

即时通信 IM 控制台指南 产品文档





【版权声明】

©2013-2025 腾讯云版权所有

本文档著作权归腾讯云单独所有,未经腾讯云事先书面许可,任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其他腾讯云服务相关的商标均为腾讯集团下的相关公司主体所有。另外,本文档涉及的第三方主体的商标,依法 由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况,部分产品、服务的内容可能有所调整。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。



文档目录

控制台指南 新版控制台介绍 创建并升级应用 基本配置 功能配置 账号管理 群组管理 回调配置 用量统计 资源包查看指南 实时监控 开发辅助工具 访问管理 授予子账号操作控制台权限 预设策略 自定义策略 高级功能 云端搜索功能 效果展示 集成指引 Android&iOS&Windows&Mac Web & uni-app & 小程序 文本消息翻译功能 效果展示 集成指引 Android & iOS 投票功能 效果展示 集成指引 Android & iOS Flutter 接龙功能 效果展示 集成指引 Android & iOS



会话标记功能
 效果展示
 集成指引
 Android & iOS
 会话分组功能
 效果展示
 集成指引

Android & iOS



控制台指南 新版控制台介绍

最近更新时间:2025-01-07 16:42:32

本文旨在帮助您了解如何快速上手新版控制台。

说明:

1. 新版 Chat 控制台保留了腾讯云 Tencent Cloud 官网的账号体系和 CAM 鉴权等能力,您可以用原先的账号进行登录并管理使用您的 Chat 资源。

2. 新版 Chat 控制台支持英、中、日、韩等多语言,如您在使用过程中遇到问题,请联系我们。

新控制台界面介绍

1. 顶部导航

包含了账号信息、费用中心、工单入口、产品文档、Demo体验馆。



2. 概览页全新布局

控制台概览包括了应用管理、套餐包管理、常用链接等公共模块。新的控制台可以同时管理Chat和RTC Engine、 Call、Conference、Live、Beauty AR等产品应用,对于同时使用多产品的用户来说,无需在多个控制台内反复跳转。



 · 应用 · · ·	概览 ① 忽尚未提供支付方式,在您用完免费资源后, ;	我们将留停您账户的服务。为避免服务中断,请 完善	8的信息并刷新 。			
⑦ 监控仪表盘 🛛 🗸	0 应用				所有应用(10)	
 ご 套餐包管理 、 ご 相关服务 へ 		ch	te	ch	•	应用管理
• 美颜特效	1 创建应用	Products ✓ Call ✓ Chat	Products O Call	Products ① Chat	۲	
	示例代码 & 体验Demo		餐包 ① 不支持按量计费		套簽管理 (1)	
	运行示例代码 体验V 现在就开始构建音视频通 所见即 话应用	Veb Demo 斯得	Free Trial		•	套餐包管
	\rightarrow \land 2 \rightarrow	自动续	间: 2025-01-08 (UTC+08:00) 费: -			
	 → 2 → ● ●<	日初時	间: 2025-01-08 (U1C+08:00) 费: -		文稿 [2	
金約這用在10000分钟免 费时间后结束。	 → ◆ ◆<td>1)(m) 日本(新 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本) 日本(新 日本) 日本) 日本) 日本) 日本) 日本) 日本) 日本) 日本) 日本)</td><td>□11 2025-01-08 (U1C+08:00) 贡: - - - - - - - - - - - - - - - - - - -</td><td>① 加入我们的社区 GitHub Twit Telegram You</td><td>文档 [3] ter ube</td><td>常用链</td>	1)(m) 日本(新 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本(新 日本) 日本) 日本(新 日本) 日本) 日本) 日本) 日本) 日本) 日本) 日本) 日本) 日本)	□11 2025-01-08 (U1C+08:00) 贡: - - - - - - - - - - - - - - - - - - -	① 加入我们的社区 GitHub Twit Telegram You	文档 [3] ter ube	常用链

说明:

当前套餐包管理页面只展示与用量相关的套餐包。如果想查看Chat和美颜的预付费账单信息和所有后付费账单信息,请前往【费用中心】-【账单中心】查看。

3. 应用管理

Chat通过应用来管理不同的业务和项目,需创建应用并开通Chat服务(领取任一版本)才能接入Chat服务。 新控制台下,每个应用下可以开通多个服务,除Chat外,还可以管理开通Call、Conference、Live、RTC Engine、 In-game Voice Chat等。

3.1 应用列表

可以在这里创建新的应用,并查看创建的所有应用的基本信息,包括SDKAppID、状态、部署区域等。



器 概览		← 应用				应用列表				
∅ 应用										
🔄 使用统计		🥑 我的应用	搜索应用			Q	创建应用		•	划建应用
✓ 监控仪表盘	~	应用名称	SDKAppID	状态	部署区域	产品信息 🔽	过期时间	SDKSecret	操作	
	~									
🔁 相关服务	^					Call:体验版 Chat:体验版	2024-12-18 17:26:55 (UTC+08:00) 2025-01-11 17:25:12 (UTC+08:00)	***** ©	Ð	ø
• 美颜特效										
8 开发工具	~					Call:无版本		***** ©	Ð	2

3.2 应用管理详情页

在应用列表中点击应用名称,即可进入应用管理详情页。

应用概览:有关该应用的基本信息。

高级功能:有关这个应用下通用的高级服务配置。

服务管理:每个应用下可以开通多个服务,除Chat外,还可以管理开通Call、Conference、Live、RTC Engine、In-game Voice Chat等服务。详情页可以查看该服务的在用版本信息,并进行相关配置和升级操作。

应用概览	☑ 应用概览	*	应用概览	
高级功能	◎ 高级功能			
	产品		准备开始构建了	了吗?
	🕲 Call			
	武 Conference		您可以选择从这里开 与我们的专家交谈 [2	始,或
	((•)) Live			
服务管理	😚 RTC Engine		■基本信息	
	💬 Chat		应用名称	chat
	凤 游戏内语音 🖸		SDKAppID	20015596
	相关服务		描述	-
	旧大战劳		状态	已启用 更多 ~
	🙂 Beauty AR 🛽			

3.3 Chat服务配置



选中一个应用,点击左侧导航栏中的Chat选项,可以在这里对每个应用中的Chat服务进行单独配置。

Ø 应用概定	概览	概览 当前数据中心:新加坡 ③ 加入 Tele	gram 技术交流群组 加入 WhatsApp 3	交流群			
◇ 高級功能	账号管理						
产品	群组管理	■ 应用信息			升级套餐包		快速开始
🕲 Call	功能配置 ^	应用版本 体验版 ①	到期时间	2025-01-11		通	过文档逐步指导您完成操作
[2] Conference	• 登录与消息	应用状态 使用中 ①	续费状态 密钥	未开启 ****** 显示廖钼		ж	Jikit 开始 从 SDK 开始
((+)) Live	• 好友与关系链	зымфро					
🕅 RTC Engine	• 用户自定义字段						
💬 Chat	• 群组配置	O数据总览(截止2024-12-12 17:56	:51)				1
🕟 游戏内语音 🖸	回调配置	本月累计MAU(个)	截至昨日累计用户数(个)		昨日新注册用户数(个)	4	5月峰值群组数(个)
相关服务	数据统计	0 0	1		1	(0
😳 Beauty AR 🖸							
	推送 个						
	• 接入设置	内 休验核心功能和预览 []]					
	• 接入测试						
	• 推送记录			WED	MODILE	Faallah ya	
	- 推送数据			WED	MOBILE	English V light V	
	• 推送排查		Lester	<u>Q</u> Call	Robot	B (C	
	高级功能		O Search	•	No More		
			C Obarchi		Click the call button in the top r experience it immediately.Let is	right corner to have a call to discuss	
	实时监控		Call Robot Click the call button in the to 1	17:56	where to go next weekend.	17:56	
	开发工具 ^		Rosa You can try sending me text1	17:56			
	。 高线推送自查			ß			
	• 自助排障日志		Alice:[image] 1	17:56			
	集成指南						
=							

