.1.5到17.0.0版本
Node.js	v14.2.0

# 支持平台

目前支持 macOS 和 Windows 两个平台。

# 体验 DEMO

在开始接入前,您可以体验我们的 DEMO ,快速了解腾讯云 IM Electron SDK。

## 前提条件

您已 注册腾讯云 账号,并完成 实名认证。

### 操作步骤

# 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 信息, 直接跳到 步骤2: 获取密钥信息 。
- 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**,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。

即时通信 IM		应用管理	IM 技术服务交流群 🖸	IM 出海交	流专区
式 应用管理		创建新应用	申请开发版		
管理与配置		应用名称	SDKAppID	i	应用版本 🛈
创建新应用					×
应用名称 *	请输入应用名称	尔			
数据中心 🛈 *	中国 数据存储	在中国,支持全球	接入		•
标签 🛈	标签键		标签值		×
	+ 添加   键	值粘贴板			
		确定	I		
3. 创建完成后,可在原	<b>立用管理</b> 页面查	<b>:</b> 看、搜索、管理	[应用。 <b>请记录 SD</b> K	AppID 信息	3.



创建新应用	申请开发版							请输入 SDKAppID 或 应用名称 或 标签	
应用名称	SDKAppID 🛈	应用版本 (1)	服务状态	数据中心 ④ ▼	创建时间	到期时间 🛈	标签 🤅	D 操作	
—	1600	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	
	1600	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	
	1600	旗舰版	使用中 🛈	中国	2023-12-08	2024-01-08	-	应用详情 版本对比 查看密钥 标签管理	

# 步骤2: 获取密钥信息

1. 在**应用管理**页面的操作中单击**查看密钥**。

应用名称	SDKAppID	应用版本 🛈	服务状态	数据中心 ③ ▼	创建时间	到期时间 ()	标签 (1)	操作
	1600	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
测试	1600	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
测试	1600	旗舰版	使用中 🛈	中国	2023-12-08	2024-01-08	-	应用详情 版本对比 查看密钥 标签管理

#### 2. 在弹出的对话框中,单击显示密钥,复制并保存密钥信息。

查看密钥	×
⑦ 密钥信息为敏感信息,请注意保密,不要泄露。	
密钥 ************************************	
⑦ 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。	

3. 在左侧导航栏"开发辅助"分类下,单击**开发工具 > UserSig 生成&校验**,创建一个 UserID 及其对应的 UserSig,复制签名信息,供后续在 Demo 中登录 时使用。



即时通信 IM	UserSig生成&校验	- 测试								
三: 应用管理										
管理与配置										
<b>田</b> 概览	此上具可以快速生成金;	sulleryをTwikiとLiwEin (Usensing), m ) やくBood Demo なな少形で見た。								
<u> </u>	用户名 (UserID)	请输入								
品 群组管理	密钥	104a2ef2								
□□□□ 功能配置 · · · · · · · · · · · · · · · · · · ·										
◎ 回调配置										
① 数据统计										
插件服务		約者豐憲 语列应用其政体自由每制際組								
□ 推送 ∨										
🕣 安全打击		生成签名(UserSig)								
⑤ 内容审核 🛛 🗸	当前生成签名 (UserSig)									
⊙ 智能机器人										
◎ 客服插件										
凹 插件市场										
开发辅助										
③ 实时监控		复制签名 (UserSig)								
③ 开发工具 ^										
• 离线推送自查										
・ UserSig生成&校 验										
<ul> <li>自助排障日志</li> </ul>										

# 步骤3:选择适合的方法集成 Electron SDK

IM 提供了两种方式来集成,您可以选择最合适的方案来集成:

集成方式	适用场景
使用 DEMO	IM Demo 包含完整的聊天功能,代码已开源,如果您需要实现聊天类似场景,可以使用 Demo 进行二次开发。可立即体验 Demo 。
自实现	如果 Demo 不能满足您应用的功能界面需求,可以使用该方法。

为帮助您更好的理解 IM SDK 的各 API,我们还提供了 API 文档。

## 步骤4:使用 Demo

1. 克隆即时通信 IM Electron Demo 源码到本地。

git clone https://github.com/TencentCloud/tc-chat-demo-electron.git

2. 安装项目依赖。

// 项目根目录 npm install			
// 渲染进程目录 cd src/client npm install			

3. 项目运行。

// 项目根目录 npm start

4. 项目打包。



#### // mac打起 npm run build:mac

// windows**打包** npm run build:windows

#### 🕛 说明

Demo 中主进程的目录为 src/app/main.js ,渲染进程目录为 src/client 。如运行过程出现问题,可优先通过常见问题查找解决。

## 步骤5: 自实现

**安装 Electron SDK** 使用如下命令,安装 Electron SDK 最新版本 在命令行执行:

npm install im\_electron\_sdk

#### 完成 SDK 初始化

1. 在 TimMain 中传入您的 sdkAppID 。

```
// 主进程
const TimMain = require('im_electron_sdk/dist/main')
const sdkappid = 0;// 可以去腾讯云即时通信IM控制台申请
const tim = new TimMain({
    sdkappid:sdkappid
})
```

2. 调用 TIMInit ,完成 SDK 初始化。

```
//渲染进程
const TimRender = require('im_electron_sdk/dist/render')
const timRender = new TimRender();
// 初始化
timRender.TIMInit()
```

#### 3. 登录测试用户。

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 timRender.TIMLogin 方法,登录一个测试用户。 当返回值 code 为0时,登录成功。

```
const TimRender = require('im_electron_sdk/dist/render')
const timRender = new TimRender();
let {code} = await timRender.TIMLogin({
    userID:"userID",
    userSig:"userSig" // 参考userSig生成
})
```

#### 🕛 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 。

#### 发送信息

此处以发送文本消息为例, code 返回 0 则为消息发送成功。 代码示例:

const TimRender = require('im\_electron\_sdk/dist/render')



```
const timRender = new TimRender();
let param:MsgSendMessageParamsV2 = { // param of TIMMsgSendMessage
conv_id: "conv_id",
conv_type: 1,
params: {
    message_elem_array: [{
        elem_type: 1,
        text_elem_content:'Hello Tencent!',
      }],
      message_sender: "senderID",
    },
    callback: (data) => {}
    }
let {code} = await timRender.TIMMsgSendMessageV2(param);
```

#### () 说明

如果发送失败,可能是由于您的 SDKApplD 不支持陌生人发送消息,您可至控制台开启,用于测试。请点击此链接,关闭好友关系链检查。

#### 获取会话列表

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。

常见应用场景为:

在启动应用程序后立即获取会话列表,然后监听长链接以实时更新会话列表的变化。

let param:getConvList = {
 userData:userData,
 }
let data:commonResult<convInfo[]> = await timRenderInstance.TIMConvGetConvList(param)

此时,您可以看到您在上一步中,使用另一个测试账号发来的消息的会话。

#### 接收消息

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. 监听长链接,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
```

```
let param:MsgGetMsgListParams = {
    conv_id: conv_id,
    conv_type: conv_type,
    params: {
        msg_getmsglist_param_last_msg: msg,
        msg_getmsglist_param_count: 20,
        msg_getmsglist_param_is_remble: true,
        },
        user_data: user_data
    }
    let msgList:commonResult<Json_value_msg[]> = await timRenderInstance.TIMMsgGetMsgList(param);
```

#### 监听实时获取新消息 绑定 callback 示例代码如下:

```
let param : TIMRecvNewMsgCallbackParams = {
    callback: (...args)=> {},
    user_data: user_data
    }
timRenderInstance.TIMAddRecvNewMsgCallback(param);
```

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,离线推送,本地搜索 等相关功能开发。


详情可查看 API 文档。

### 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

#### Electron 日志目录在哪?

日志默认在 path.resolve(os.homedir(), ".tencent-im/sdk-log")下(可以在 nodejs 的 cli 环境下执行)。 如果在调用 TIMInit API 时传入了 config\_path,那日志就在传入的 config\_path 目录下。

#### 支持哪些平台?

目前支持 macOS 和 Windows 两个平台。

#### 错误码如何查询?

IM SDK 的 API 层面错误码,请查看 错误码。

#### 安装开发环境问题,出现

npm ERR! gyp ERR! stack TypeError: Cannot assign to read only property 'cflags' of object '#<Object>' 错误如何 解决?

请降低 node 版本,建议使用16.18.1。

#### 安装开发环境问题,出现 gypgyp ERR!ERR 错误如何解决?

请参见 gypgyp ERR!ERR!。

## 执行 npm install 出现错误 npm ERR! Fix the upstream dependency conflict, or retry ,如何解决?

npmV7之前的版本遇到依赖冲突会忽视依赖冲突,继续进行安装 npmV7版本开始不会自动进行忽略,需要用户手动输入命令 请执行以下命令:

## npm install --force

#### 执行 npm run start 出现错误 Error: error:0308010C:digital envelope routines::unsupported ,如何解决?

请降低node版本,建议使用16.18.1。

#### Mac 端 Demo 执行 npm run start 会出现白屏,如何解决?

Mac 端执行 npm run start 会出现白屏,原因是渲染进程的代码还没有 build 完成,主进程打开的3000端口为空页面,当渲染进程代码 build 完成重新刷新窗 口后即可解决问题。或者执行 cd src/client && npm run dev:react , npm run dev:electron ,分开启动渲染进程和主进程。

#### vue-cli-plugin-electron-builder 构建的项目如何使用 native modules ?

使用 vue-cli-plugin-electron-builder 构建的项目使用 native modules 请参见 No native build was found for platform = xxx 。

#### 用 webpack 构建的项目如何使用 native modules ?

自己使用webpack 构建的项目使用native modules 请参见 Windows 下常见问题。

#### 出现 Dynamic Linking Error?

Dynamic Linking Error. electron-builder 配置

#### extraFiles:[

"from": "./node\_modules/im\_electron\_sdk/lib/",





## 使用 electron-vite 出现 \_\_dirname is not defined ?

由于 electron-vite 不支持在渲染进程 (renderer) 进行进程间的通信,需要将 IM SDK 写到 preload 中使用 (主进程的代码正常写到主进程即可)。具体可参 考 electron-vite 文档。

使用方法相同,可参考文档的实例代码。下面以初始化为例,使用方法如下:



# 联系我们

点此进入 IM 社群,享有专业工程师的支持,帮助您解决您的难题。



# 快速入门(Unity)

最近更新时间: 2024-04-15 14:57:51

### 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 。



於 購訊云 息號	ī 云产品 → │ 网站备案 +
即时通信 IM	← UserSig生成&校验 ▼
弐 基本配置	
品 功能配置	签名(UserSig)生成⊥具 登录鉴权介绍 ☑
晶 群组管理	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。
③ 回调配置	请输入用户名 (UserID)
🕣 安全打击	user1
贵 数据监控器	密钥 (Key)
@ 辅助工具	
• 离线推送自查	
・ UserSig生成&校 验	生成签名 (UserSig)
	当前生成的签名(UserSig)是:
	复制签名 (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。

• • •		使用 Unity 2020.3.23	f1c1 创建新项目
模板			设置
	©	()	项目名称* im_unity_test
2D	3D	High Definition RP	位置* /Users/ •••
	★ 下载	<u>.</u> ₽	□ 启用 PlasticSCM 并同意 政策条款
Universal Render Pipeline	2D Platformer Microgame	Karting Microgame	
* •	*	± (9)	
			取消 创建

或打开一个已有 Unity 项目。

2. 通过 IDE (如: Visual Studio Code) 打开项目:



ſ'n	资源管理器 ···	() manifest.json M ×
	> 打开的编辑器	Packages > () manifest.json > () dependencies
<u></u>	$\sim$ IM-UNITY-UIKIT	20 province and the second sec
	✓ Assets	21 "com.unity.modules.imageconversion": "1.0.0",
Q	> Editor	22 "com.unity.modules.imgui": "1.0.0",
	> Pretabs	23 🕐 Com.unity.modules.jsonserialize": "1.0.0",
j°2	> Library	24 ····································
~		25 ·····"com.unity.modules.physics": "1.0.0",
±2	✓ Packages	26 ·····"com.unity.modules.physics2d": "1.0.0",
	{} manifest.json M	27 ·····"com.unity.modules.screencapture": "1.0.0",
<u>L</u> 0	{} packages-loc M	28 ·····"com.unity.modules.terrain": "1.0.0",
	> Temp	29 ·····"com.unity.modules.terrainphysics": "1.0.0",
ш		30 ·····"com.unity.modules.tilemap": "1.0.0",
	<ul> <li>.gitignore</li> </ul>	31 ····· "com.unity.modules.ui": "1.0.0",
	<ol> <li>README.md</li> </ol>	32 "com.unity.modules.uielements": "1.0.0",
$\bigcirc$		33 ·····"com.unity.modules.umbra": "1.0.0",
		34
		35 ·····"com.unity.modules.unitywebrequest": "1.0.0",
ß		36
Ť		37 "com.unity.modules.unitywebrequestaudio": "1.0.0",
		38 ·····"com.unity.modules.unitywebrequesttexture": "1.0.0",
		39 ·····"com.unity.modules.unitywebrequestwww": "1.0.0",
		40 ···· com.unity.modules.vehicles": "1.0.0",
		41 "com.unity.modules.video": "1.0.0",
		42 ····"com.unity.modules.vr": "1.0.0",
		43 ···· com.unity.modules.wind": "1.0.0",
		44 com.unity.modules.xr": "1.0.0",
_		45 com.tencent.imsdk.unity":"https://github.com/TencentCloud/chat-sdk-unity.git#unity
		46
Ø		47 }

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



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

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
    ł
     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 json_param, string user_data)=>{
string desc,
     // 消息发送异步结果
       // 消息发送同步返回的消息ID messageld
}
```