4. 用量管理

使用的所有服务所产生的用量,都可以在用量管理进行查看。在我们的用量统计面板中,可以通过选择不同应用、 功能、场景和时段来来获取多样的用量统计结果。



○ 小工 ① 計量計1番気分物効果-次.数量可度和5-180分物効果の生まれ、 送择应用与产品 ② 监控仪表盘 > 2015488 - test ○ 加車 ③ 监控仪表盘 > ○ 加山 1000000000000000000000000000000000000	Max. 选择应用与产品 健康CSS的录制功能、放井X 制任券费約增值 C计员、请在CSS控制台 企量着您的使用情况. 送择场景 202412-06 ~ 202412-12 Chat今日毀援憲次日可见:该页面时间按照UTC+6时区记录和显示 选择时民 健至昨日累计用户数 (个) 1 软前一日 + +0.00% PO-MAU	∅ 应用
② 监控仪表盘 ~ ① D15488-test ▲ Drat ● 用户规模 消息活跃 詳组現機 选择场景 ③ 常长回管理 ~ ● 天 昨天 最近7天 最近4天 202412-06 - 202412-12 ○ Dau-9 日致浅陽常次日可见; 该页面时间按照UTC+8时区记录和显示 ④ 相关器务 ^ -	用户成模 消息活跃 群组成健 选择场易 202412-06 - 202412-12 Chat今日数据需次日可见:该页面时间按照UTC+8时区记录和显示 选择时的 截至昨日累计用户数 (个) 作日新注册用户数 (个) 1 较前一日 + +0.00% ① ①	☑ 使用统计
2 客餐包管理 ◆天 作天 最近4天 2024-12-06 ~ 2024-12-12 Chat+今日数据需次日可见:该页面时间按照UTC+8时区记录和显示 1 村关照务 ▲ 2 展标物 ▲ 2 展标物 ▲ 3 开发工具 ▲ 1 以SerSig 工具 ▲ 1 USerSig 工具 ● RTMP地址生成器 本月累计MAU (个) ① ① ● 数据趋势 2024-12-16 To 2024-12-11 聚计MAU 累计注册 用户新地 ▲ ▲ ● ●	202412-06 ~ 202412-12 Char今日数据需次日可见:该页面时间按照UTC+8时区记录和显示 截至昨日累计用户数(介) 作日新注册用户数(介) 1 较前一日 * +0.00% ① ①)监控仪表盘 >
1 日长銀务 へ 美鮮校X ・ <td>截至昨日累计用户数(个) 1 较前−日 ↑+0.00% ● MAU</td> <td>] 套餐包管理</td>	截至昨日累计用户数(个) 1 较前−日 ↑+0.00% ● MAU] 套餐包管理
美酸特效 「 「	截至昨日累计用户数(个) 1 较前一日 t +0.00% ● ○ ○	1 相关服务 ^
 ○ 开发工具 ^ ▶ UserSig 工具 ▶ RTMP地址生成器 ▲ 月累计MAU (个) ① ① ① ① ① ① ① ① ① ② ● 数据趋势 2024-12-06 To 2024-12-11 ■ 累计注册 用户断增 ● - MAU 1 	截至昨日累计用户数(个) 1 按前一日 ★+0.00% ● ③	美颜特效
UserSig I具 O ① O ① I 较前一日 ↑ +0.00% O ① O ① O ① D	0 ⊙ -O- MAU	开发工具 个
RTMP地址生成器 数据趋势 2024-12-06 To 2024-12-11 東注油 用户新増 -〇- MAU	-O- MAU	UserSig 工具
数据趋势 2024-12-06 To 2024-12-11	-O- MAU	
数据趋势 2024-12-06 To 2024-12-11	-O- MAU	• RTMP地址生成器
累计独和U 累计注册 用户新增 -O- MAU MAU 1a 1a	-O- MAU	RTMP地址生成器
-O- MAU	-O- MAU	RTMP地址生成器
		RTMP地址生成器
		• RTMP地址生成器
您的免费试用已经结束。 0.8-		 RTMP地址生成器
升级计划 0.6-		 RTMP地址生成器 愈的免费试用已经结束。

5. 监控仪表盘

可以选择订阅我们的监控仪表盘服务,此服务支持房间通话调查功能、实时监控和数据大盘功能。了解详情。

6. 套餐包管理

6.1 套餐订阅

我们的每个产品都为提供了多种套餐选择,在这里详细介绍了每个套餐所包含的内容,可以根据自己的项目选择适 合自己的产品和套餐。



概览		Call Conference Live RTC Engine	Chat Beauty AR (Web)) 数据监控	选择产品
应用					
使用统计					
监控仪表盘	^		体验版	标准版	进阶版
房间通话调查			60/5日	6 300 / E	t (00 / E
实时监控			\$0/母月	\$3997 H	\$ 6997 H
数据大盘 Beta			免费体验	立即订阅	立即订阅
nder der fan federeren			不支持按量计费	支持按量计费*	支持按量计费*
(餮包官埋	^	全球覆盖	×	×	×
惠活动 🎍		最大田户数	无限制	无限制	无限制
餐订阅			ZUNANO	ZUPARS	ZUPANS
餐管理		单个用户的最大好友数	20	3,000	3,000
目关服务	^	单个用户可加入的最大群组数	50	500	1,000
美颜特效		单个用户可加入的最大非音视频群组数	20	200	2,000
		最大非亲如惩群纽数(不句托已解数			• • •
干发工具	Ý	的)	100	无限制	无限制
		最多可创建的音视频群组数	50	50	无限制
		每日群组数量最大净增量	100	10,000	10,000
		正由兴自会表现初期	77	77	207
		/// 文/月忌光贷休笛期		17	30天
		免费月活用户(MAU)配额 🛈	100/月	10,000/月	10,000/月

6.2 套餐管理

在这里可以了解到订阅的套餐的基本信息,包括套餐的开始时间、结束时间、剩余时间、扣除详情等。

器 概 览	*	套餐管理										
∅ 应用			大你用你会表次酒	后 杂灯波频度保收占	的职业头戏的现在分词	5、海中草的小水白 计图66						
一 使用统计		1 应问木提供又刊力为	1。任志市元光贞贞称	心,我们村首学态成/~	的服务。为此无服务中的	1, 用元音忘的旧志开剧剧。						
✓ 监控仪表盘 ^		右边 干边										
• 房间通话调查		有效 无效									费用	
• 实时监控		套餐名称	套管包ID	套餐来源	时长	本周期剩余时间	本周期开始时间 ①	本周期结束时间 ①	套餐包开始时间 ①	整个套餐包的结 束时间 ①	本周期状态	操作
• 数据大盘 Beta												
		TRTC Free Trial		免费	10000分钟 * 12 月	10000 mins 100%	2024-12-09 00:00:00	2025-01-08 23:59:59	2024-07-09 00:00:00	2025-07-08 23:59:59	● 未使用	扣除详情
• 优惠活动 ●		共1条								10 *	条/页 🕅 ┥ 1	/1页 →
。 套餐订阅												
 · 套餐管理 												

说明:

当前套餐包管理页面只展示与用量相关的套餐包。如果想查看 Chat 和 美颜 的预付费账单信息和所有后付费账单信息,请前往【费用中心】-【账单中心】查看。

7. 相关服务-美颜特效

可以在这里管理并配置 Beauty AR 产品。了解Beauty AR。

8. 开发工具



8.1 UerSig工具

在这个页面进行 UerSig 的生成和验证。

概览	6 UserSig T B	生成UserSig	验证	EUserSig
 ③ 应用 (二) 使用统计 	 のものものは、このでは、このでは、このでは、このでは、このでは、このでは、このでは、この	服务。为避免服务中断,请完备您的信息并刷新。		
 ◇ 监控仪表盘 × ◇ 套餐包管理 × 	签名(UserSig)生成器 此工具可以快速生成UserSig,可用于运行演示和调试功能。	•	◆ 签名 (UserSig) 验证器 此工具用于验证您使用的 UserSig 的有效性。	•
 目 相关服务 ^ 美顔特效 	<u>应</u> 用 (SDKAppID) 用户名 	(UserID) ① 周户名	应用 (SDKAppID) 选择一个应用	用户名 (UserID) ① 设置用户名
 因 开发工具 へ UserSig 工具 	SDKSecretKey 选择应用后自动生成		SDKSecretKey 选择应用后自动生成	
- RTMP地址生成器	生成		UserSig 请输入	
企 您的试用在10000分钟免 费时间后结束。 升级计划	生成结果	石 复制	Verify	
 3.2 RTMP 地址生成器	<u>.</u>			

此页面用于生成 RTMP 推流功能的推流地址。



器 概览	← RTMP地址生成	39 10		
@ 应用				
🖾 使用统计	 您尚未提供支付方式。在您用完 	免费资源后,我们将暂停您账户的服务,为避免服务中断,请完善您的信息并刷新。		
✓ 监控仪表盘 ∨	此页面用于生成TRTC RTMP推	^斋 己功能的推流地址,适用于测试阶段。对于正式线上业务,请将地址生成逻辑部署到	川后端服务进行	计算生成,以避免加密密钥泄露导致流量被盗用。
☑ 套餐包管理 ∨				
□ 相关服务 ^	风 推流地址			
• 美颜特效	应用 (SDKAppID)	选择一个应用	~	
氏 开发工具 个	推流房间名(RoomID)	设置房间名		Ũ
• UserSig 工具	推流用户名 (UserID)	设置用户名	(0
• RTMP地址生成器				
	生成	拼接地址说明		
	生成结果			TRTC RTMP推流
	RTMP推流地址	٩ ٩	□ 复制	
	OBS服务器	٩ ٩	□ 复制	
٨	OBS 推流密钥	q	□ 复制	
您的试用在10000分钟免 费时间后结束。	输入房间授权参数Usersig有效	期,默认为180天。		
升级计划				
Ξ				

Chat 应用创建流程和配置页面位置调整

1. Chat 应用创建流程

在 控制台概览页 点击创建应用。



Tencent RTC		
品 概览		概览
Ø 应用		① 您尚未提供支付方式。在您用完免费资源后,我们将暂停您账户的服务。为避免服务中断,请完善您的信息并紧
🖉 使用统计		
✓ 监控仪表盘	~	② 应用
☑ 套餐包管理	~	
🖯 相关服务	^	ch chat
• 美颜特效		· · · · · · · · · · · · · · · · · · ·
开发工具	*	创建应用 ✓ Call ✓ Chat

在弹出的窗口中输入应用名称,选择产品、版本与部署区域之后,点击创建。



说明:

有关部署区域的选择:

1. Chat 在全球部署多个数据中心,所有数据中心均支持全球接入,但不同数据中心之间的用户无法互通。请根据主要用户分布情况或合规要求,谨慎选择最适合您业务的数据中心。一旦选择,后续将无法变更。



 2. TRTC的数据中心默认为新加坡,不支持用户自行选择。国际站相关数据和隐私条款可见 隐私条款。
 3. 我们支持全球接入,不受部署区域所限制。我们在全球各国家部署加速节点近3000个,终端登录时,Chat SDK和 TRTC SDK会访问就近接入点或加速点,通过最优调度策略实现加速传输。

2. 如何在控制台中配置Chat应用

在 控制台概览页 中选择Chat应用,来到应用概览页。

Tencent RTC				
吕 概览		概览		
∮ 应用		 您尚未提供支付方式。在您用完免费资源后, 	,我们将暂停您账户的服务。为避免服务中断,请 <mark>完善您的信息</mark> 并	-R
<и />				
♂ 监控仪表盘	~	② 应用		
✓ 套餐包管理	~			
う 相关服务	^	+	ch	
• 美颜特效		创建成用	Products	
○ 开发工具	~	初建应用		

在 应用概览页 中总览应用概况,包括基础信息、增值功能和开通的能力。



Tencent RTC 🝙 🛛 chat	v			8 新手特意:前3个月仅需9.9美	元起 🌖	Demo 文档 帮助 ~ 简体中文 ~	ŝ
④ 应用概范	← 应用概览						۵
◎ 高級功能							-
##	准备开始构建了吗?	集成文档	Integration drawerstr	运行示例代码	*** 体验	Web Demo	
Call	你可以讲想!!?? 田平地 动	一步步帮助您完成	Neurona de referencia ha especiale de la comparte d	下载并在几分钟内运行代码	*************************************	式清晰的视频通话	
Conference	与我们的专家交谈[2]	•	 - With Network Parts of another - Define the service - The destruction of the the destruction of the Control 				
(··) Live							
	■ 基本信息				♥ 高级功能		1
「 激致の損害 !?	应用名称 chat SDKAppID ()	Si	DKSecretKey ****** 課題計値 2024-12-11 17:25:10 (UTC+08:00)		云端录制 ① 已禁用 转推CDN 已禁用		
相关服务	描述	er.	署区域 新加坡		服务端回调 ① 已禁用		
© Beauty AR [2	状态				高級权限控制 ③ 已禁用		
	■ 产品						
	😵 Call		🤛 Chat				
	版本 Call:体验版 > 过期时间 2024-12-18 17:26:55 ①		版本 Chat:体验版 过期时间 2025-01-11 17:25:12 ①				
	自动续费		自动续费				
	购买審餐	立即集成	购买窖餐	立即集成			
	激活更多产品						
	B Conference	激活	(0) Live	激活	8 RTC Engine		激
A	游戏内语音聊天	215					
您的试用在10000分钟免 费时间后结束。 升级计划							

在左侧导航栏单击 Chat, 进入 Chat 功能管理。旧版控制台的三大管理模块:配置与管理、插件服务、开发辅助, 皆在此展示。

配置与管理:概览、账号管理、群组管理、功能配置、回调配置和数据统计。

插件服务:推送和插件市场。

- 3

开发辅助:实时监控、开发工具和集成指南。



	@ 应用概览	概览	概览 当前数据中心:	新加坡 ③ 加入 T	「elegram 技术交流群组 加,	入 WhatsApp 交涉	ίt#¥		
配置与管理	◇ 高級功能	账号管理 	■ 应用信息						
	产品 ② Call	群组管理 功能配置	□ 四用信息	TABLE O	网络叶间	2025 01 11	升级套餐包		快速开始
	[2] Conference	回调配置	应用版本 伊	用中①	续费状态	2025-01-11 未开启		C,	人 Ulkit 开始 从 SDK 开始
	(+) Live	数据统计	SDKAppID		密钥	***** 显示者	图		
	💮 RTC Engine								
插件服务	💬 Chat	高级功能	○ 数据总览(4	截止2024-12-18 16	59:13)				
	○ 游戏内语音		本月累计MAU(个)		截至昨日累计用户数(个)		昨日新注册用户数(个)		本月峰值群组数(个)
	個大服务 ⓒ Beauty AR [2]	实时监控	0 0		】较前一日 ↑+0.00%		0 0		0 0
开发辅助									
			📮 体验核心功能	能和预览 UI				免费体验 Call	通过音频视频通话增强您的
						WEB N	OBILE	English V	/ light v
				Leonard		Q Call Robo	ıt		e (C
				Q Search	Ð	Olivit	No Mor	9	

常见问题

Q:应该选择使用国际站?还是国内站?

A:建议当您使用非中国大陆主体,需要使用美金与腾讯云结算时,使用国际站的产品和服务。

Q:两边的服务和功能有不同吗?