## 🕛 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,发来消息的会话。

#### 监听长链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. 监听长链接,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-moudle.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="./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 <mark>云端审核</mark> 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 untiy 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。



# 快速入门(UE)

最近更新时间: 2024-03-05 16:35:21

本文旨在介绍如何快速将腾讯云即时通信 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。

即时通信 IM	应用管理 IM 拍	支术服务交流群 🖸	IM 出海交流专区
<b>壬</b> 应用管理	创建新应用	申请开发版	
管理与配置	应用名称	SDKAppID (	) 应用版本 ()

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。

创建新应用	申请开发版 获	取 SDKAppID 信息					请输入 SDKApp	DID 或 应用名称 或 标签		Q
应用名称	SDKAppID (j)	应用版本 ()	服务状态	数据中心 🛈 🔻	创建时间	到期时间	标签 (i)	操作	获取密钥信息	
test1	1600XXXXXX	旗舰版	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 访	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 Luncher,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:

资源管理器	
∨ UE_APP	
> Binaries	
> Config	
> Content	
> DerivedDataCache	
> Intermediate	
> Saved	
> Source	
> UE_APP.xcworkspace	
≡ UE_APP.uproject	
	Γ.
·骤3:下载 Plugin	



## 步骤4: Plugin 引入

在项目根目录创建插件目录,命名问 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	e > 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:



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 控制台中见到如下日志,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 <mark>云端审核</mark> 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 \$pathToframgeWork

- 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。



# 快速入门(Flutter)

最近更新时间: 2024-04-12 15:13:01



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 SDK 或 UIKit 组件库的方法。

# 环境要求

环境	版本
Flutter	IM SDK 最低要求 Flutter 2.2.0 版本,TUIKit 集成组件库最低要求 Flutter 3.0.0 版本。
Android	Android Studio Dolphin   2021.3.1 及以上版本,App 要求 Android 7.0 及以上版本设备。
iOS	Xcode 12.0 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 Flutter 全平台的即时通信 IM SDK 及 TUIKit,帮助您一套代码,全平台运行。

平台	无 UI SDK (tencent_cloud_chat_sdk)	含 UI 及基础业务逻辑 TUIKit (tencent_cloud_chat_uikit)
iOS	支持	支持
Android	支持	支持
Web	支持,4.1.1+2版本起	支持,0.1.5版本起
macOS	支持,4.1.9版本起	支持, 2.0.0 版本起
Windows	支持,4.1.9版本起	支持,2.0.0 版本起
<mark>混合开发</mark> (将 Flutter SDK 添加至现有原生应 用)	5.0.0版本起支持	1.0.0版本起支持

## () 说明

Web/macOS/Windows 平台需要简单的几步额外引入,详情请查看本文 拓展更多平台。

# 体验 Demo

您可以通过我们的 DEMO,快速在线体验 TUIKit 各项能力。 以下各版本 DEMO,均由同一 Flutter 项目引入TUIKit 制作打包而成。单击获取 Demo 源码 。

平台

体验链接 / 下载二维码

备注



Android / iOS		扫描下载应用程序,适用于 Android 和 iOS。自动识别平台。
Web		支持桌面和移动浏览器,并根据需要自动调整布局。与下面链接的网站相同。
Web	立即访问	支持桌面和移动浏览器,并根据需要自动调整布局。与之前的二维码链接到同一个网站。
macOS	立即下载	按住 Control 键点按应用程序图标,然后从快捷菜单中选取"打开"。>>>单击查看详细步骤
Windows	立即下载	我们 Demo 的 Windows 版本,它是一个 UWP 应用程序。

# 前序工作

- 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 。
- 3. 在 IM 控制台 选择您的应用,在左侧导航栏依次点击 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。

🔗 腾讯云 🛛 🕸	: 云产品 -   网站备案 +
即时通信 IM	← UserSig生成&校验 -
ご 其木配男	
品 功能配置	签名(UserSig)生成工具 登录鉴权介绍 🖸
晶 群组管理	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。
③ 回调配置	请输入用户名 (UserID)
🕣 安全打击	user1
🗟 数据监控器	密钥 (Key)
@ 辅助工具	
• 离线推送自查	
● UserSig生成&校 验	生成签名 (UserSig)
	当前生成的签名(UserSig)是:
	复制签名 (UserSig)

#### 🕛 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 。

## 选择合适的方案集成 Flutter SDK

IM 提供了三种方式来集成,您可以选择最合适的方案来集成:



集成方式	适用场景
使用 DEMO 修改	IM Demo 是一个完整的聊天 App,代码已开源,如果您需要实现聊天类似场景,可以使用 Demo 进行二次开发。
含 UI 集成	IM 的 UI 组件库 TUIKit 提供了通用的 UI 组件,例如会话列表、聊天界面和联系人列表等,开发者可根据实际业务需求通过该 组件库快速地搭建自定义 IM 应用。 <b>推荐优先使用该方案</b> 。
自实现 UI 集成	如果 TUIKit 不能满足您应用的界面需求,或者您需要比较多的定制,可以使用该方案。

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。

# 方案一:使用 Demo 修改

## 跑通 Demo

1. 下载 Demo 源码、安装依赖:

# Clone the code git clone https://github.com/TencentCloud/chat-demo-flutter.git # Install dependencies flutter pub get

#### 2. [可选] 在Web 上运行或部署

如果您想在 Web 上运行或部署此示例应用程序,您需要先完成一些额外的步骤。 导航到您项目的 web/ 目录,并使用 npm 或 Yarn 安装所需的 JavaScript 依赖项。

cd web npm instal

完成这些步骤后,您就可以在Web上运行或部署示例应用程序了。

3. 运行 Demo 项目:

```
#启动demo项目,请替换SDK_APPID、KEY两个参数
```

flutter run --dart-define=SDK\_APPID={YOUR\_SDKAPPID} --dart-define=ISPRODUCT\_ENV=false --dart-define=KEY= {YOUR\_KEY}

## 🕛 说明

- --dart-define=SDK\_APPID={YOUR\_SDKAPPID} 其中 {YOUR\_SDKAPPID} 需替换成您自己应用的 SDKAppID。
- --dart-define=ISPRODUCT\_ENV=false 对开发生产环境做判断,如您是开发环境请用 false。
- --dart-define=KEY={YOUR\_KEY} 其中 {YOUR\_KEY} 需替换成官网控制台中的密钥(Key)信息。
- 请注意,把大括号去掉。
- 请注意,如果直接运行 Demo 源码,离线推送默认不可用,需要向厂商申请离线推送配置,才能使用。

### 如果需要在 IDE 中运行

```
Android Studio
```

- 1. 在 Android Studio 中安装 Flutter 和 Dart 插件。
- Mac 平台: 打开插件设置(在 v3.6.3.0 以上的系统打开 Preferences > Plugins) => 选择 Flutter 插件并点击 安装 => 当弹出安装 Dart 插件提示 时,点击 Yes => 当弹出重新启动提示时,点击 Restart。
- Linux 或者 Windows 平台: 打开插件设置 (位于 File > Settings > Plugins)= > 选择 Marketplace (扩展商店),选择 Flutter plugin 然后点击 Install (安装)。





## 2. 打开项目并获取依赖。

在 Android Studio 中打开 chat-demo-flutter 目录。



并在该路径执行命令安装依赖。





	Run/D	ebug Configurations	
+ - 🖻 🛤 🕸			
🗸 🎸 Flutter			
🗲 main.dart			
	Additional run args:	art-define=SDK_APPID=dart-def	ine=ISPRODUCT_ENV=false 🖉
	Additional attach args:		
	Build flavor:		
	<ul> <li>Before launch</li> </ul>		
	🗌 Show this page 🗹		
Edit configuration templates			
?			

## 4. 创建 Android 模拟器。

启动您刚安装好的模拟器,并选中其。



单击界面右上角 Device Manager,完成 Create devices,创建模拟器。如果您需要使用 Google FCM 推送能力,建议安装支持 Google Play Store 的设备。



## 5. 运行项目。

根据需要,单击下图左侧 Run ,或右侧 Debug,以运行项目。





## 🕛 说明

UI 可能会有部分调整更新,请以最新版为准。

#### Xcode

- 1. 在 Xcode 中打开 chat-demo-flutter/ios 目录。
- 2. 连接 iPhone 真机,单击 Build And Run, iOS 工程等待编译完成,会有新窗口弹出 Xcode 工程。
- 3. 打开 iOS 工程,设置主 Target 的 Signing & Capabilities (需要苹果开发者账号),让项目可以在 iPhone 真机上运行。
- 4. 启动项目,在真机上进行 Demo 的调试。



## Demo 代码结构概览

#### () 说明

我们 Demo 的 UI 及业务逻辑部分,使用 Flutter TUIKit。Demo 层本身仅用于构建 App,处理导航跳转及调用实例化 TUIKit 中各个组件。

文件夹	介绍
lib	程序核心目录
lib/i18n	国际化相关代码。这里的国际化,不包含 TUIKit 本身的国际化能力和国际化词条,您可按需引入
lib/src	项目主体目录
lib/src/pages	本 Demo 几个重点导航页。项目初始化完成后,由 app.dart 负责展示加载动画,并判断登录态,将用户引导至 login.dart 或 home_page.dart 。用户登录后,会将登录信息通过 shared_preference 插件,存储至本地。以后每次 启动应用,若在本地发现原来的登录信息,则自动使用该信息进行登录,若无或登录失败,则引导至登录页。自动登录过程中, 用户还在 app.dart ,可看到加载动画。 home_page.dart 含一个底部 Tab,支撑本 Demo 的四个主功能页的切换。
lib/src/pages/cross_ platform/wide_scree n/	桌面端适配相关文件。包含一个桌面端的全局 home_page.dart 页面,还有若干用于各个子组件展示的 dart 文件。
lib/utils	一些工具函数类。

基本上, lib/src 内每个 dart 文件引入了一个 TUIKit 组件,在文件内,实例化组件后,即可渲染页面。 主要文件如下:



lib/src 主要文件	文件介绍
add_friend.dart	申请添加好友页面,使用 TIMUIKitAddFriend 组件
add_group.dart	申请入群页面,使用 TIMUIKitAddGroup 组件
blacklist.dart	黑名单列表页面,使用 TIMUIKitBlackList 组件
chat.dart	主聊天页面,使用全套TUIKit聊天能力,使用 TIMUIKitChat 组件
chatv2.dart	主聊天页面,使用原子化能力,使用 TIMUIKitChat 组件
contact.dart	联系人页面,使用 TIMUIKitContact 组件
conversation.dart	会话列表界面,使用 TIMUIKitConversation 组件
create_group.dart	发起群聊页面,纯Demo实现,未使用组件
group_application_list.dart	入群申请列表页面,使用 TIMUIKitGroupApplicationList 组件
group_list.dart	群列表页面,使用 TIMUIKitGroup 组件
group_profile.dart	群资料及群管理页面,使用 TIMUIKitGroupProfile 组件
newContact.dart	联系人好友申请页面,使用 TIMUIKitNewContact 组件
routes.dart	Demo的路由,导航至登录页 login.dart 或主页面 home_page.dart 。
search.dart	全局搜索及会话内搜索页面,使用 TIMUIKitSearch (全局搜索)及 TIMUIKitSearchMsgDetail (会 话内搜索)组件
user_profile.dart	用户信息及关系链维护页面,使用 TIMUIKitProfile 组件

在移动端上,大部分 TUIKit 组件需要传入导航跳转方法,因此需要 Demo 层处理 Navigator 。 以上介绍了我们的 Demo,您可以直接直接修改它二次开发,或参照它实现您的业务需求。

# 方案二: 含 UI 集成,使用 TUIKit 组件库,半天完成 IM 能力植入

# UIKit V2 新版

UIKit V2 新版现已开放接入使用!





UIKit V2 经过全面重构,为您带来更出色的开发及使用体验。

包括全新简洁美观的 UI 界面, 开发者更友好的程序架构, 内置可切换的黑暗/日间模式, 更全面的国际化支持, 手机平板桌面端及 Web 全平台支持, 更多细节丰富体 验, 更进一步的性能等更多强大的能力及设计.

您可阅读 集成基础功能 > Flutter 接入使用.