A:国内站(产品名为"即时通信IM")和国际站(产品名为"Chat")的服务和功能、以及接入使用方式一致,请放心使用。

如果在使用过程中有任何疑问,欢迎向我们反馈。点此提交工单或联系我们。



创建并升级应用

最近更新时间:2025-03-10 15:00:57

操作场景

本文指导您如何在即时通信 Chat 控制台创建开发版应用获取 SDKAppID,以及指导您如何将应用升级为标准版或专业版、专业版Plus、企业版应用。

前提条件

已 注册腾讯云 账号并添加支付方式。

创建应用

1. 登录 即时通信 Chat 控制台, 单击 Create Application。



Overview	Starter Deal! First 3 months from only \$9.9/mo.! Enjoy an 90% off starter discount on both Conference and RTC Engine for at least 3 months!
 Applications Usage Statistics Data Monitoring ~ 	Overview Overview Overview Overview Overview
Relevant Services Beauty AR Development Tools	Applications Alt Applications (1) Products Create Application
	 Sample Code & Demo Fry Web Demo Wat you see is what you get → →

2. 在**创建应用**对话框中输入您的应用名称,单击**确认**。在创建弹窗中选择 Chat 并输入应用名称,选择数据存储地区,单击 Create。



pplication name	first_app				
	The application name ca	n contain only digit	s, letters, and unders	cores.	
elect product	Call UIKit		in in Chat C 0	M diam C Diamana Diama	
	Conference	JIKit	Image: Section 2010 The subsection 2010 and a final field in the section 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010 Image: Section 2010 The subsection 2010 Image: Section 2010	We use hereinformer as hereit sore sore where search's series where search's series We use in date date with the figure	
	• Chat UIKit		Image: A set of the s	And the second second	
	RTC Engine			T Det senage 0 0 0	
/ersion	Free Trial 1Month	Free for 100 MAI	U every month	Version De	tails 🗸
)eployment Region (j)	Silicon Valley (Glob	ally communicat	ole)		~
	All our services are globa specify Chat service dep	ally communicable, ployment and data	regardless of region a storage.	selection. Regions o	nly

3. 创建完成后,可在当前 Chat 应用详情页面或 Application Overview 查看新建应用的状态、业务版本、 SDKAppID、创建时间以及到期时间。

说明

新建应用的业务版本默认为开发版,状态默认为启用。

同一个腾讯云账号,最多可创建300个即时通信 Chat 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。



Ø Application Overview	Overview	Overview Cur	rent data center: Singapore 🛈	Telegram group Whats	App group		
	Users Groups	Data Overview(I	End of 2024-09-24 14:33:32)			Application	View mc
𝔅 Call☑ Conference	Configuration V	MAU O ©	Yesterday's Total Users	Yesterday's New Users	Current Month's Peak Groups	Information Application Name	first_app
(··) Live	Statistics					Application status Application	In use Develop
💬 Chat	Push 🗸	Users Group	25			version	er edition
 In-game Voice Chat 	Plugins	7 Days 14 MAU Tota	9-24 23:59:59 📋	Expiration time SDKAppID	2024-10- 24		
	Monitor						6
	Integration Guide						
			No	data			

升级应用

说明

应用升级至标准版或专业版、专业版Plus、企业版,不支持退回至开发版。当应用出现欠费、退费导致停服后,如需继续使用原应用,您可以续费标准版或专业版、专业版Plus、企业版套餐包。如果您想重新使用开发版,请创建新的应用。

1. 单击 Application version 的注意符号,在弹框内查看当前版本功能支持详情。





2. 单击 Version comparison 查看所有版本差异比较。



Ø Application Overview	Overview							
	Users	Compare Upgradeable I	tems		\otimes			
& Call	Groups	Item	Current Spec	After Upgrade to Standard	After Upgrade to Premium	ak	Application information	View mo
Conference	Webhook	Max Users	Unlimited	Unlimited	Unlimited		Name	iirst_app
(••) Live		Max Friends	20	3000 🕇	3000 🕇		Application	In use
TC Engine		Max Groups One Can Join	50	500 🕇	1000 🕇		Application	Develope
G KIC Engine		Max Audio-Video Groups	10	50 †	Unlimited †		version	r edition
Chat	Push	Max Members per Non- Audio-Video Group	20	200 🕇	2000 🕇		Expiration	() 2024-10- 24
In-game Voice Chat	Plugins	Free Peak Groups	100 groups/month	Unlimited †	Unlimited †		SDKAppID	20012392
	Monitor Dev Tools Integration Guide	Free peak MAU limit	100 MAU/month	10000 MAU/month 1	10000 MAU/month †			ē
		Historical Message Storage	7 days	7 days	30 days 🕇			
		Audio/video call	-	Can increase to Group call version †	Can increase to Group call version †			
		Concurrent logins on multiple devices on the same platform	Supported	Unsupported	Supported			
		User status, etc. functions	Supported	Unsupported	Supported			
		Read receipts for group messages, targeted group messages and other features	Supported	Unsupported	Supported			
		Message history for new members, list of online members, broadcast messaging and other features of the audio- video group	Supported	Unsupported	Supported			
		Community, topic and other features	Supported	Unsupported	Supported	\mathbf{C}		
			Upgra	de Plan				

3. 单击 **Upgrade Plan**,前往购买页并选择需升级的版本。购买完成后,您可前往 Application Overview 查看应用信息。



基本配置

最近更新时间:2025-06-23 11:23:59

登录 控制台,单击目标应用卡片,进入应用的基础配置页面,您可以根据实际业务需求管理该应用的基本配置。

应用套餐包信息

在**应用套餐包**区域内您能看到应用当前的业务套餐信息,并能进行以下操作: 升级该应用的套餐包。 停用/删除该应用。

升级套餐包

单击应用套餐包区域的升级套餐包,您可以升级应用的业务套餐版本或配置。具体操作请参见升级应用。

延长体验版有效期

当应用套餐包为体验版且超过一个月有效期时,应用将处于到期停用状态。如需继续使用**体验版**测试开发,可在控制台对该应用进行**申请延期**操作。

1. 前往控制台-Chat,单击目标应用卡片,进入该应用的基础配置页面。

2. 单击应用套餐包区域的申请延期。

3. 单击确认完成体验版延长一个月有效期申请。

停用/删除应用

前往 控制台-应用概览 进行操作,请注意删除、停用应用后,所有子服务将会停服。

体验版应用

支持手动停用

在基本信息区域,单击状态右侧的停用,在弹出的停用提醒对话框中,单击确定即可停用。

支持手动删除

在基本信息区域,单击状态右侧的删除,在弹出的停用提醒对话框中,单击确定即可删除。

标准版、专业版、专业版Plus、企业版应用

欠费24小时后自动变为**停用**状态,如需删除,请联系我们删除该应用。

退费后自动变为停用状态,如需删除,请联系我们删除该应用。



配置应用资料

在**应用资料**区域,您可以编辑该应用的基本信息,包括**应用名称、描述**和**服务可用区**。

编辑基本信息

1. 单击修改按钮,进入应用设置的编辑状态。

2. 您可以修改应用名称、描述和服务可用区。

3. 单击**保存**。

配置基础信息

在基础信息区域,您可以获取该应用的密钥。

获取密钥

密钥信息为敏感信息,请妥善保密,谨防泄露。2019年8月15日前创建的应用(SDKAppID)默认使用区分公钥和私 钥的 ECDSA-SHA256 签名算法,您可以选择升级至 HMAC-SHA256 签名算法,升级过程不会影响现网业务。 1.单击密钥右侧的显示密钥。

2. 单击**复制**即可复制并储存密钥信息。 密钥可用于生成 UserSig,详细操作请参见 生成 UserSig。

管理离线推送证书

添加离线推送证书

添加 Android 证书

1. 在 Android 标签下选择目标平台, 单击 Add Certificate。

2. 在弹出的添加证书对话框,填写相关参数信息。



3. 单击确认保存配置。

添加 iOS 证书

1. 在 iOS 标签下, 单击 Add Certificate。

2. 在弹出的添加证书对话框,填写相关参数信息。

3. 单击确认保存配置。

编辑离线推送证书

1. 单击已有证书区域的编辑。

2. 在弹出的对话框中修改相关参数,单击确认保存配置。

删除离线推送证书

注意

删除证书会导致消息推送不可触达,且删除后数据不可恢复,请谨慎操作。

1. 单击已有证书区域的删除。

2. 在弹出的确认删除证书中,单击确认。



功能配置

最近更新时间:2025-06-12 16:49:27

登录与消息

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Login and Message,您可以根据实际业务需求管理登录与消息相关配置。

登录设置

1. 在**登录与消息**页面,单击**登录设置**右侧的编辑。

2. 在弹出的登录设置对话框中,选择多端登录类型,设置 Web 端以及其他平台实例同时在线数量。

说明

专业版、专业版Plus、企业版选择多平台登录时,Web端可同时在线个数最多为10个;Android、iPhone、iPad、Windows、Mac、Linux平台,每种平台可同时在线设备个数最多为3个。

3. 单击确定保存设置

历史消息存储时长配置

历史消息默认存储7天,**延长历史消息存储时长为增值服务**,具体计费说明请参见价格说明。每个自然月仅支持更改 一次配置。

1. 在**登录与消息**页面,单击历史消息存储时长配置右侧的编辑。

2. 在弹出的历史消息存储时长配置对话框中,设置延长历史消息存储时长。

3. 单击确定保存配置,该配置即刻生效。

消息撤回设置

1. 在**登录与消息**页面,单击**消息撤回设置**右侧的**编辑**。

2. 在弹出的消息撤回设置对话框中,设置消息可撤回时长。



3. 单击确定保存配置。

多端同步设置

在**登录与消息**页面,在多端同步设置区域可以选择开启或关闭删除会话后多端同步:

开启:如果多端同时在线,删除会话会同步至各端,同步删除会话。

关闭:如果多端同时在线,一端删除会话后不同步至其它端。默认为关闭。

说明

删除会话后多端同步仅支持**终端 SDK 5.1.1, Web SDK 2.14.0**及**以上版本**,低版本用户如需开启同步请**升级 SDK 版本**。

用户状态查询及状态变更通知配置

在**登录与消息**页面,在**用户状态查询及状态变更通知配置**区域可以开启用户状态查询及状态变更通知开关。 **说明**

用户状态查询及状态变更通知默认关闭,此时客户端查询/订阅/取消订阅用户状态会收到 72001 错误码。支持终端 SDK 6.3及以上版本开启。该功能仅支持专业版、专业版Plus、企业版使用,您可以 单击升级。

消息扩展设置

在**登录与消息**页面,在**消息扩展设置**区域可以开启功能开关。

说明

消息扩展可以为消息增加 key/value 状态标识,基于此功能可实现投票、接龙、问卷调查等功能,详情请参见 功能文档。该功能仅对专业版、专业版Plus、企业版客户开放,并且仅支持终端 SDK 增强版 6.7.3184 及以上版本,低版本用户请升级 SDK 版本。

全员推送设置

在**登录与消息**页面,在**全员推送设置**区域可以开启功能开关。

说明

全员推送,是用户运营利器,不仅支持全员发送特定内容,还可根据标签、属性,针对特定用户群体发送个性化内容,如会员活动、区域通知等,助力拉新、转化、促活等各个阶段运营工作的有效进行,详情请参见全员推送。



会话列表拉取个数配置

在**登录与消息**页面,在**会话列表拉取个数配置**区域可以配置从云端拉取的会话个数,客户端默认可从云端拉取100个 会话,可配置从云端拉取最多500个会话。

说明

会话列表拉取个数配置仅支持**专业版、专业版Plus、企业版**使用,非专业版、专业版**Plus、**企业版用户请升级后使用。

会话列表拉取个数配置仅支持终端 SDK 5.1.1, Web SDK 2.0及以上版本, 低版本用户如需配置请升级 SDK 版本。

黑名单检查

在**登录与消息**页面,在**黑名单检查**区域可以选择开启或关闭发送消息后展示发送成功:

启用:若对方已将您的账号加入黑名单,您向对方发送单聊消息时将提示消息发送成功,实际对方不会收到消息。 默认为启用。

关闭:若对方已将您的账号加入黑名单,您向对方发送单聊消息时将提示消息发送失败。

好友关系检查

在**登录与消息**页面,在**好友关系检查**区域可以选择开启或关闭发送单聊消息检查关系链:

开启:发起单聊时会先检查好友关系, 仅允许好友之间发送单聊消息, 陌生人发送单聊消息时 SDK 会收到 错误码 20009。

关闭:发起单聊时无需检查好友关系,允许用户和好友、陌生人之间发送单聊消息。默认为关闭。

接收被踢离线回调

在登录与消息页面,在接收被踢离线回调区域可以选择开启或关闭被踢离线回调,按钮分断网重连与主动登录: 同时启用断网重连时触发,主动登录时触发时,在任何情况下重连登录,均会收到错误码 6208。 仅启用断网重连时触发,不开启主动登录时触发时,在除主动登录的情况下,会收到错误码 6208。



注:此功能仅对 Native SDK 生效,Web SDK 不会返回6208。

好友与关系链

设置添加好友的验证方式、好友自定义字段。

设置添加好友验证方式

1. 登录即时通信 Chat 控制台,选择目标应用进入应用管理页面。

2. 在左侧导航栏选择 Chat > Configuration > Friends and Relationship, 在默认加好友验证方式设置功能卡片右 侧单击**编辑**。

3. 选择合适的方式单击确定进行保存。

好友自定义字段

说明

最多可添加20个好友自定义字段,该字段不支持修改字段名与字段类型且不支持删除,请根据业务需求合理规划该 字段。

1. 登录 即时通信 Chat 控制台,选择目标应用进入应用管理页面。

 2. 在左侧导航栏选择 Chat > Configuration > Friends and Relationship,单击好友自定义字段功能卡片右侧的新 增。

3. 在弹出的好友自定义字段对话框中, 输入自定义字段名, 设置字段类型。

说明

字段名必须是英文字母,且长度不能超过8个字符。

用户自定义字段

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Custom User Field,您可以根据实际业务需求管理用户自定义字段。

注意

最多可添加20个用户自定义字段,该字段不支持修改字段名与字段类型且不支持删除,请根据业务需求合理规划该 字段。



添加用户自定义字段

1. 在用户自定义字段页面,单击右侧新增按键。

2. 在弹出的用户自定义字段对话框中,输入自定义字段名,设置字段类型,设置读写权限。 说明

字段名必须是英文字母,且长度不能超过8个字符。 需至少设置一种读权限和一种写权限。

3. 单击确定保存设置

修改用户自定义字段权限

1. 在用户自定义字段页面,单击目标自定义字段所在行的修改权限。

2. 在弹出的用户自定义字段对话框中, 修改读写权限。

3. 单击确认保存配置。

4. 单击确定保存配置。

群组配置

群成员自定义字段

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Group configuration,您可以根据实际业务需求管理群成员自定义字段。

注意

最多可添加5个群成员自定义字段,该字段仅支持修改群组类型及其对应的读写权限且不支持删除,请根据业务需求 合理规划该字段。

添加群成员自定义字段

1. 在**群成员自定义字段**页面,右侧单击新增按键。

2. 在弹出的群成员维度自定义字段对话框中, 输入字段名称, 设置群组类型及其对应的读写权限。

说明

字段名称只能由字母、数字以及下划线(_)组成,不能以数字开头,且长度不能超过16个字符。

群成员自定义字段名称不允许与群自定义字段名称一致。

单击**添加群组类型**可新增一组群组类型参数,群组类型不可重复。



单击目标群组类型参数组所在行的**删除**可删除该组群组类型参数,需至少保留一组群组类型参数。

3. 勾选我已知道"群成员自定义字段"添加后,仅可修改已添加群组类型的读写权限;无法删除该字段,无法重新选择也无法删除已添加的群组类型。

4. 单击确定保存设置。

编辑群成员自定义字段

1. 在**群成员自定义字段**页面,单击目标群成员自定义字段所在行的编辑。

2. 在弹出的群成员维度自定义字段对话框中,修改已选群组类型的读写权限,或单击添加群组类型新增群组类型并 设置其参数,群组类型不可重复。

3. 勾选我已知道"群成员自定义字段"添加后,仅可修改已添加群组类型的读写权限;无法删除该字段,无法重新选 择也无法删除已添加的群组类型。

4. 单击确定保存设置。

群自定义字段

登录 即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Group configuration,选择 Custom Group Member Field 标签栏,您可以根据实际业务需求管理群自定义字段。 注意