## UIKit 稳定版

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,例如会话列表、聊天界面和联系人列表等,开发者可根据实际业务需求通过该组 件库快速地搭建自定义 IM 应用。参见 TUIKit 图文介绍。

本部分为快速使用 TUIKit 介绍,详细入门指引可参见 TUIKit 集成基础功能。

3:2		÷-	••	Q. Search	+ Cus	tomer using Flutter		
( 🕖	Customer using Flutter		Char	Customer using Flutter	Now	115	7	
	11:57, yesterday		Contac	Miles , Runlin Wang,	Now		Read Welcome to Tencent Cloud Chat Flutter TUIKit	4
Deve	Welcome to Tencent Cloud Chat Flutter TUIKit	4	÷.	What's going on?		We are confident that our Flutter Chat SDK and experience.	TUIKit can ignite your creativity and enhance your product	4
				910635538 Let's go to the cafeteria and ha	Now Now	Read		
	We are confident that our Flutter Chat SDK and TUIK	it 📲	_	222	16	「你们的 TUIKK 都設节省了很多时间搭建应用内部天模块 ♥   Flutter Support  ♥   Customer using Flutter		
	and enhance your product			Fiel p2/10.mp4		Your TUKit has saved me a lot of time in building chat module:	s in the app.	
Read				Hi welcome to this group chat	Pi	Translate successfully		
	你们的 TUIKIt 帮我节省了很 多时间搭建应用内聊天模块			Flutter 支持团队等12人	14		10245383: 你们的TUNOI帮我节啥了很多时间跟建应用内聊天模块	4
	🗢   Flutter Support			New Group Chat			Oh, great! 🔌 I'm happy to hear it. 🤣	
	👏   Customer using Flutter			Try it out		0 1	Read	
	Your TUIKit has saved me a lot of time in building chat modules in the app.			2011年1月1日1月1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日	Wed 5		<b>(</b>	ą
	<ul> <li>Translate successfully</li> </ul>			Elon, Hi Flutter, Jobs ( (Video Call): Call canceled	4/m	KROEE	TANT	-
	10045363: 你们的 TUKIt 模拟节音了很多时间目 建成用内副天模块	3		Elon and 7 more o	405			
	Oh, great! 🚴 I'm happy to hear it 😆		4	New Group Chat o	0/01			
٩		: ⊕						

## 前提条件

您已经完成创建 Flutter 项目,或有可以基于的 Flutter 项目。

接入步骤

配置权限



由于 TUIKit 运行,需要拍摄/相册/录音/网络等权限,需要您在 Native 的文件中手动声明,才可正常使用相关能力。

#### Android

打开 android/app/src/main/AndroidManifest.xml ,在 <manifest></manifest> 中,添加如下权限。

<ul> <li><ul> <li><ul> <li><ul> <li><ul> <li><ul></ul></li></ul></li></ul></li></ul></li></ul></li></ul>	
android:name="android.permission.INTERNET"/>	
<ul> <li><ul> <li><ul> <li><ul></ul></li></ul></li></ul></li></ul>	
android:name="android.permission.RECORD_AUDIO"/>	
<ul> <li><ul> <li><ul> <li><ul></ul></li></ul></li></ul></li></ul>	
android:name="android.permission.FOREGROUND_SERVICE"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.ACCESS_NETWORK_STATE"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.VIBRATE"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.READ_EXTERNAL_STORAGE"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.CAMERA"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.READ_MEDIA_IMAGES"/>	
<uses-permission< td=""><td></td></uses-permission<>	
android:name="android.permission.READ_MEDIA_VIDEO"/>	

## iOS

#### 打开 ios/Podfile ,在文件末尾新增如下权限代码。

## 🕛 说明

如您需要用到推送能力,还需要添加推送相关权限,详情可查看 Flutter 厂商消息推送插件集成指南。

### 安装 IM TUIKit

我们的 TUIKit 已经内含 IM SDK,因此仅需安装 tencent\_cloud\_chat\_uikit ,不需要再安装基础 IM SDK。

#在命令行执行: flutter pub add tencent\_cloud\_chat\_uikit



如果您的项目需要支持 Web,请在执行后续步骤前,查看 Web 兼容说明章节,引入 JS 文件。

#### 初始化

- 1. 在您应用启动时,初始化 TUIKit。
- 2. 请务必保证先执行 TIMUIKitCore.getInstance() ,再调用初始化函数 init() ,并将您的 sdkAppID 传入。
- 3. 为方便您获取API报错及建议提醒用户的提示语,此处建议挂载一个 onTUIKitCallbackListener 监听,详见此部分。

### /// main.dart

import 'package:tencent\_cloud\_chat\_uikit/tencent\_cloud\_chat\_uikit.dart';

final CoreServicesImpl \_coreInstance = TIMUIKitCore.getInstance();

// init sdk. call this api where you need.

\_coreInstance.init(

sdkApplD: 0, // Replace 0 with the SDKApplD of your IM application when integrating \_\_\_\_\_/ language: LanguageEnum.en, // 界面语言配置,若不配置,则跟随系统语言

loglevel: LogLevelEnum.V2TIM\_LOG\_DEBUG,

onTUIKitCallbackListener: (TIMCallback callbackValue){ }, // [建议配置,详见此部分]

(https://cloud.tencent.com/document/product/269/70746#onTUIKitCallbackListener)

listener: V2TimSDKListener(),

);

#### 🕛 说明

请在本步骤 await 初始化完成后,再执行后续步骤。

#### 登录测试账户

- 1. 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。
- 2. 调用 \_coreInstance.login 方法,登录一个测试账户。

import 'package:tencent\_cloud\_chat\_uikit/tencent\_cloud\_chat\_uikit.dart';

final CoreServicesImpl \_coreInstance = TIMUIKitCore.getInstance(); \_coreInstance.login(userID: userID, userSig: userSig);

#### () 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 。

#### 实现: 会话列表页面

您可以以会话列表作为您的 IM 功能首页,其涵盖了与所有有聊天记录的用户及群聊的会话。



Cascuch			Ð		Own	1	Customer using Flutter (Emol)	Nov						
<ul> <li>Montane using Future 1 manage</li> <li>Montane using Future 1 manage</li> <li>Montane 1 manage</li>     &lt;</ul>		Q Search			Contacts	*	Miles , Runlin Wang, What's going on?	Nov					Welcome to Tencent Cloud Chet Flutter TUIKIt	4
<ul> <li>See Routing Vange, to voor de la see Routing Vange,</li></ul>	Custo We're	omer using Flutter looking forward to cooperat	2 days ago ing with			4	910635538 Let's go to the cafeteria and ha	Nov av						9
Image: Server Group Chat: two   Image: Server Group Chat: two   Image: Server Ser	Miles What's	s , Runlin Wang, s going on?	2 days ago	2		4	222 (File) p6:10.mp4	н	<b>1</b> 13	你们的 TUIKk 都报号	8省了很多时间搭建应用内聊天模块 Notice Function Platter			
<ul> <li>Futter 2,HBU, 912A too</li> <li>Futter 2,HBU, 912A too</li> <li>91330.064 to transmut</li> <li>91330.064 to transmut</li> <li>91330.064 to transmut</li> <li>1000000000000000000000000000000000000</li></ul>	New (Image	Group Chat	Nov			ili	New Group Chat Hi welcome to this group chat	ы		Your TUIKit has save © translate successfully	ed me a lot of time in building cha	t modules in the ap	эр.	
913380464       weeking         10 Statuted       Weeking         11 Tracent Chat Team       weeking         12 Tracent Chat Team       weeking         13 Tracent Chat Team       weeking         14 Tracent Chat Team       weeking         15 Tracent Chat Team       weeking         16 Tracent Chat Team       weeking         17 Tracent Chat Team       weeking         18 Tracent Chat Team       weeking         19 Trac	Flutte Sure le	er 支持团队 等12人 <sup>et's go</sup>	Now			ili	Flutter 支持团队等12人 Good merring sin	ы						4
Interact Chat Team www.     Justice/[ustater]     Team www.     Interact Chat Tea	9133 [Video	80464 Callj: No answer	Yesterday			ili	New Group Chat Try it out	ы				e 🗆 > Read		
<ul> <li>Tencent Chai Team works</li> <li>Tencent Chai Team works</li> <li>User Platter Support plotted the group</li> <li>Bin and Toney loss</li> <li>Bin and Toney loss</li> <li>List spit the caleful and the some food.</li> </ul>	Tence [soldie	ent Chat Team	Yesterday			2	在线客银示例 局可目动组示说作项: 由于型长8	Wed 17					<b>(</b> )	4
910635538 Zrans op Litt's go to the celletria and takes some field.	Tence	ent Chat Team	Yesterday			ili	Elon, Hi Flutter, Jobs ( [Video Cal]: Call canceled	04/11	© X				Trait	
Let's go to the cateteria and have some tood.	9106	35538	2 days ago			ili	Elon and 7 more o (Chat history)	04/05						
New Grave Chail Cast	Let's g	to the cafeteria and have	some food.			ste	New Group Chat	03/31	_	_	_	_		

请创建一个 Conversation 类, body 中使用 TIMUIKitConversation 组件, 渲染会话列表。

#### 实现:会话聊天页面

该页面由顶部主体聊天历史记录及底部发送消息模块组成。

9	Customer using Flutter		 Ches .	Customer using Flutte	H Nove			11:57		
	11:57, yesterday		 Contacts					Read	Welcome to Tencent Cloud Chat Flutter TUIKit	4
	Welcome to Tencent Cloud	-	 ÷.	What's going on?	- Nyur		We are confident that our Flutter Chat t	SDK and TUIKit ca	n ignite your creativity and enhance your product	4
Rea	Chat Hutter TUIKIt		 1	910635538	Now		experience.			-
	We are confident that our Elutter Chet SDK and TUIKit	<b>a</b>			1	\$K1081	TUIKit 帮我节省了很多时间搭建应用内聊天模块			
	can ignite your creativity and enhance your product			222 (File) p6/10.mp4	20	•   11	atter Support 👋   Customer using Flutter			
Rea	experience.			New Group Chat	Fri	Your TL © Transit	IKit has saved me a lot of time in building cha In successfully	t modules in the a	зр.	
	你们的 TUIKit 帮我节省了很			Change Training March						
	多时间拾细应用内聊大模块 ■ Dutter Surgert			Good morning sir					你们的 TUKE 帮我节留了很多时间搭建应用内服天概块	
	📚   Customer using Flutter			New Group Chat	10			0 = > -	e   Flutter Support Customer using Flutter	
	Your TLIIKit has saved me a							Read		
	lot of time in building chat modules in the app.		 4	4.04各级示例 45月15日末期前达46年期前达46年期前前5日	Wed 想长啊					4
	C Translate successfully			Elon, Hi Flutter, Jobs	04/11				Trant	
	10045363:	4		(voio carji- car caveri		KOL				
	你们的 TURI 帮批节留了很多时间括 建成用内服天模块	_		Elon and 7 more (Chat history)	04/06					
	Oh, great! À I'm happy to hear it 🥝		 4	New Group Chat	03/31					
		© ⊕								

请创建一个 Chat 类, body 中使用 TIMUIKitChat 组件,渲染聊天页面。 您最好传入一个 onTapAvatar 事件的处理函数,用于跳转至联系人的详细信息页。关于 UserProfile 类,会在下一步讲解。

#### 实现: 用户详情页面

该页面默认,可在只传入一个 userID 的情况下,自动根据是否是好友,生成用户详情页。 请创建一个 UserProfile 类, body 中使用 TIMUIKitProfile 组件,渲染用户详情及关系链页面。



Profi	ile		Contacts Groups	10045363 D: 12045983	a Ale
10045363		Contac	the New contracts	Tencent Cloud Chat - 勝讯云 IM	
Status: Tencent Clou	id Chat - 腾讯云 M	÷	Blocked Users	Remarks Customer using Flatter	
marks	Customer using > Flutter >		· 不会認知思的机器人A	Gender Male	
nder	Male		C	Date of birth Not Specified	
te of birth	Not Specified		Customer using Flutter		
arch Chat History				Pin chat to top	
,			🕵 m	Mute notifications	
ck user	0		<b>B</b> <sub>0</sub> 222		
h chat to top				Send message	
te notifications		_		Delete friend	
Send me	essage	_			
Voice	call				
Video	call	4			
Delete I	triand.				

此时,您的应用已经可以完成消息收发,管理好友关系,展示用户详情及展示会话列表。

#### 更多能力

您还可以继续使用以下 TUIKit 插件快速实现完整 IM 功能。

- TIMUIKitContact: 联系人列表页面。
- TIMUIKitGroupProfile: 群资料页面,使用方式与 TIMUIKitProfile 基本一致。
- TIMUIKitGroup: 群列表界面。
- TIMUIKitBlackList:黑名单列表界面。
- TIMUIKitNewContact: 联系人(好友)申请列表。如需在外部显示小红点,可使用 TIMUIKitUnreadCount 小红点组件,其会自动挂载监听。
- 本地搜索: TIMUIKitSearch 全局搜索组件,支持全局搜索联系人/群组/聊天记录,也支持使用 TIMUIKitSearchMsgDetail 在特定会话中搜索聊天记录。两 种模式取决于是否传入 conversation 。

UI 组件全貌可参见 本图文概览 或 详细文档。

本部分仅为快速使用 TUIKit 介绍,详细入门指引可参见 TUIKit 集成基础功能。

## 方案三: 自实现 UI 集成

## 前提条件

您已经完成创建 Flutter 项目,或有可以基于的 Flutter 项目。

## 接入步骤

### 安装 IM SDK

#### 本节详细文档

使用如下命令,安装 Flutter IM SDK 最新版本。 在命令行执行:

#### flutter pub add tencent\_cloud\_chat\_sdk

## 🕛 说明

如果您的项目还同时需要用于 Web 或 桌面端(macOS、Windows),还需要一些额外步骤,具体请点击链接查看。

## 完成 SDK 初始化

## 本节详细文档

```
调用 initSDK ,完成 SDK 初始化。
将您的 sdkAppID 传入。
```

import 'package:tencent\_cloud\_chat\_sdk/enum/V2TimSDKListener.dart'; import 'package:tencent\_cloud\_chat\_sdk/enum/log\_level\_enum.dart';



import 'package:tencent\_cloud\_chat\_sdk/tencent\_cloud\_chat\_sdk.dart'; TencentImSDKPlugin.v2TIMManager.initSDK( sdkAppID: 0, // Replace 0 with the SDKAppID of your IM application when integrating loglevel: LogLevelEnum.V2TIM\_LOG\_DEBUG, // Log listener: V2TimSDKListener(), }

在本步骤,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 TencentImSDKPlugin.v2TIMManager.login 方法,登录一个测试账户。 当返回值 res.code 为0时,登录成功。

```
import 'package:tencent_cloud_chat_sdk/tencent_cloud_chat_sdk.dart';
V2TimCallback res = await TencentImSDKPlugin.v2TIMManager.login(
    userID: userID,
    userSig: userSig,
):
```