最多可添加10个群自定义字段,该字段仅支持修改群组类型及其对应的读写权限且不支持删除,请根据业务需求合 理规划该字段。

添加群自定义字段

1. 在群自定义字段页面,单击添加群维度自定义字段。

2. 在弹出的群维度自定义字段对话框中, 输入字段名称, 设置群组类型及其对应的读写权限。

说明

字段名称只能由字母、数字以及下划线(_)组成,不能以数字开头,且长度不能超过16个字符。

群自定义字段名称不允许与群成员自定义字段名称一致。

单击添加群组类型可新增一组群组类型参数,群组类型不可重复。

单击目标群组类型参数组所在行的删除可删除该组群组类型参数,需至少保留一组群组类型参数。

3. 勾选我已知道"群成员自定义字段"添加后,仅可修改已添加群组类型的读写权限;无法删除该字段,无法重新选 择也无法删除已添加的群组类型。

4. 单击确定保存设置。



编辑群自定义字段

1. 在群自定义字段页面,单击目标群自定义字段所在行的编辑。

2. 在弹出的群维度自定义字段对话框中,修改已选群组类型的读写权限,或单击添加群组类型新增群组类型并设置 其参数,群组类型不可重复。

3. 勾选我已知道"群成员自定义字段"添加后,仅可修改已添加群组类型的读写权限;无法删除该字段,无法重新选择也无法删除已添加的群组类型。

4. 单击确定保存设置。

群消息配置

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Group configuration,选择 Group message configuration 标签栏,您可以根据实际业务需求进行群消息配置。

拉取入群前漫游消息

1. 在拉取入群前漫游消息页面,针对不同的群组类型,单击编辑。

2. 在弹出的拉取入群前漫游消息配置对话框中,选择业务所需的配置项。

说明

配置修改后约十分钟生效。请耐心等待。

直播群(AVChatRoom)不支持配置拉取入群前漫游消息。

群系统通知配置

登录 即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Group configuration,选择 Group system notification configuration 标签栏,您可以根据实际业务需求进行群系统通知配置。

群成员变更通知

1. 在群成员变更通知页面,针对不同的群组类型,单击编辑。

2. 在弹出的群成员变更通知对话框中,选择业务所需的配置项。

说明

配置修改后约十分钟生效。请耐心等待。 直播群(AVChatRoom)不支持配置群成员变更通知。



群资料变更通知

在群资料变更通知页面,针对不同的群组类型,单击编辑。
 在弹出的群资料变更通知对话框中,选择业务所需的配置项。
 说明

配置修改后约十分钟生效。请耐心等待。

直播群(AVChatRoom)不支持配置群资料变更通知。

群成员资料变更通知

1. 在群成员资料变更通知页面,针对不同的群组类型,单击编辑。

2. 在弹出的群成员资料变更通知对话框中,选择业务所需的配置项。

说明

配置修改后约十分钟生效。请耐心等待。

群功能配置

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Group configuration,选择 Group feature configuration 标签栏,您可以根据实际业务需求进行群功能配置。

社群

社群,是可容纳10万人的超大群组,创建后可随意进出,支持历史消息存储。开通社群开关默认关闭,开启后可创 建社群并使用社群相关功能。

如果需要使用话题功能,请在开通社群后,打开开通话题开关。同一社群下可创建多个话题,多个话题共享一套社群成员关系,但不同话题内独立收发消息、互不干扰。

说明

社群功能仅支持终端 SDK 5.8.1668增强版, Web SDK 2.17.0及以上版本使用,低版本用户如需使用社群功能请升级 SDK 版本。

该功能仅支持专业版、专业版Plus、企业版使用,您可以 单击升级。



直播群在线成员列表

直播群在线成员列表开关默认关闭,可开启。

说明

开启之后,直播群成员将保存最新入群并且在线的1000人,客户端可以拉取该列表。 该功能仅支持终端 SDK 6.3及以上版本使用,低版本用户如需使用请升级 SDK 版本。 该功能仅支持专业版、专业版Plus、企业版使用,您可以 单击升级。

直播群广播消息

是否开启直播群广播消息功能开关默认关闭,可开启。

说明

直播群广播消息功能默认关闭,支持终端 SDK 6.5及以上版本开启。 开启该功能后可配置直播群广播消息调用频率,默认1条/秒,最高可配置为5条/秒。 该功能仅支持专业版、专业版Plus、企业版使用,您可以 单击升级。

直播群封禁

开启后,可对直播群成员进行封禁。成员封禁后无法接收消息,且封禁时间内无法再进群。

说明

该功能仅支持终端SDK 6.6, Web SDK 2.22及以上版本使用,低版本用户如需使用请升级SDK版本。 该功能仅支持专业版、专业版Plus、企业版使用,您可以 单击升级。

直播群新成员查看入群前消息配置

直播群新成员查看入群前消息,提升直播间用户粘性的重要功能,可帮助用户了解"前情提要",迅速融入互动讨论中,让用户更有参与感,无论何时进入直播间都可获得高沉浸式体验,进而提升直播间用户停留时长。

1. 在**登录与消息**页面,单击**直播群新成员查看入群前消息配置**右侧的编辑。

2. 在弹出的**直播群新成员查看入群前消息配置**对话框中,设置新成员可查看的消息数。

3. 单击确定保存配置。

群消息已读回执配置

群消息已读回执,高效沟通必备功能。作为一种强反馈工具,支持查看已读/未读已发送消息的人数和成员,尤其在 商务、办公场景下,可帮助团队构建更加及时、高效的沟通氛围。

1. 在**登录与消息**页面,单击**群消息已读回执配置**右侧的编辑。

2. 在弹出的**群消息已读回执配置**对话框中,设置支持消息回执的群类型。

3. 单击确定保存配置。

说明

该功能**仅对专业版、专业版Plus、企业版用户开放**,非专业版、专业版Plus、企业版用户 请升级 后使用。群消息已 读回执功能支持**终端 SDK 6.1.2155及以上版本**,适用于群容量200以内的好友工作群(Work)、陌生人社交群 (Public)、临时会议群(Meeting)。



服务端 API 调频

登录 Chat 控制台,在左侧导航 Chat > 功能管理 > 服务端 API 调频,您可以根据实际业务需求进行服务端 API 调频 配置。

1. 单击需要调频的服务端 API 右侧的编辑。

2. 滑动滑杆至所需的频率大小,并单击保存。

3. 在弹出的保存当前配置对话框中,单击确认保存配置。

说明:

服务端 API 调频是收费服务仅对专业版 Plus 和企业版开放。每个调频项按当日配置的最高数值计算,仅针对超出默 认限频的部分收费,计费说明详见Chat 套餐外计费说明 - 服务端 API 调用频率。 支持调整的服务端 API:

功能说明	接口	默认限频	单个叠加包大 小
批量发单聊 消息	v4/openim/batchsendmsg	12000条/分 钟, 200次/秒	6000条/分钟, 可免费提升至 1000次/秒
单发单聊消 息	v4/openim/sendmsg	200条/秒	100条/秒
在群组中发 送普通消息	v4/group_open_http_svc/send_group_msg	200条/秒	100条/秒
在群组中发 送系统通知	v4/group_open_http_svc/send_group_system_notification	200条/秒	100条/秒
查询单聊未 读消息计数	v4/openim/get_c2c_unread_msg_num	200次/秒	100次/秒
增加群成员	v4/group_open_http_svc/add_group_member	200次/秒	100次/秒
删除群成员	v4/group_open_http_svc/delete_group_member	200次/秒	100次/秒
创建群组	v4/group_open_http_svc/create_group	200次/秒	100次/秒
解散群组	v4/group_open_http_svc/destroy_group	200次/秒	100次/秒
获取群详细 资料	v4/group_open_http_svc/get_group_info	200次/秒	100次/秒
获取群成员 详细资料	v4/group_open_http_svc/get_group_member_info	200次/秒	100次/秒
获取用户所	v4/group_open_http_svc/get_joined_group_list	200次/秒	100次/秒