#### () 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,其流程为:

```
1. 调用 createTextMessage(String) 创建一个文本消息。
```

2. 根据其返回值,拿到消息 ID。

```
3. 调用 sendMessage() 发送该ID的消息。 receiver 可填入您此前创建的另一个测试账户 ID。发送单聊消息无需填入 groupID 。
代码示例:
```

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

V2TimValueCallback<V2TimMsgCreateInfoResult> createMessage =

    await TencentImSDKPlugin.v2TIMManager

    .getMessageManager()

    .createTextMessage(text: "The text to create");

String id = createMessage.data!.id!; // The message creation ID

V2TimValueCallback<V2TimMessage> res = await TencentImSDKPlugin.v2TIMManager

    .getMessageManager()

    .sendMessage(

    id: id, // Pass in the message creation ID to

    receiver: "The userID of the destination user",

    groupID: "The groupID of the destination group",

    );
```

🕛 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。

		9						
	÷		Customer using Flutte	M Now				
Q Search		Contacts	Miles , Runlin Wang, .	New			Welcome to Tencent Cloud Chat Flutter TUIKit	4
Customer using Flutter We're looking forward to coopera	2 days ago ating with	÷.	What's going on?     What's going on?     Second Seco	Now	We are confident that our Flutter Chat SI seed	DK and TUIKit car	s ignite your creativity and enhance your product	4
Miles , Runlin Wang, What's going on?	2 days ago		222 (File) p6:10.mp4					
New Group Chat	Now		New Group Chat H welcome to this group	chat Fri	Your TURKI has saved me a lot of time in building chat	modules in the ap	эр.	
Flutter 支持团队 等12人 Sure let's go	Now		Flutter 支持团队等12。 Cood morning sir	х н				4
913380464 [Video Call]: No answer	Yesterday		New Group Chat Try it out	PI		e = > Read		
Tencent Chat Team [soldier][soldier]	Yesterday		在线客报示例     和时目动组示说示: 由于     日	Wed 181017			Ö	á
Tencent Chat Team User Flutter Support joined the g	Yesterday		tion, Hi Flutter, Jobs (Video Cal): Call cancel	64/11 0	xcmem			
910635538 Let's go to the cafeteria and have	2 days ago a some food.		Chat history	04/05				
■	÷							

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

为了获取会话列表,需要维护 nextSeq ,记录当前位置。



此时,您可以看到您在上一步中,使用另一个测试账号,发来消息的会话。

## 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。

```
await TencentImSDKPlugin.v2TIMManager
.getConversationManager()
.setConversationListener(
    listener: new V2TimConversationListener(
        onConversationChanged: (List<V2TimConversation> list) {
        _onConversationListChanged(list);
    },
        onNewConversation:(List<V2TimConversation> list) {
        _onConversationListChanged(list);
    },
```

2. 处理回调事件,将最新的会话列表展示在界面上。



import 'package:tencent\_cloud\_chat\_sdk/tencent\_cloud\_chat\_sdk.dart';

```
List<V2TimConversation> _conversationList = [];
_onConversationListChanged(List<V2TimConversation> list) {
  for (int element = 0; element < list.length; element++) {
    int index = _conversationList.indexWhere(
      (item) => item!.conversationID == list[element].conversationID);
  if (index > -1) {
    _conversationList.setAll(index, [list[element]]);
  } else {
    _conversationList.add(list[element]);
  }
  }
```

# 接收消息

## 本节详细文档

通过腾讯云 IM Flutter 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:

import 'package:tencent\_cloud\_chat\_sdk/tencent\_cloud\_chat\_sdk.dart';

V2TimValueCallback<List<V2TimMessage>> res = await TencentImSDKPlugin
.v2TIMManager
.getMessageManager()
.getGroupHistoryMessageList(
 groupID: "groupID",
 count: 20,
 lastMsgID: "",
);

```
List<V2TimMessage> msgList = res.data ?? [];
```

```
// here you can use msgList to render your message list }
```

#### 监听长连接实时获取新消息

历史消息列表初始化后,新消息来自长连接 V2TimAdvancedMsgListener.onRecvNewMessage 。 onRecvNewMessage 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:





TencentImSDKPlugin.v2TIMManager .getMessageManager() .addAdvancedMsgListener(listener: adVancesMsgListener);

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 ,离线推送 ,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 。

## 高阶功能集成

## 使用更多插件丰富 Flutter IM 使用体验

除 SDK 及 TUIKit 本体基础功能外,我们还提供了四个选装插件,帮助您丰富 IM 能力。

- 消息推送插件:支持厂商原生离线推送能力及在线推送能力,并支持推送您的其他业务消息,帮助您提高消息触达率。
- 音视频通话插件:支持类似微信的一对一/群组 音视频 通话。
- 地理位置消息插件:提供选取位置/发送位置及解析展示位置消息的能力。
- 自定义表情插件:TUIKit0.1.5版本后,无自带表情包,需要使用此插件,快速简便集成表情能力。支持 Emoji Unicode 编码及自定义图片表情。集成代码 可参考我们的 Demo。

() 说明

如果您有好的想法及建议,欢迎随时 联系我们。

## 拓展更多平台

腾讯云IM for Flutter 相关SDK默认支持 Android / iOS平台, 如果您需要拓展更多平台( Web/Desktop ),请参考本部分。

## Flutter for Web支持

我们的 SDK, TUIKit(tencent\_cloud\_chat\_uikit) 0.1.5版本,无 UI SDK(tencent\_cloud\_chat\_sdk) 4.1.1+2 版本起,可兼容 Web 端。 相比 Android 和 iOS 端,需要一些额外步骤。如下:

#### 升级 Flutter 3.x 版本

Flutter 3.x 版本 针对 Web 性能做了较多优化,强烈建议您使用其来开发 Flutter Web 项目。

## 引入 Flutter for Web 增补SDK

flutter pub add tencent\_im\_sdk\_plugin\_web

## 引入 JS

```
    说明
    如果您现有的 Flutter 项目不支持 Web,请在项目根目录下运行 flutter create. 添加 Web 支持。
```

进入您项目的 web/ 目录,使用 npm 或 yarn 安装相关JS依赖。初始化项目时,根据屏幕指引,进行即可。

cd	web

npm init

npm i tim-js-sdk

npm i tim-upload-plugin

打开 web/index.html ,在 <head> </head> 间引入这JS文件。如下:

<script src="./node\_modules/tim-upload-plugin/index.js"></script>
<script src="./node modules/tim-is-sdk/tim-is-friendship.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></scri



• • •			
im-flutter-uikit ) web ) 📇 index.html			
g 🚍 Project 👻 😔 호 🗧 ¢	— I.dert 🛪 🚜 conversation_services.dert 🛪 😤 open_file.dert 🗴 🔮 pubspec.lock 🗴 🐔 platform_channel.dert 🗴 📇 index.html 🚿 🛱 suckage.json 🛪 👯 message.		
> Rebuild	26 name="keywords"		
> Hereitos	27 content="Chat SDK, Tencent Cloud Chat, 腾讯云IM, Flutter Chat, Flutter IM, Chat API, 腾讯云即时通信IM"		
> 🖿 IIb			
> macos			
2 package_err			
V Web	31 <meta content="yes" name="apple-mobile-web-app-capable"/>		
A Decide modulos	32 <meta content="Tencent Cloud Chat - 陳讯云IM - Flutter" name="apple-mobile-web-app-title"/>		
	33		
timunload-olugin	34 secrint spect/in-unload-plugin/index.is">		
fal.package-lock.ison	<pre>35 <script src=",/node_nodules/tin-1s-sdk/tim-1s-friendship.js"></pre>/script></pre></td><td></td></tr><tr><td>≥ > bisplash</td><td>36 <script src="https://unpkg.com/yconsole#latest/dist/yconsole.min.is"></script></pre>		
for the second s	37 secient src="./trte-wrapper.is">		
📇 index.html	38 ctitle>Tengent (loud Dat - MitraTM - Fluttens/title>		
fij manifest.json	<pre>39 slink rel="manifest" bref="manifest.ison" /&gt;</pre>		
👸 package.json			
🖏 package-lock.json	4 Stann		
publish	41 Tet- Aton		
n is.js			
tim-js-friendship.js			
tim-upioad-piugin.js	the the indef a contract of the contract of th		
in tric.js			
> web publish			
4 onde vml	47 PeL="SNOPTOUT LCON"		
d flutter-plugins	48 nPeter, /tavicon.ico"		
d .flutter-plugins-dependencies	49 nce_pret="/favicon.ico"		
🛃 .gitignore	50 type="inage/x-icon"		
di .metadata			
🚜 .tox.yml	52 <script src="splash/splash.js"></script>		
e 📇 analysis_options.yaml			
g 🔮 code.text	54 content="width=device-width, initial-scale=1.0, naximun-scale=1.0, user-scalable=no"		
🖇 💑 get_all_code.js			
keystore_tuikit.jks			
n package.jsón			
B) package-lock.json	58 🗅		
S Pubspec.iock			

## Flutter for Desktop(PC) 支持

我们的TUIKit(tencent\_cloud\_chat\_uikit) 自 2.0.0 版本,无 UI SDK(tencent\_cloud\_chat\_sdk) 4.1.9 版本起,可兼容 Desktop 端,包含 Windows / macOS / 桌面 Web。相比移动端,需要一些额外步骤。如下:

## 升级 Flutter 3.x 版本

从 Flutter 3.0 版本起,才可用于打包 desktop 端,因此,如需使用,请升级至 Flutter 3.x 版本。

## 引入 Flutter for Desktop 增补 SDK

flutter pub add tencent\_im\_sdk\_plugin\_desktop

#### macOS修改

打开 macos/Runner/DebugProfile.entitlements 文件。 在 <dict></dict> 中,加入如下 key-value 键值对。

```
<key>com.apple.security.app-sandbox</key>
<false/>
```

## 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

## iOS 端 Pods 依赖无法安装成功

尝试方案一: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



🗯 Xcode File Edit View Find	Navigate Editor	Product Debug Source Control V	Vindow Help	ଟ ଜ
	P Runner feat/push_support	Run %R	3 Pro	Preparing the watch for develop
	88   < > rant	Test %U Profile %I	list (zh-Hant) 🛛 🎬 In	foPlist (zh-Hans) 🛛 🖾 Assets 🔰 🗲
V \Lambda Runner	🔼 Runner	Analyze 🗘 💥 B		
> 🔚 Flutter		Archive	ng & Capabilities Resou	rce Tags Info Build Settings Build F
V Runner	PROJECT	Build For > Perform Action >		
> X Main	Kunner 🗠	Build + B	Display Name	Tencent IM
<ul> <li>X LaunchScreen</li> </ul>	TARGETS	Clean Build Folder 수 #K	Bundle Identifier	com.tencent.im.flutter.uikit
🔀 LaunchScreen (Base)	Runner	Clean Test Results へて第K	Version	1.0.0
뜰 LaunchScre, Traditional)	- Kunner	Stop %.	Build	1
LaunchScree, Simplified)		Build Documentation ^ 🕁 🕸 D		
Info		Show Build Folder in Finder		
InfoPlist (English)		Export Localizations	iOS 12.0 0	✓ iPhone
InfoPlist (Ch, Traditional)		Import Localizations		iPad
InfoPlist (Che, Simplified)				
h GeneratedPluginRegistrant		Scheme >	Main Interface	Main
M GeneratedPluginRegistrant		Destination >	Device Orientation	Z Portrait
AppDelegate		iest Fidil		Upside Down
h Runner-Bridging-Header		Xcode Cloud >		🗹 Landscape Left
> Products				✓ Landscape Right

## 尝试方案二: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



## 佩戴 Apple Watch 时,真机调试 iOS 报错



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。

## 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。

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



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



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





## 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: https://www.nuget.org/downloads
- 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 。

#### 错误码如何查询?

- IM SDK 的 API 层面错误码,请查看 该文档。
- TUIKit 的场景码,用于界面弹窗提示,通过 onTUIKitCallbackListener 监听 获得。全部场景码清单,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。



# 快速入门(用 Flutter 快速集成至您现有原生应用)

最近更新时间: 2024-03-26 15:29:41

通过阅读本文,您可以了解在您现有的 Android / iOS 原生开发项目中,集成腾讯云 IM Flutter 的方法。

有的时候,使用 Flutter 重写您现有的应用程序是不现实的。如果您想在现有 App 中,使用腾讯云 IM 的能力,推荐采用混合开发方案,即将 Flutter 模块,嵌入 您的原生开发 App 项目中。

#### 可在很大程度上,降低您的工作量,快速在双端原生 App 中,植入 IM 通信能力。



## 环境要求

环境	版本	
Flutter	SDK 最低要求 Flutter 2.2.0版本,TUIKit 集成组件库最低要求 Flutter 2.10.0 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。	
腾讯云IM SDK	tencent_cloud_chat_sdk 5.0 及以上版本, tencent_cloud_chat_uikit 1.0 及以上版本。	

# 快速了解

```
观看视频
```

```
① 说明
对于以上的 Demo 项目,源代码可在我们的 GitHub 仓库 中找到,欢迎查阅。
```

## 前置知识点

开始之前,您需要了解腾讯云 IM Flutter SDK 及 TUIKit 的用法,及 Flutter-原生混合开发原理。

## 前序工作

- 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 。
- 3. 在 IM 控制台 选择您的应用,在左侧导航栏依次点击 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。



🔗 腾讯云 🔍	ち 云产品 > 🛛 网站留案 🕇
即时通信 IM	← UserSig生成&校验 -
王 基本配置	
日 功能配置	签名(UserSig)生成工具 · · · · · · · · · · · · · · · · · · ·
晶 群组管理	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。
③ 回调配置	请输入用户名 (UserID)
🐨 安全打击	user1
数据监控器	密钥 (Key)
@ 辅助工具	Contract and Contract of
• 离线推送自查	
・ UserSig生成&校	生成签名 (UserSig)
	当前生成的签名(UserSig)是:
	and an a contract the second
	复制签名 (UserSig)

#### () 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 。

## 腾讯云 IM

## 总体入门

在开始前,您首先需要了解腾讯云 IM Flutter 的 SDK 构成及使用方式。 主要包括两个 SDK: 无 UI 版本 及含UI组件库。本文将以 含UI组件库(TUIKit) 为例,介绍混合开发方案。 关于腾讯云 IM Flutter 详细用法,可从我们的 快速入门文档 看起。

#### 两个模块

腾讯云 IM 主要有两个部分,包括 Chat 聊天模块 和 Call 通话模块。

- Chat 聊天模块主要包括消息收发、会话管理、用户关系管理等。
- Call 通话模块主要包括音视频通话,包括一对一通话和群组多人通话。

## Flutter 混合开发

核心原理是,将 module 形式的 Flutter 项目,打包成 Native 端的可执行程序,嵌入 Native 项目中。因 Flutter module 可以通用,因此仅需编写一次 Flutter module,即可嵌入 Android/iOS APP 中。

当您现有应用需要展示腾讯云IM相关页面时,可加载对应用于承载 Flutter 的 Activity(Android)或 ViewController(iOS)。 当需要两端通信时,如传递当前用户信息,传递音视频通话数据,触发离线推送数据,可采用 Method Channel 方式进行。触发另一端的方法使用 invokeMethod ,监听另一端发来的方法调用使用 预挂载的 Method Channel 监听器 。

### 将 Flutter 模块添加至 Android 项目中

#### 详细学习

将 Flutter module 添加为 Gradle 中现有应用程序的依赖项。有两种方式可以实现这一点。

## Android方式一: 依赖 Android Archive (AAR)

AAR 机制创建通用的 Android AAR 作为打包 Flutter module 的中介。如果您经常构建,它会增加一个构建步骤。 该选项将 Flutter 库打包为由 AAR 和 POMS 构件组成的通用本地 Maven 存储库。此选项允许您的团队在不安装 Flutter SDK 的情况下构建主机应用程序。然 后,您可以从本地或远程存储库中分发构件。 因此,建议在线上生产环境,使用本方案。 **具体步骤:** 



## 1. 在您的Flutter module中,运行:

#### flutter build aar

2. 然后,按照屏幕上的说明进行集成。

Running "flutter pub get" in tencent_chat_module	4.25			
<pre>&gt; tencent_chat_module git:(main) × flutter build aar</pre>				
💪 Building with sound null safety 🦾				
Running Gradle task 'assembleAarDebug'				
Built build/host/outputs/repo.				
Running Gradle task 'assembleAarProfile'				
Built build/host/outputs/repo.				
Running Gradle task 'assembleAarRelease'				
✓ Built build/host/outputs/repo.				
Consuming the Module				
<ol> <li>Open <host>/app/build.gradle</host></li> </ol>				
<ol><li>Ensure you have the repositories configured, otherwise</li></ol>	add them:			
String storageUrl = System.env.FLUTTER_STORAGE_BASE_U	L ?: " <u>https://storage.googleapis.com</u> "			
repositories {				
maven {				
url '/Users/wangrunlin/Documents/GitHub/tencent	hat-add-flutter-to-app/Multiple Flutter Engines/tencent_chat_module/build/host/outputs/repo'			
}				
maven {				
url "\$storageUrl/download.flutter.io"				
}				
}				
<ol><li>Make the host app depend on the Flutter module:</li></ol>				
dependencies {				
debuginplementation 'com.tencent.cnat.flutter.module:	LUTTEP:1.8:debug			
profileImplementation 'com.tencent.chat.flutter.modul	:flutter:1.0:profile'			
releaseImplementation 'com.tencent.cnat.flutter.module	tflutter:1.0:release'			
}				
6 Add the `specile` build tune:				
4. Add the profite boitd type.				
android f				
buildTypes f				
anofile 4				
}				
To learn more, visit https://flutter.dev/go/build-aar				
A tencent chat module nit:(nein) x				

3. 您的应用程序现在将 Flutter 模块作为依赖项包括在内。

#### Android 方式二: 依赖Flutter module源代码

源代码子项目机制是一个方便的一键构建过程,但需要 Flutter SDK。这是 Android Studio IDE 插件使用的机制。

此方式可为您的 Android 项目和 Flutter 项目实现一步构建。当您同时处理两个部分并快速迭代时,此选项很方便,但您的团队必须安装 Flutter SDK 才能构建 应用程序。

因此,建议在开发测试环境,使用本方案。

#### 具体步骤:

1. 将 Flutter module 作为一个子项目,添加至宿主 APP 的 settings.gradle 中:



2. 在您应用中的 app/build.gradle => dependencies 中引入对Flutter module的 implementation :



3. 您的应用程序现在将 Flutter 模块作为依赖项包括在内。

#### 将 Flutter 模块添加至 iOS 项目中

#### 详细学习

有两种方法可以在现有应用程序中嵌入 Flutter。


#### iOS方式一:嵌入 CocoaPods 和 Flutter SDK 集成

使用 CocoaPods 依赖项管理器并安装 Flutter SDK。这种方法要求每个从事项目工作的开发人员都有一个本地安装的 Flutter SDK 版本。 只需在 Xcode 中构建您的应用程序,即可自动运行脚本来嵌入您的 DART 和插件代码。这允许快速迭代最新版本的颤振模块,而无需在 Xcode 之外运行其他命 令。

因此,建议在开发测试环境,使用本方案。

具体步骤:

1. 将以下代码添加到 Podfile 中:

// 上一步构建的Flutter Module的路径 flutter\_chat\_application\_path = '../tencent\_chat\_module'

load File.join(flutter\_chat\_application\_path, '.ios', 'Flutter', 'podhelper.rb')

2. 对于每个需要嵌入 Flutter 的 Podfile target,调用 install\_all\_flutter\_pods(flutter\_chat\_application\_path)。



3. 在 Podfile 的 post\_install 块中,调用 flutter\_post\_install(installer),并完成 腾讯云IM TUIKit 所需的权限声明,包括麦克风权限/相机权限/相册权 限。



执行 pod install 。

#### () 说明

- 在 tencent\_chat\_module/pubspec.yaml 中更改Flutter插件依赖时,请在Flutter Module目录中运行 flutter pub get 以刷新 podhelper.rb 脚本读取的插件列表。然后,从您iOS应用程序的根目录,再次执行 pod install 。
- 对于 Apple Silicon 芯片 arm64 架构的 Mac电脑,可能需要执行 arch -x86\_64 pod install --repo-update 。

podhelper.rb 脚本将您的插件 / Flutter.framework / App.framework 植入您的项目中。

#### iOS 方式二:在 Xcode 中嵌入 frameworks

为 Flutter 引擎、已编译的 DART 代码和所有 Flutter 插件创建框架。手动嵌入框架,并在 Xcode 中更新现有应用程序的构建设置。 通过手动编辑现有的 Xcode 项目,您可以生成必要的 framework 并将它们嵌入到应用程序中。如果您的团队成员无法在本地安装 Flutter SDK 和 CocoaPods,或者如果您不想在现有应用程序中使用 CocoaPods 作为依赖项管理器,则可以这样做。每次您在您的颤动模块中修改代码时,您都必须运行

flutter build ios-framework .

因此,建议在线上环境,使用本方案。

### 具体步骤:

1. 在您的Flutter module中,运行如下代码。下面的示例,假设您想要将framework生成到 some/path/MyApp/Flutter/。

flutter build ios-framework --output=some/path/MyApp/Flutter/



2. 在 Xcode 中将生成的 frameworks 集成到您的既有应用中。例如,您可以在 some/path/MyApp/Flutter/Release/ 目录拖拽 frameworks 到您的应用 target 编译设置的 General > Frameworks, Libraries, and Embedded Content 下,然后在 Embed 下拉列表中选择 "Embed & Sign"。

# 混合开发选型

我们推荐您使用 Flutter Module 方式进行混合开发集成。

在 Native 原生项目中,构建 Flutter 引擎,来承载 Flutter 中的 Chat 及 Call 模块。有关两个模块的介绍,请看此处。 对于 Flutter 引擎的创建管理,目前两种方式:单 Flutter 引擎及多 Flutter 引擎。

引擎 模式	介绍	优点	缺点	Demo 源码 下载
Flutt er单 引擎	Chat 模块和 Call 模块在同一个 Flutter 引擎中承载。	方便,所有 Flutter 代码统一维护。	由于 Call 插件,在有电话呼入时,需要自动 展示来电页面。如果在同一个引擎中,需要 强制跳转至 Flutter 所在页面,体验较差。	点击下载
Flutt er多 引擎	Chat 模块和 Call 模块分别承载 于不同的 Flutter 引擎中,使用 Flutter 引擎组来统一管理这两个 引擎。	Call 插件独立存在于一个 Flutter 引擎中,独立页面控制,来电时,直 接将该页面弹窗即可,不影响用户当 前所在页面,体验较好。	通话模块无法最小化成浮窗形式。	点击下载

此外,我们还提供,将腾讯云 IM Native SDK 与 Flutter SDK 结合使用的方案,适用场景和步骤介绍可查看这里 。Demo 源码下载 。

# 方案一: Flutter 多引擎方案(推荐)

本方案中,Chat 和 Call 模块分别独立于不同的Flutter引擎。

使用多个 Flutter 引擎的优点是,每个实例都是独立的,并维护其自己的内部导航堆栈、UI和应用程序状态。这简化了整个应用程序代码的状态保持责任,并提高 了模块化能力。



在 Android 和 iOS 上添加多个 Flutter 引擎,主要基于一个 FlutterEngineGroup 类(Android API、iOS API)来构造并管理多个 FlutterEngine (Flutter 引擎)。

在我们的项目中,我们基于一个统一的 FlutterEngineGroup,来管理两个 FlutterEngine(Flutter 引擎),分别用于承载 Chat 和 Calling 模块。



由于代码较多,本文无法全部列举,全量详细代码 (Native + Flutter),请下载查阅

# Flutter Module 开发

要将 Flutter 嵌入到现有应用程序中,请首先创建一个 Flutter 模块。 在您项目的根目录外层,运行。



# cd some/path/

flutter create --template module tencent\_chat\_module

这会在 some/path/tencent\_chat\_module/ 创建一个 Flutter 模块项目。 在该目录中,您可以运行与在任何其他 Flutter 项目中相同的 Flutter 命令,例如 flutter run --debug 或 flutter build ios 。您还可以使用 Flutter 和 Dart 插件在 Android Studio, IntelliJ 或 VS Code 中运行该模块。该项目在嵌入到 现有应用程序之前包含模块的单视图示例版本,这对于测试代码的仅 Flutter 部分很有用。

tencent\_chat\_module 模块目录结构类似于普通的 Flutter 应用程序:



现在,我们可以在 lib/ 中,编写代码了。

### 梳理 Flutter lib 目录

① 说明 以下代码结构,仅供参考,您可根据需要灵活组织,以引入腾讯云 IM Flutter。

在 lib/ 我们创建三个目录, call, chat, common 。分别用于放置通话引擎, IM引擎, 及通用model类。

tencent\_chat\_module/ |--- lib/ | ---- call/ | ---- chat/ | ---- common/

#### 通用 model 类模块

新建 common/common\_model.dart <mark>文件,如下所示,新建两个class,用于定义Flutter与原生应用</mark>通信规范。

```
class ChatInfo {
 String? sdkappid;
 String? userSig;
 String? userID;
 ChatInfo.fromJSON(Map<String, dynamic> json) {
  sdkappid = json["sdkappid"].toString();
  userSig = json["userSig"].toString();
  userID = json["userID"].toString();
 Map<String, String> toMap(){
  final Map<String, String> map = {};
  if(sdkappid != null){
   map["sdkappid"] = sdkappid!;
  3
  if(userSig != null){
   map["userSig"] = userSig!;
  if(userID != null){
   map["userID"] = userID!;
  return map;
 3
```



#### }

```
class CallInfo{
String? userID;
String? groupID;
```

CallInfo();

```
CallInfo.fromJSON(Map<String, dynamic> json) {
  groupID = json["groupID"].toString();
  userID = json["userID"].toString();
}
Map<String, String> toMap(){
  final Map<String, String> map = {};
  if(userID != null){
    map["userID"] = userID!;
    }
  if(groupID != null){
    map["groupID"] = groupID!;
    }
  return map;
}
```

### **Chat 模块**

#### 首先编写 IM 引擎。本模块所有代码及文件,均在 lib/chat 目录下。

1. 新建全局状态管理 Model, 名为 model.dart 。

#### () 说明

该 Model 用于挂载初始化并管理腾讯云 IM Flutter 模块,离线推送能力,全局状态管理,维护与 Native 间通信。是整个 Chat 模块的核心。

详细代码可查看 Demo 源码。重点关注三个部分:

- Future<dynamic> \_handleMessage(MethodCall call): 动态监听 Native 透传来的事件,包括登录信息及点击推送事件。
- Future < void > handleClickNotification(Map < String, dynamic > msg): 点击通知处理事件,来自Native透传,从Map中取出数据,跳转至对应的子模块,如某个具体会话。
- Future < void > initChat(): 初始化腾讯云IM/登录腾讯云 IM/并完成离线推送的初始化及Token上报。该方法使用线程锁机制,保证同时只能执行一个,并在初始化成功后,不重复执行。