加入的群组			
查询用户在 群组中的身 份	v4/group_open_http_svc/get_role_in_group	200次/秒	100次/秒
拉取群聊历 史消息	v4/group_open_http_svc/group_msg_get_simple	200次/秒	100次/秒
设置资料	v4/profile/portrait_set	200次/秒	100次/秒
拉取资料	v4/profile/portrait_get	200次/秒	100次/秒
查询账号在 线状态	v4/openim/query_online_status	200次/秒	100次/秒
拉取黑名单	v4/sns/black_list_get	200次/秒	100次/秒


账号管理

最近更新时间:2024-10-09 14:27:27

登录 即时通信 IM 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Users,您可以根据实际业务需求管理账号。

新建账号

- 1. 在账号管理页面, 单击新建账号。
- 2. 在弹出的新建账号对话框中, 配置以下参数:

Application Overview	Overview	Account Management Current data center: Singapore O Telegram group WhatsApp group
	Users	Create account Batch Import Batch Export The current version does not allow account deletion by default.
دي Call	Groups	Click here to remove the restriction
S Conference	Configuration 🗸	Username (UserID) Nickname Account Type ∀ Profile Photo Creation time Operation
(··) Live	Webhook Statistics	administrator Administrator 2024-09-24 Export Edit 14:33:26 Cancel Administrator
Chat	Push 🗸	Total items: 1 10 - v / page - v - 1 / 1 page - v
 In-game Voice Chat 	Plugins	Create account 🛞
	Monitor	Account O General Admin Type
	Dev Tools 🗸 🗸 🗸	Username * Enter User ID
	Integration Guide	Nickname Enter a nickname (optional)
		Profile Photo Enter the profile photo URL (optional)
		Confirm Cancel

账号类型:区分普通账号和管理员账号。"App管理员"是对 App 具有最高管理权限的角色,可调用 REST API 接口,进行创建/解散群组、发送全员推送消息等操作。每个应用最多支持配置 10 个管理员。

用户名:需输入用户名(UserID),必填。

用户昵称:可输入用户昵称,非必填。

头像:可输入用户头像链接 URL, 非必填。

3. 单击确定保存配置。

账号新建完成后,在账号列表中可以查看用户名、昵称、账号类型、头像和创建时间。



删除账号

1. 默认不允许删除账户。解除限制后可删除账户。点击 Click here to remove the restriction,在弹出的确认对话框中,了解相关风险后,点击确认解除限制。解除

Account Managemer	1t Current data center:	Singapore 🕕 Telegra	m group WhatsApp	o group	
Create account Batch Ir	mport Batch Export	The current version do	oes not allow account d	leletion by default.	Username (UserID)
Click here to remove the restriction	n				
Username (UserID)	Nickname	Account Type 🖓	Profile Photo	Creation time	Operation
1		General		2024-09-24 16:37:55	Export Edit Delete
administrator		Administrator	7	2024-09-24 14:33:26	Export Edit Cancel Administrator
Are you sure you wan application does not	t to lift the restriction t allow account deletion	that "the current 🛞 "?		10 🔻 / page 🛛 🕅	< 1 /1page ►
It is recommended that yo the restrictions. At the same time, please I	ou upgrade the SDK to v7.8 be aware of the risks after y	or higher before lifting our account is deleted:			
The account cannot be r	etrieved				
All user data cannot be	recovered				
Re-registration of the co	prresponding UserID is not a	allowed within 3 months			
Apps with SDK lower the accounts	an v7.8 may have abnormal	data on deleted			
If you still want to lift the account deletion", please	restriction of "the current ap click [Confirm].	oplication does not allow			
	Confirm Cancel)			

2. 在账号管理页面,在列表中选中需要删除的账号,单击批量删除。



Account Management Current data center	Singapore 🛈 Telegram group	WhatsApp group	
Create account Batch Delete Batch Import	Batch Export		Username (UserID) Q
Username (UserID) Nickname	Account Type 🍸 Profi	le Photo Creation time	Operation
☑ 1	General	2024-09-24 16:37:55	Export Edit Delete
administrator	Administrator	2024-09-24 14:33:26	Export Edit Cancel Administrator
Total items: 2		10 🔻 / page	I /1page ►

3. 在弹出的确认对话框中,了解相关风险后,点击确认删除账户。删除后数据无法恢复,请谨慎操作。



Acc	ount Managemen	t Current data cent	er: Singapore 🛈 🛛 Telegra	m group WhatsApp	p group		
C	eate account Batch De	elete Batch Impo	ort Batch Export			Username (UserID) Q	
	Username (UserID)	Nickname	Account Type 🏹	Profile Photo	Creation time	Operation	
	1		General		2024-09-24 16:37:55	Export Edit Delete	
	administrator		Administrator		2024-09-24 14:33:26	Export Edit Cancel Administrator	
То	Confirm deletion of th	e selected account?	\otimes		10 🔻 / page 🛛 🕅	▲ 1 /1page ▶	
	1 accounts selectedShow N	lore ∽					
	Before the official deletion the account is deleted:	operation, please be av	vare of the risk points after				
	Account cannot be retrieved	ved					
	All data such as the relati deleted and cannot be re	ionship chain and data u covered	nder the account will be				
	The UserID correspondin within three months	g to the account cannot	be created repeatedly				
	Apps whose integrated C data of deleted accounts	hat SDK is later than v7.	8 may display abnormal				
	If you still want to delete th	he selected account, clic	k OK.				
		Confirm Cancel	D				

编辑账号

在账号管理页面,在列表中希望编辑的账号操作列,单击编辑。
 在弹出的编辑账号对话框中,配置以下参数:
 用户名:不可编辑。
 昵称:可编辑用户昵称,非必填。
 账号类型:不可编辑。
 性别:可编辑性别,非必填。
 生日:可编辑生日,非必填。
 所在地:可编辑所在地,非必填。



个性签名:可编辑个性签名,非必填。 加好友验证方式:可编辑加好友验证方式。 语言:可编辑语言,非必填。 头像:可编辑头像,非必填。 消息设置:可选择消息设置选项,非必选。 管理员禁止加好友标识:可配置管理员禁止加好友标识,非必选。 等级:可编辑等级,非必填。 角色:可编辑角色,非必填。



3. 单击确定保存配置。

导出账号

1. 单独导出:在账号管理页面,在列表中希望导出的账号操作列,单击导出。



Account Manageme	nt Current data cer	nter: Singapore 🛈 🛛 Telegra	am group WhatsApp	o group	
Create account Batch Click here to remove the restriction	Import Batch Exp	The current version d	oes not allow account d	leletion by default.	Username (UserID)
Username (UserID)	Nickname	Account Type 🖓	Profile Photo	Creation time	Operation
administrator		Administrator		2024-09-24 14:33:26	Export Edit Cancel Administrator
Total items: 1				10 🔻 / page 🛛 🕅	I /1page ▶

2. 批量导出:在账号管理页面,在列表中选中需要导出的账号,单击批量导出。

Account Manageme	ent Current data cer	ter: Singapore 🛈 🛛 Telegra	am group WhatsApp) group	
Create account Batch Click here to remove the restrict	Import Batch Exp	ort The current version d	ioes not allow account d	eletion by default.	Username (UserID)
Username (UserID)	Nickname	Account Type 🏹	Profile Photo	Creation time	Operation
✓ administrator		Administrator		2024-09-24 14:33:26	Export Edit Cancel Administrator
Total items: 1				10 🔻 / page 🛛 🕅	I /1page ►