```
    说明
请根据 离线推送接入指引,完成厂商离线推送功能接入,才可正常上报推送 Token,使用推送功能。
```

- 2. 新建 chat\_main.dart 文件,用于 Chat 模块主入口。
  - 该页面也是 Flutter Chat 模块的首页。
  - 在 Demo 中,该页面在未登录前为加载状态,登录后展示会话列表。
  - 此外,还需要在这里,完成 didChangeAppLifecycleState 监听与前后台切换事件上报。
  - 详细代码可查看 Demo 源码。
- 3. 新建 push.dart 文件,用于单例管理 离线推送插件 能力。用于获取并上报Token/获取推送权限等操作。详细代码可查看 Demo 源码。
- 4. 新建 conversation.dart 文件,用于承载 TUIKit 的会话模块组件 TIMUIKitConversation 。详细代码可查看 Demo 源码。
- 5. 新建 chat.dart 文件,用于承载 TUIKit 的历史消息列表和发送消息模块组件 TIMUIKitChat 。 该页面还有跳转至 Profile 及 Group Profile 页面的能力。 详细代码可查看 Demo 源码。
- 6. 新建 user\_profile.dart 文件,用于承载 TUIKit 的用户信息及关系链管理模块组件 TIMUIKitProfile 。详细代码可查看 Demo 源码。
- 7. 新建 group\_profile.dart 文件,用于承载 TUIKit 的群信息及群管理模块组件 TIMUIKitGroupProfile 。详细代码可查看Demo源码。 此时,Chat 模块已开发完成。最终结构如下:



tencent_chat_module/
lib/
call/
│ └── chat.dart
model.dart
└── chat_main.dart
push.dart
conversation.dart
user_profile.dart
group_profile.dart
│ └── chat/
common/

### Call 模块

该模块用于承载音视频通话能力,该能力由 音视频通话插件 提供。

该模块的核心是,监听收到新的通话邀请时,通过调用Native方法,自动弹出通话页面;并接受 Chat 模块经由Native转发来的通话请求,主动发起通话。 **首先编写 IM 引擎。本模块所有代码及文件,均在 lib/call 目录下。** 

1. 新建全局状态管理 Model, 名为 model.dart 。

该 Model 用于挂载初始化并管理 音视频通话插件,全局状态管理,维护与Native间通信。

是整个Call模块的核心。

详细代码可查看Demo源码。重点关注两个部分:

- \_onRtcListener = TUICallingListener(...): 定义了通话事件的监听器,通过 Method Channel 通知 Native 层,动态控制 Call 模块所属的 ViewController(iOS)/Activity(Android) 的前端展示与否。
- Future<dynamic> \_handleMessage(MethodCall call) : 动态监听 Native 透传来的主动发起通话请求,来自 Call 模块的调用,主动发起通话。

2. 新建 call\_main.dart 文件,用于 Call 模块主入口。该组件用于注入 音视频通话插件所需绑定的navigatorKey。详细代码可查看 Demo 源码。

#### 配置各个 Flutter 引擎的入口

开发完上述三个模块后,现在可完成最终对外暴露的 main 方法,作为 Flutter 引擎的入口。

 1. 默认入口。打开 lib/main.dart 文件,将 main() 方法改成一个空 MaterialApp 即可。该方法作为 Flutter Module 的默认入口,在Flutter多引擎,使 用 FlutterEngineGroup 管理的背景下,如果没有子 Flutter Engine 不设置任何 entry point,这个方法就不会被用到。例如,在我们的场景中,这个默 认 main() 方法就没有被用上。

oid main() { WidgetsFlutterBinding.ensureInitialized();
runApp(MaterialApp(
title: 'Flutter Demo',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: Container(),

2. 配置 Chat 模块的入口。使用 @pragma('vm:entry-point') 注解,将该方法标记为一个 entry-point 入口。方法名 chatMain 即该入口的名称,在 Native 中,也使用该名称,创建对应 Flutter 引擎。使用全局 ChangeNotifierProvider 状态管理,维护 ChatInfoModel 数据及业务逻辑。







3. 配置 Call 模块的入口。同理,该入口命名为 callMain 。使用全局 ChangeNotifierProvider 状态管理,维护 CallInfoModel 数据及业务逻辑。



至此,Flutter Module 部分,Dart 代码编写完成。接下来,开始编写 Native 代码。

# iOS Native 开发

本文以 Swift 语言为例。

 说明 以下代码结构,仅供参考,您可根据需要灵活组织。

进入您的 iOS 项目目录。如果您现有的应用程序,假设叫做 MyApp , 还没有 Podfile,请按照 CocoaPods 入门指南 将 Podfile 添加到项目中。

### 引入 Flutter Module

请参考 此部分,将 Flutter module 引入您的原生应用程序中。建议采用方式一。

# 在 iOS 项目中,管理 Flutter 引擎



在 AppDelegate.swift 文件中,添加如下代码:



# @UIApplicationMain

lazy var flutterEngines = FlutterEngineGroup(name: "chat.flutter.tencent", project: nil)

]

#### 创建一个用于管理Flutter引擎的单例对象。

这个 Swift 单例对象,用于集中管理 Flutter 实例,并方便在项目中各处,直接调用。

Demo 代码的逻辑是,使用新的路由,承载 Chat 的 ViewController; Call 的 ViewController,通过 present 和 dismiss 动态弹窗维护。 新建 FlutterUtils.swift 文件,编写代码。本部分详细代码,可查看 Demo 源码。

重点关注:

- private override init():初始化各 Flutter 引擎实例,注册 Method Channel,监听事件。
- func reportChatInfo(): 将用户登录信息和 SDKAPPID 透传至 Flutter Module,使 Flutter 层得以初始化并登录腾讯云IM。
- func launchCallFunc(): 用于拉起 Call 的 Flutter 页面,可被 Call 模块收到通话邀请触发,也可被 Chat 模块主动发起通话触发。
- func triggerNotification(msg: String): 将 iOS Native 层收到的离线推送消息点击事件,及其包含的 ext 信息,以 JSON String形式,透传至 Flutter 层绑定的监听处理事件。用于处理离线推送点击跳转,例如至对应会话。

#### 监听及转发离线推送点击事件

离线推送的初始化/Token上报/点击事件对应的会话跳转处理,已在 Flutter Chat 模块中进行,因此,Native 区域,仅需透传点击通知事件的 ext 即可。 之所以这么做,是因为点击通知事件已在 Native 被拦截消费,Flutter 层无法直接拿到,必须经由 Native 转发。 在 AppDelegate.swift 文件中,新增如下代码。具体代码,可以参考 Demo 源码。

	20	
	21	GUIApplicationMain
	22	class AppDelegate: FlutterAppDelegate {
	23	lazy var flutterEngines = FlutterEngineGroup(name: "chat.flutter.tencent", project: nil)
	24	
	26	override func application ( application; UTApplication, didEinishlaunchingWithOptions; aunchOptions; (UTApplication; application; Appli
	26	
	22	if Havialahio(ins 10 a +) (
	29	INtransition for the current() delegate = relf
		biosenformenter content () for egete = ser
1		1
	30	
		if ist remotemetrication = launchopitons/tolapplication.Launchopitonskey.remotemetrication.t
	32	let notificationext: [AnyHashable:Any] = femoteMotification as: [AnyHashable:Any]
	33	let remoteMotificationString: String = notificationExt.jsonStringKepresentation ?? "()"
	34	FlutterUtils.shared.triggerNotification(msg: remoteNotificationString)
	35	3
	• 36	
L		return super.application(application, didFinishLaunchingWithOptions: launchOptions);
	38	}
1	39	
	40	override func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response: UNNotificationResponse, withCompletionHandler completionHandler:
	_	<pre>@escaping () → Void) {</pre>
	- 41	<pre>let notificationExt: Dictionary = response.notification.request.content.userInfo</pre>
	42	<pre>let notificationExtString: String = notificationExt.jsonStringRepresentation ?? "()"</pre>
	- 43	FlutterUtils.shared.triggerNotification(msg: notificationExtString)
	- 44	3
1	45	
1	46	3
18		

此时,iOS Native 层编写完成。

### Android Native 开发

本文以 Kotlin 语言为例。

说明
 以下代码结构,仅供参考,您可根据需要灵活组织。

#### 引入 Flutter Module

请参考 此部分,将 Flutter module 引入您的原生应用程序中。建议采用方式二。

### 在 Android 项目中,管理Flutter引擎

#### 创建一个用于管理Flutter引擎的单例对象。

这个 Kotlin 单例对象,用于集中管理 Flutter 实例,并方便在项目中各处,直接调用。 新建 FlutterUtils.kt 文件,并定义 FlutterUtils 静态类。

```
@SuppressLint("StaticFieldLeak")
object FlutterUtils { }
```

### 创建 FlutterEngineGroup (Flutter 引擎组),统一管理多个引擎实例。

在 FlutterUtils.kt 文件中,定义一个 FlutterEngineGroup ,及配套各个 Flutter Engine 实例和 Method Channel,并在初始化时,将其初始化。

lateinit var context : Context lateinit var flutterEngines: FlutterEngineGroup private lateinit var chatFlutterEngine:FlutterEngine



### private lateinit var callFlutterEngine:FlutterEngine

lateinit var chatMethodChannel: MethodChannel lateinit var callMethodChannel: MethodChannel

// 初始化 flutterEngines = FlutterEngineGroup(context)

#### 继续完成该用于管理Flutter引擎的单例对象。

Demo 代码的逻辑是,使用新的路由,承载 Cha t和 Call 的 Activity。

Chat 的 Activity,由用户主动进入及退出,Call 的 Activity,由监听器或主动外呼,自动导航进及返回出。

重点关注:

- fun init():初始化各 Flutter 引擎实例,注册 Method Channel,监听事件。
- fun reportChatInfo():将用户登录信息和 SDKAPPID 透传至 Flutter Module,使 Flutter 层得以初始化并登录腾讯云IM。
- fun launchCallFunc(): 用于拉起 Call 的 Flutter 页面,可被 Call 模块收到通话邀请触发,也可被 Chat 模块主动发起通话触发。
- fun triggerNotification(msg: String):将 iOS Native 层收到的离线推送消息点击事件,及其包含的ext信息,以 JSON String形式,透传至 Flutter 层 绑定的监听处理事件。用于处理离线推送点击跳转,例如至对应会话。

本单例 object 的详细代码,可以参考 Demo 源码。

#### 在总入口 MyApplication 中,初始化上述对象。

在 MyApplication.kt 文件中,将全局context传入单例对象,并执行初始化。

```
class MyApplication : MultiDexApplication() {
    override fun onCreate() {
        super.onCreate()
        FlutterUtils.context = this // new
        FlutterUtils.init() // new
    }
}
```

#### 监听及转发离线推送点击事件

离线推送的初始化/Token上报/点击事件对应的会话跳转处理,已在 Flutter Chat 模块中进行,因此,Native 区域,仅需透传点击通知事件的ext即可。 之所以这么做,是因为点击通知事件已在 Native 被拦截消费,Flutter 层无法直接拿到,必须经由 Native 转发。

### 🕛 说明

由于不同厂商的离线推送接入步骤不一致,本文以 OPPO 为例,全部厂商接入方案,可查看本文档.

在腾讯云 IM 控制台中,新增 OPPO 的推送证书, 点击后续动作 选择 打开应用内指定页面 , 应用内页面 以 Activity 方式,配置一个用于处理离线推送信息的页面,建议为应用首页。如,我们的 Demo 配置为: com.tencent.chat.android.MainActivity .

<b>UPPU</b> (1D: 260	554)	加小中本	印刷书具
AppKey	4c2a7b09bb8b404		
AppID			
Master			
ChannellD	new_message		
点击后续动作	打开应用内指定页面		
应用内页面	activity: com.tencent.chat.android.MainActivity		

在上方控制台配置的用于离线推送的 Activity 文件中,新增如下代码。

该代码的作用是,当厂商拉起相应 Activity 时,从 Bundle 中取出 HashMap 形式 ext 信息,触发单例对象中的方法,将这个信息,手动转发至 Flutter 中。具 体代码,可以参考 Demo 源码。





此时,Android Native 层编写完成。

# 方案二: Flutter 单引擎方案

本方案,将 Chat 模块和 Call 模块,写在同一个 Flutter 引擎实例中。

腾讯云IM [Flutter - 原生] 混合开发模式 - <b>单弓</b>	擎
Swift Dbjective-C 双端 Native 原生开发APP 包含隙IIID及通话功能外,您自己的业务模块	Java Kotlin
道用 Flutter 模块 FlutterEngine 全局状态Model 時元町Chart 進株地法 Provider <u>8.dart 子页面</u> conversation.dart(TUIKiKtonver	rsation)
Calling 音視算過话 method Channel 開理 chat.dart (TUIKitChat) user_profile.dart (TUIKitPro	file)

这两个模块只能同时出现同时隐藏,仅需维护一个 Flutter 引擎即可。



# Flutter Module 开发

要將 Flutter 嵌入到现有应用程序中,请首先创建一个 Flutter 模块。 在您项目的根目录外层,运行。

cd some/path/ flutter create --template module tencent\_chat\_module

这会在 some/path/tencent\_chat\_module/ 创建一个 Flutter 模块项目。 在该目录中,您可以运行与在任何其他 Flutter 项目中相同的 Flutter 命令,例如 flutter run --debug 或 flutter build ios 。您还可以使用 Flutter 和 Dart 插件在 Android Studio, IntelliJ 或 VS Code 中运行该模块。 该项目在嵌入到 现有应用程序之前包含模块的单视图示例版本,这对于测试代码的仅 Flutter 部分很有用。



tencent\_chat\_module 模块目录结构类似于普通的 Flutter 应用程序:



现在,我们可以在 lib/ 中,编写代码了。

#### main.dart

修改 main.dart 文件,引入 TUIKit,离线推送插件 及 音视频通话插件 。

全局状态,我们的 IM SDK 及 Method Channel 与 Native 通信状态,管理于 ChatInfoModel 中。

接收到 Native 传来的用户信息及 SDKAPPID 后,调用 \_coreInstance.init() 及 \_coreInstance.login() 初始化并登录腾讯云IM。并初始化音视频推送插件 及离线推送插件,完成推送 Token 上报。

```
① 说明
请根据 离线推送接入指引,完成厂商离线推送功能接入,才可正常上报推送 Token,使用推送功能。
```

- 对于音视频通话插件,需要关注:监听收到新的通话邀请时,通过调用 Native 方法,让 Native 检测用户当前是否在本 Flutter 模块页面,如果不在,需要强制将前端页面调整至本模块,以展示来电页面。
- 对于离线推送插件,需要关注:点击通知处理事件,来自 Native 透传,从 Map 中取出数据,跳转至对应的子模块,如某个具体会话。

完成首页的制作,在未登录时展示加载动画;登录成功后,展示会话列表页面。

此外,还需要在这里,完成 didChangeAppLifecycleState 监听与前后台切换事件上报。

详细代码可查看 Demo 源码。

#### 其他 TUIKit 模块引入

- 1. 新建 push.dart 文件,用于单例管理 离线推送插件 能力。用于获取并上报Token/获取推送权限等操作。详细代码可查看 Demo 源码。
- 2. 新建 conversation.dart 文件,用于承载 TUIKit 的会话模块组件 TIMUIKitConversation 。详细代码可查看 Demo 源码。
- 3. 新建 chat.dart 文件,用于承载 TUIKit 的历史消息列表和发送消息模块组件 TIMUIKitChat 。该页面还有跳转至 Profile 及 Group Profile 页面的能力。 详细代码可查看Demo源码。
- 4. 新建 user\_profile.dart 文件,用于承载 TUIKit 的用户信息及关系链管理模块组件 TIMUIKitProfile 。详细代码可查看 Demo 源码。
- 5. 新建 group\_profile.dart 文件,用于承载 TUIKit 的群信息及群管理模块组件 TIMUIKitGroupProfile 。详细代码可查看 Demo 源码。
- 至此,统一的 Flutter Module 开发完成。

#### iOS Native 开发

本文以 Swift 语言为例。

```
() 说明
```

以下代码结构,仅供参考,您可根据需要灵活组织。

#### 进入您的 iOS 项目目录。

如果您现有的应用程序,假设叫做 MyApp , 还没有 Podfile,请按照 CocoaPods入门指南 将 Podfile 添加到项目中。

#### 引入 Flutter Module

请参考 此部分,将 Flutter module 引入您的原生应用程序中。建议采用方式一。

#### 在 iOS 项目中,管理 Flutter 引擎

创建一个 FlutterEngine。

创建 FlutterEngine 的适当位置特定于您的主应用程序入口。作为一个例子,我们演示了如何在 AppDelegate 中的app启动时创建一个 FlutterEngine,并 公开为一个属性。



# import UIKit

import FlutterPluginRegistrant

#### @UIApplicationMain

```
class AppDelegate: FlutterAppDelegate { // More on the FlutterAppDelegate.
lazy var flutterEngine = FlutterEngine(name: "tencent cloud chat")
```

override func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
 // Runs the default Dart entrypoint with a default Flutter route.
 flutterEngine.run();
 GeneratedPluginRegistrant.register(with: self.flutterEngine);
 return super.application(application, didFinishLaunchingWithOptions: launchOptions);
}

#### 创建一个用于管理 Flutter 引擎的单例对象。

这个 Swift 单例对象,用于集中管理 Flutter Method Channel,并提供一系列与 Flutter Module 通信的方法,方便在项目中各处,直接调用。 这些方法包括:

- private override init():初始化 Method Channel,并为其绑定事件监听方法。
- func reportChatInfo():将用户登录信息和 SDKAPPID 透传至 Flutter Module,使 Flutter 层得以初始化并登录腾讯云IM。
- func launchChatFunc(): 拉起或导航至 Flutter Module 所在 ViewController。
- func triggerNotification(msg: String): 将 iOS Native 层收到的离线推送消息点击事件,及其包含的ext信息,以 JSON String形式,透传至 Flutter 层绑定的监听处理事件。用于处理离线推送点击跳转,例如至对应会话。

#### 详细代码可查看 Demo 源码。

#### 监听及转发离线推送点击事件

离线推送的初始化/Token上报/点击事件对应的会话跳转处理,已在 Flutter Chat 模块中进行,因此,Native 区域,仅需透传点击通知事件的ext即可。 之所以这么做,是因为点击通知事件已在 Native 被拦截消费,Flutter 层无法直接拿到,必须经由 Native 转发。

#### 在 AppDelegate.swift 文件中,新增如下代码。具体代码,可以参考 Demo 源码。

	QUIAplicationkin class AppOklapits: FlutterAppOklapit ( // Hore on the FlutterAppOklapite. lasy var flutterEngine = FlutterEngine(name: 'lo.flutter')	
	everide func spplication(_application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Avy]?) → Bool {	
27	17 #available(105 16.0, *) (	
	UNUserNotificationCenter.current().celegate = seif asy UNUserNotificationCenterDelegate 🛆 Conditional cast from "AppleMegate" to any UNUserNotificationCenterDelegate = aways s	
	if let remoteNotification = launchOstions?[UIApplication.LaunchOptionsKev.remoteNotification]{	
<b>32</b> 33	<pre>let notificationExt: (AnyHashable:Any) = remoteNotification as! (AnyHashable:Any) let remoteNotificationString: String = notificationExt.jsonStringRepresentation ?? "{}"</pre>	
	FlutterUtils.shared.triggerNotification(nsg: remoteNotificationString)	
	// Runs the default Dart entryppint with a default Flutter route.	
	// Used to connect plugins (only if you have plugins with iOS platform code).	
40	GeneratedPluginRegistrant.register(with: self.flutterEngine);	
	return super.application(application, didFinishLaunchingWithOptions: launchOptions);	
42		
43	override func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response: UNNotificationResponse, withCompletionHandler completionHandler: @escaping () → Void) {	
	let notificationExt: Dictionary = response.notification.request.content.userInfo	
	let notificationExtString: String = notificationExt.jsonStringRepresentaiton ?? "{}"	
	FlutterUtils.shared.triggerNotification(msg: notificationExtString)	
	,	

此时,iOS Native 层编写完成。

# Android Native 开发

本文以 Kotlin 语言为例。

```
    说明
以下代码结构,仅供参考,您可根据需要灵活组织。
```

### 引入 Flutter Module

请参考 此部分,将 Flutter module 引入您的原生应用程序中。建议采用方式二。

#### 在 Android 项目中,管理 Flutter 引擎

#### 创建一个用于管理 Flutter 引擎的单例对象。

这个 Kotlin 单例对象,用于集中管理 Flutter Method Channel,并提供一系列与 Flutter Module 通信的方法,方便在项目中各处,直接调用。 新建 FlutterUtils.kt 文件,并定义 FlutterUtils 静态类。



# @SuppressLint("StaticFieldLeak")

object FlutterUtils { }

#### 创建一个 FlutterEngine (Flutter 引擎)。

在 FlutterUtils.kt 文件中,定义一个 FlutterEngine ,并在初始化时,将其初始化。

lateinit var context : Context private lateinit var flutterEngine:FlutterEngine

// 初始化 flutterEngine = FlutterEngine(context)

### 继续完成该用于管理 Flutter 引擎的单例对象。

Demo 代码的逻辑是,使用新的路由,承载 Chat 和 Call 的 Activity。

Chat 的 Activity, 由用户主动进入及退出; Call 的 Activity, 由监听器或主动外呼, 自动导航进及返回出。

#### 重点关注:

- fun init():初始化 Method Channel,并为其绑定事件监听方法。
- fun reportChatInfo():将用户登录信息和 SDKAPPID 透传至 Flutter Module,使 Flutter 层得以初始化并登录腾讯云IM。
- fun launchChatFunc(): 拉起或导航至 Flutter Module 所在 ViewController。
- fun triggerNotification(msg: String):将 iOS Native 层收到的离线推送消息点击事件,及其包含的ext信息,以 JSON String形式,透传至 Flutter 层 绑定的监听处理事件。用于处理离线推送点击跳转,例如至对应会话。

本单例 object 的详细代码,可以参考Demo源码。

在总入口 MyApplication 中,初始化上述对象。

在 MyApplication.kt 文件中,将全局context传入单例对象,并执行初始化。

```
class MyApplication : MultiDexApplication() {
    override fun onCreate() {
        super.onCreate()
        FlutterUtils.context = this // new
        FlutterUtils.init() // new
    }
}
```

### 监听及转发离线推送点击事件

离线推送的初始化/Token上报/点击事件对应的会话跳转处理,已在 Flutter Chat 模块中进行,因此,Native 区域,仅需透传点击通知事件的 ext 即可。 之所以这么做,是因为点击通知事件已在 Native 被拦截消费,Flutter 层无法直接拿到,必须经由 Native 转发。

```
① 说明
由于不同厂商的离线推送接入步骤不一致,本文以 OPPO 为例,全部厂商接入方案,可查看本文档。
```

在腾讯云 IM 控制台中,新增 OPPO 的推送证书, 点击后续动作 选择 打开应用内指定页面 , 应用内页面 以 Activity 方式,配置一个用于处理离线推送信 息的页面,建议为应用首页。如,我们的 Demo 配置为: com.tencent.chat.android.MainActivity 。

<b>OPPO</b> (ID: 26654)				删除	编辑
АррКеу	4c2a7b09bb8b404				
AppID					
Masterezza					
ChannelID	new_message				
点击后续动作	打开应用内指定页面	ר			
应用内市商	activity: com tencent chat android MainActivity				

在上方控制台配置的用于离线推送的 Activity 文件中,新增如下代码。



该代码的作用是,当厂商拉起相应 Activity 时,从 Bundle 中取出 HashMap 形式 ext 信息,触发单例对象中的方法,将这个信息,手动转发至 Flutter 中。具 体代码,可以参考 Demo 源码。



此时,Android Native 层编写完成。

# 附加方案:在 Native 层,初始化并登录腾讯云IM

有的时候,对于 Chat 和 Call 模块能力,您希望对于高频的简单应用场景,能深入嵌入您现有的业务逻辑中。

例如对于游戏场景,在对局内,希望能直接发起会话。

而您的完整功能 Chat 模块,使用 Flutter 实现,仅是您 APP 中一个重要性较低的子模块,因此不希望一上来就启动一个完整的 Flutter Module。 这个时候,您可以在 Native 层调用腾讯云 IM Native SDK 的初始化及登录方法,此后,便可在您需要的高频简单场景,直接使用腾讯云 IM Native SDK,构 建 In-App Chat 能力。

### 🕛 说明

当然,在此种情况下,您也可以选择提前先在 Flutter 初始化并登录腾讯云IM,此时,您将不再需要在 Native 层再次初始化并登录。两端仅需初始化并 登录一次,即可在双端都能使用。

由于 Flutter SDK 已自带 Native SDK,您不需要在 Native 层,再次引入,即可直接使用。

# Native 初始化并登录

以 iOS Swift 代码为例,演示如何在 Native 层,初始化并登录。

```
import imSDK_Pius
func initTencentChat(){
    if(isLoginSuccess == true){
        return
    }
    let data = V2TIMManager.sharedInstance().initSDK(您的SDKAPPID, config: nil);
    if (data == true){
        V2TIMManager.sharedInstance().login(
            chatInfo.userID,
            userSig: chatInfo.userSig,
            succ: {
               self.isLoginSuccess = true
               self.reportChatInfo()
            },
            fail: onLoginFailed()
            )
        }
}
```

此后,在 Native 层面,便可直接使用Native SDK,搭建您的业务功能模块。详情可查阅 iOS 快速入门 或 Android 快速入门。



# 初始化 Flutter TUIKit

如果您已在 Native 层完成初始化并登录,您不需要再次在 Flutter 层再次执行,但需要调用 TUIKit 的 \_coreInstance.setDataFromNative() ,将当前用户信息传入。

final CoreServicesImpl \_coreInstance = TIMUIKitCore.getInstance(); coreInstance.setDataFromNative(userId: chatInfo?.userID ?? "");

更详细代码,请查阅我们的 Demo 源码。



由于代码较多,本文无法全部列举,全量详细代码 (Native + Flutter),请下载查阅

# 相关文档

- 1. Integrate a Flutter module into your Android project.
- 2. Integrate a Flutter module into your iOS project.
- 3. Adding a Flutter screen to an iOS app.
- 4. Multiple Flutter screens or views.



# 快速入门(APICloud)

最近更新时间: 2024-04-12 16:42:31

### 本文主要介绍如何在 APICloud 平台使用腾讯云 IM 模块。

# 环境要求

平台	版本
iOS	最低 9.0
Android	最低 7.0

# 支持平台

支持 iOS 和 Android,并只适合用于在 ApiCloud 平台创建的应用。

在 ApiCloud 平台上开发者可使用标准 HTML5(包括HTML/CSS/JS技术以及Vue/React等框架技术)或 AVM.JS 技术进行开发,一套代码同时生成 Android & iOS 原生 App。

# 前提条件

您已 注册腾讯云 账号,并完成 实名认证。

### 操作步骤

### 步骤1: 创建腾讯云应用

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 并 获取密钥信息。

同一个腾讯云账号,最多可创建300个即时通信 IM 应用。若已有300个应用,您可以先 停用并删除 无需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. 单击创建新应用,在创建应用对话框中输入您的应用名称,单击确定。

Ś	腾讯云	· 总览 z	→品 ▼			
总览	IM 技术	服务交流群				
						创建新应用
	Test	创建应用				×
	套餐包 SDKAr	应用名称 *	请输入应用名称			
	创建时 际签	标签 (i)	+添加			
	到期时			确定		
	查					-8

3. 请保存 SDKAppID 信息。可在控制台总览页查看新建应用的状态、业务版本、SDKAppID、标签、创建时间以及到期时间。



🔗 腾讯云	总览 云产品 🗸	网站备案	+	
总览				
创建新应用				
Test	体验版			
状态	启用			
SDKAppID	6			
标签	无			
创建时间	2021-03-04 16:14:09			
到期时间	-			
升级版本				
共 Z 读				

4. 单击创建后的应用,左侧导航栏单击辅助工具 > UserSig 生成&校验,创建一个 UserID 及其对应的 UserSig,复制签名信息,后续登录使用。

於 購訊云 息號	; 云产品 → │ 网站备案 +
即时通信 IM	← UserSig生成&校验 ▼
弐 基本配置	
11 功能配置	签名 (UserSig) 生成工具   登录鉴权介绍 ビ
晶 群组管理	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。
③ 回调配置	请输入用户名 (UserID)
🕣 安全打击	user1
🗟 数据监控器	密钥 (Key)
@ 辅助工具	
- 离线推送自查	
・ UserSig生成&校	生成签名 (UserSig)
- <del>D</del> Y	当前生成的签名(UserSig)是:
	复制签名 (UserSig)

# 步骤2: 创建 ApiCloud 应用

- 1. 注册并登录 ApiCloud 平台。
- 2. 创建 ApiCloud 应用,请参见 ApiCloud 新手开发指南。

# 步骤3:在应用中添加腾讯云IM模块



#### 腾讯云官方IM模块名为 tencentCloudChatSDK 。

1. 进入 ApiCloud 网站的 模块Store,搜索 tencentCloudChatSDK 或者直接进入 tencentCloudChatSDK 模块页面。

精选	界面布局	设备访问	功能扩展	开放SDK	AI	直播 音视频	物联网	多端组件	Q			
┃ 首页 > 模块Sto	re > 开放SDK > 推	送与IM > tencen	tCloudChatSDK									
	• tencentCloudChatSDK v1.0.0											
	简洁清晰的 动能完整、 8月, 8月30日 94	iM 会活の表 A2E明年商単、 品作絵 ・ ・ ・ ・	<b>移</b> 本 節 広	5 Ⅰ <b>▲</b> ₩ ios <b>续块概述</b> 下、系统通知等能力 2用于社交沟通、互	&Android 」腾讯云即时通 。并提供完备 「动直播、游戏:	▲ 腾讯云提供 言IM SDK,提供简单 約 App 接入、后台律 交流等	2023-2-7 单实现高可靠且和 管理接口,可轻材	急定的全球介入、单耻 公拥有微信和 QQ 一标	II、群聊、资料关系链托 洋强大的即时通信能力。可			
	Constant and a second sec	2010 RE 100 RE 100 10-1070 10-1070 10-0		立即使用	■ 查看依赖库	<ul><li>○ 给模块提建议</li><li>↓ 打赏作者</li></ul>	④ 技术咨询					

# 2. 单击 立即使用 ,将模块添加到应用里。

3. 进入个人开发控制台,点击添加模块的App - 模块 - 已添加模块。如果看到 tencentCloudChatSDK 模块在已添加模块中(如图所示),则表示模块添加成 功。

		描位店	白史义横体	塔収记录	白宝Vloader	友容问调试					
▶ 码前 NEW		读水片		121人纪永	EXCLOADER	汉王问师叫					
端开发		1.	免费模块无使用限制	2. 收费模块可通过测试版	进行试用 <b>3.</b> 正式编	饭如包含收费模块,需授权后;					
🔹 端设置											
证书	类别全部	★         第         ▲             ◆         10S         □         Android         □         小程序         □         H5         □         録結模块									
<> 代码	✓ 始终使用模块最新版										
▶ 模块	es mite										
▲ 云编译	即时通信 (៣ 🕟 🚟	tencentCloudChatSD	< <u>1.0.0</u> • <b>-</b>		mam 1.0.6 ▼ -						
应用服务	ФВ. 210. 211 <u> </u> <u> </u>	本模块封装了完整的腾讯云即 现高可靠且稳定的全球介入、	时通信IM SDK,提供简单实 单聊、群聊、资料关系链托		默认APICloud会自动检测版本更 xml里配置autoUpdate为false,	新,用户也可以在config. 然后使用mam模块来检测					
🛒 版本		÷ é	免费 打赏		<b>#\$</b>	费打赏					
▶ 云修复											
三 推送											
■ 闪屏广告											

# 步骤4:调用 SDK 接口

### 1. SDK 初始化

在调用其他接口之前,需要用 sdkAppID 调用初始化接口。

var tencentCloudChatSDK = api.require("tencentCloudChatSDK") //引用模块



tencentCloudChatSDK.initSDK({ //调用初始化接口 sdkAppID: '', logLevel: 0, uuid:'', },function(ret,err){})

可查看 模块文档 在 App 中调用相应的 SDK。

#### 2. 用户登录

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。

```
var tencentCloudChatSDK = api.require("tencentCloudChatSDK")
tencentCloudChatSDK.login({
    userID:"userID",
    userSig:"userSig"
},function(ret,err){})
```

当返回值的 code 为0时,表示登录成功。其他接口的成功返回值的 code 均为0。

### 🕛 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。

### 3. 发送单聊普通文本消息

初始化并登录成功后可以尝试发送消息。除普通文本消息之外,SDK 还支持发送自定义、图片、文件、视频、地理位置等消息。具体内容可以参考模块文档。

var tencentCloudChatSDK = api.require("tencentCloudChatSDK")
tencentCloudChatSDK.sendC2CTextMessage({text:"",userID:""},function(ret,err){})

### 🕛 说明

文本消息支持云端脏词过滤,如果用户发送消息时有敏感词,回调就会返回80001错误码。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。请在控制台关闭好友关系链检查。

#### 4. 获取会话列表

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。 常见应用场景为:

在启动应用程序后立即获取会话列表,然后监听长链接以实时更新会话列表的变化。

var tencentCloudChatSDK = api.require("tencentCloudChatSDK")
tencentCloudChatSDK.getConversationList({nextSeq:,count:1},function(ret,err){})

### 🕛 说明

若上一步骤消息发送成功,则可以在会话列表中看到发送成功的消息。

#### 5. 添加高级消息的事件监听器

可以通过添加高级消息事件监听器,在有新的消息时调用回调函数。设置事件监听器如下:

var tencentCloudChatSDK = api.require("tencentCloudChatSDK")
tencentCloudChatSDK.addAdvancedMsgListener({"uuid":""},function(ret,err){})

设置成功后可以通过 setEventListener 监听。

以上您已经基本完成了 IM 模块的开发,可以发送接受消息并且进入会话列表。 您可以继续完成 群组,用户资料,关系链,离线推送,本地搜索 等相关功能开发。

### 步骤5:编译并运行

1. 下载 APICloud Studio。



2. 点击 模块 - 自定义loader 分别编译 iOS 和 Android 自定义 loader。编译自定义 loader 需要 iOS 和 Android 证书。

已添加模块	模块库	自定义模块	授权记录	自定义loader	友空间调试						
介绍											
AppLoader是一个集成了众多APICloud扩展模块的应用加载器。开发APICloud应用可以使用APICloud Studio3:谱试APICloud应用则需要使用AppLoader。 AppLoader分为官方Loader和自定义Loader,官方Loader只包含常用官方模块,而其他开发者的模块、第三方SDK模块等未加入到官方Loader中;自定义Loader,是开发者为自己的应用定制的Loader,开发者可以添加本应用所需要的模块后进行Loader的编译。更多区别详见《 自定义Loader说明》 在开发工具APICloud Studio3中也可以进行Loader的编译。											
编译iOS自定义loa	der 测试证书 正式证书 您还未上传iOS测试证书,	<sup>我</sup> 立即 <u>上传</u>				版本: 0.0.0	① 下载				
编译Android自定义k	pader					版本: 00.00.02	0 Tấ				

3. 编译后下载到相应的真机或模拟器中可以通过 ApiCloud Studio 使用 Wi-Fi 真机同步或 USB 真机同步对 App 进行实时编译调试。 具体内容可以参见 ApiCloud 文档 。

### 步骤6:开通云端审核功能

在消息发送、资料修改等场景中,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,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

- 添加 tencentCloudChatSDK 模块后编译失败?
   可能是与您的 App 中已添加的其他腾讯云 IM SDK 模块冲突。请您检查您的 App 中是否还存在其他腾讯云 IM SDK 模块,并删除。其他腾讯云 IM 模块已全 部停止维护,若以前使用过其他腾讯云 IM 模块,请您移植到官方的模块。
- 运行时显示模块未添加,但在已添加模块中显示已添加?
   编译时使用自带的云编译会出现上述问题。需要使用自定义 loader 进行编译。

# 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。



# 快速入门(HarmonyOS)

最近更新时间: 2024-04-09 15:26:12

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。

### 🕛 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**。

即时通信 IM	应用管理 IM 批	5术服务交流群 🖸 🛛 IM 🗄	海交流专区
<b></b>	创建新应用	申请开发版	
管理与配置	应用名称	SDKAppID	应用版本

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。

数据中心 🛈 * 中国 数据存储在中国,支持全球接入							
标签键	标签值						
	中国 数据存储在中国	中国 数据存储在中国,支持全球接入 标签键					

### 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。

<del>]</del> 应用管理	创建新应用	申请开发版	Ę		请输入 SDKApp	Q				
管理与配置	应用名称	SDKAppID (j)	应用版本 🛈	服务状态	数据中心 🔻	创建时间	到期时间 う	标签 ()	操作	
品 <sup>1400</sup> 二 账号管理		1400	体验版	使用中 🛈	中国	2023-08-16	-	-	应用详情 版本对比 查看密钥 标签管理	
<b>器 群组管理</b>		1400	体验版	使用中	中国	2023-04-24	-	-	应用详情 版本对比 查看密钥 标签管理	
□□ 功能配置 ~	1.获取	SDKApp	ID 信息	使用中	中国	2023-03-20	-	-	应 2. 点击【查看密钥】	
③ 回调配置										

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。



查看密钥	4. 复制显示后的密钥信息 ×
() 密钥信	信息为敏感信息,请注意保密,不要泄露。
密钥	
	显示密钥 ● — 3.点击【显示密钥】

# ▲ 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。

```
① 说明
为尊重表情设计版权,下载的 Demo 工程中不包含大表情元素切图,您可以使用自己本地表情包来配置代码。未授权使用 IM Demo 中的表情包可能
会构成设计侵权。
```

- **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。

<pre>D/** ##元 SDKAppId,需要整换均您自己账号下的 SDKAppId, *  * 提入欄甙云五通信[控制台](https://console.cloud.tencent.com/avc ) 创建应用, 即可着到 SDKAppId, * C2是编示元用压公客户的询● kdl. &gt; /* export const SDKAPPID = 14 2; /** * 签名过期时间, 建议不要设置的过程 *  * 括认词谓: 7 x 24 x 60 x 60 = 60:4800 = 7 天 </pre>	<b>DHAppIC</b> 14 28百 应用名称 TRTC 应用类型 游戏 应用简介 -
C =/ const EXPIRETIME = 604000;	基础信息
<pre>/**</pre>	密钥 d15b6 23bd <b>陸進花兒</b> 密閉信息为敏感信息,请注意保密,不要泄露。 的建时间 2022-11-30 最近修改时 2023-02-21 问

#### ▲ 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:



	Ś	0	DevEco Studio	File	Edit	View	Navigat	e Code	e F	Refactor	Build	Run	Tools	Git	Window	Help
(				New				>							imde	mo
	in 🖌	nder	mo	<b>b</b> 0	pen						_					
-		-	Project -	Rece	nt Proje	cts		>								
oio			Project	Close	e Projec	t										
à	Ē		imdemo ~/Proj	Close	e All Pro	jects			no							
			.hvigor	Close	e Other	Projects	s									
time:											-	7				
000	50		dependenciu	P	roject S	tructure		<b>៤</b>								
~	>		entry	File P	ropertie	es		>								
			hvigor	Loca	Histor	/		>								
			oh_modules													
			🛃 .gitignore	🖬 s	ave All											
			build-profile.	GR	eload A	ll from D	Disk									
			ᡖ hvigorfile.ts	Sync	and Re	fresh Pr	oject									
			hvigorw	Repa	ir IDE											
			🛔 hvigorw.bat	Invali	date Ca	iches										
			local.proper	Mana		Catting										
			oh-package.	Nana		Settings	5									
			oh-package	New	Projects	s Setup		,								
			I External Librarie													
		<b>(</b>	Scratches and	Ехро	rt			>								
				🖶 P	rint											
				Powe	er Save I	Mode										ĺ



2. 连接真机,编译运行:



			imdemo			_	 
	a indemo	٥	💼 entry 🔻 🔲 🗖 N	9 🔻		<b></b>	
ject	■ Project						
Pro	🗠 🐂 imdemo ~/Projects/doc/TIMSDK/HarmonyOS/Demo/imdemo		Т		Т		
	> 🖿 .hvigor						
ij	> 🖿 .idea						
mma	> 🖿 AppScope						
ŏ	> 🖿 dependencies						
~	> 📑 entry						
	> 🖿 hvigor		· · · · · ·		1.1		
	🐻 .gitignore		<b></b>				
	🧑 build-profile.json5				U		
	🛃 hvigorfile.ts						
	D hvigorw						
	🖆 hvigorw.bat						
	local.properties						
	🗑 oh-package.json5						
	🗑 oh-package-lock.json5						
	> IIIII External Libraries						
	Consoles						

# 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)