3. 在弹出的导出对话框中提示导出成功后,单击下载,即完成导出。







群组管理

最近更新时间:2025-03-10 15:00:58

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Groups,您可以根据实际业务需求管理群组。

您也可以通过调用对应的 REST API 管理群组,具体操作请参见 群组管理 API 文档。

添加群组

1. 在**群组管理**页面,单击**添加群组**。

2. 在弹出的添加群组对话框中, 配置以下参数:

群名称:请输入群组的名称,必填参数,长度不超过30字节。

群主 ID:请输入群主的 ID,选填参数,必须输入已注册的用户名。

群类型:请设置群组类型,支持好友工作群、陌生人社交群、临时会议群和直播群,群组的详细介绍请参见群组类型介绍。

Ø Application Overview	Overview	Group Management Current data center: Singapore ① Telegram group WhatsApp group
Advanced Features	Users	FAOr
Call Conference (+) Live Conference	Configuration Webhook Statistics	 PAQS Why can't I obtain the values of custom group/member fields? What are the differences between AVChatRoom and Meeting (ChatRoom in earlier versions) groups? How can I view muted members and their muting periods? What features does a community offer? What about use cases? How to enable it?
Chat In-game Voice Chat	Push v	Group Management Add Group Send Message Delete Group
	Monitor Dev Tools v Integration Guide	Group Name * Enter Group Name * Enter Group Enter (optional) Owner ID Group Type ★ Work Group ↓ Total ite Confirm

3. 单击确定保存配置。

群组创建完成后,在群组列表中可以查看群 ID、群名称、群主、类型和创建时间。



查看群组详情

在**群组管理**页面,单击目标群组所在行的 Learn more,进入**群组详情**页面,您可以查看并修改该群组的基本信息,可以管理该群组的群成员。

Add Group Send	Message Dele	te Group 🛛 🗘	Enter	group ID	(
Group ID	Group Name	Group Owner	Group Type 🏹	Creation time	Operatio
@		Not set	Work Group	2024-09-24 16:48:05	Learn more Send Message Delete
otal items: 1				10 🔻 / page 🛛 🕅 🔺	1 /1page >>

修改基本信息

- 1. 在**群组详情**页面,单击基本信息区域的编辑。
- 2. 在弹出的修改群信息的对话框中, 您可以修改群名称和群简介。



basic croc	p Information						E
Group ID							
Group Name	e test						
Group Type	Work Group)					
Group Introduction	•						
Group Owner							
Creation time	2024-09-2416	i:48:05					
Manage	Modify Group	Info	\otimes				
Add M	Group Name *	test		Ente			C
T (j) is	Group Introduction	Enter group introduction	0	ideo group without any mer bers by setting Batch mutir	nber, so the error code 1000 ng and unmuting or audio-v	7 will be returned if the ideo group member bi	operation anning.
		Up to 80 characters can be entered.					
Use		Confirm		Time of sending last message	the Member Role		Operation
				No data			

3. 单击确定保存配置。

管理群成员

添加群成员

1. 在**群组详情**页面,单击群成员管理区域的**添加群成员**。

2. 在弹出的添加成员对话框中, 输入用户名。

说明:

必须输入已注册的用户名。



Basic Group	Information					Ec
Group ID						
Group Name						
Group Type	Work Group					
Group ntroduction	-					
Group Owner	-					
Creation ime	2024-09-24 16:48:05					
Manage Gro Add Memb	Dup Member Delete Member		Enter Use	· ID		Q
Manage Gro Add Memb	bup Member Delete Member Iding or deleting group member operation i	s not applicable to an audio-vi	Enter User	· ID o the error code 10007 will be	returned if the operation	Q
Add Memb Add Memb	Dup Member Delete Member Iding or deleting group member operation i ormed on such a group. An admin can achie	s not applicable to an audio-vi ve the effect of deleting memb	Enter User ideo group without any member, s bers by setting Batch muting and	r ID o the error code 10007 will be unmuting or audio-video gro	ereturned if the operation oup member banning.	Q
Add Memb	bup Member Delete Member Iding or deleting group member operation i formed on such a group. An admin can achie ne Nickname	s not applicable to an audio-vi ve the effect of deleting memb Added Time	Enter User deo group without any member, s bers by setting Batch muting and Time of sending the Last message	ID o the error code 10007 will be unmuting or audio-video gro Member Role	returned if the operation oup member banning. Opera	Q
Add Memb	bup Member Delete Member Iding or deleting group member operation I formed on such a group. An admin can achie ne Nickname	s not applicable to an audio-vi ve the effect of deleting memb Added Time	Enter User deo group without any member, s bers by setting Batch muting and Time of sending the last message	r ID o the error code 10007 will be unmuting or audio-video gro Member Role	ereturned if the operation oup member banning. Opera	Q
Add Memb The ad is performance	bup Member Delete Member Iding or deleting group member operation i ormed on such a group. An admin can achie me Nickname	s not applicable to an audio-vi ve the effect of deleting memb Added Time	Enter User deo group without any member, s bers by setting Batch muting and Time of sending the last message	ID o the error code 10007 will be unmuting or audio-video gro Member Role	returned if the operation oup member banning. Opera	Q

3. 单击**确定**保存配置。群成员添加成功后,在群成员列表中可以查看用户名、昵称、加入时间、最后发言时间和成员身份。

删除群成员

在**群组详情**页面,您可以通过以下方式删除群成员:
 单个删除:单击目标群成员所在行的**删除**。
 批量删除:勾选所有待删除的目标群成员,单击群成员列表上方的**删除群成员**。
 在弹出的确认删除对话框中,单击确认。
 删除后,所选成员将不属于该群组。

发送消息

1. 在**群组管理**页面,您可以通过以下方式发送消息: 单发群消息:单击目标群组所在行的**发送消息**。



Group Ma	nagement	Current data center: Singapore 🛈	Telegram group	WhatsApp group	
FAQs					
 Why can^h What are How can 	t I obtain the values the differences bet I view muted mem	s of custom group/member fields? ween AVChatRoom and Meeting (Cha pers and their muting periods?	tRoom in earlier versio	ns) groups?	
▶ What fea	tures does a comm	unity offer? What about use cases? Ho	w to enable it?		
Group Ma	nagement up Send Mo Send Message	essage Delete Group 🗘	⊗ Group Ty	Enter group ID	Q Operation
@T	Are you sure you Message	want to send message to 1 group(s)?	Work Gro	up 2024-09-24 16:48:05	Learn more Send Message Delete
Total ite	content *	o / 300		10 ▼ / page 🛛 🕷 🔌	1 /1page ► ₩
Gro @Tr Total ite	Send Message Are you sure you Message content *	want to send message to 1 group(s)? Enter 0 / 300	S Group Tyr	De ♥ Creation time up 2024-09-24 16:48:05 10 ▼ / page ◄ ◀	Operation Learn more Send Message Delete

群发群消息:勾选所有待发送消息的目标群组,单击群组列表上方的**发送消息**。

Add Group Ser	nd Message Dele	te Group	Enter	group ID	(
Group ID	Group Name	Group Owner	Group Type 🍸	Creation time	Operatio
•	test	Not set	Work Group	2024-09-24 16:48:05	Learn more Send Message Delete
~	test2	Not set	Public Group	2024-09-24 17:06:51	Learn more Send Message Delete
Total items: 2				10 🔻 / page 🛛 🖛 🔺	1 /1page >

2. 在弹出的发送群消息对话框中, 输入消息内容。



说明:

消息长度不得超过300字。 3. 单击**确定**发送消息。

解散群组

解散后群组所有信息将被删除,且不能恢复,请谨慎操作。

1. 在群组管理页面, 您可以通过以下方式解散群组:

单个解散:单击目标群组所在行的解散。

批量解散:勾选所有待解散的目标群组,单击群组列表上方的解散群组。

			Enter	group ID	(
- Group ID	Group Name	Group Owner	Group Type 🍸	Creation time	Operatio
•	test	Not set	Work Group	2024-09-24 16:48:05	Learn more Send Message Delete
	test2	Not set	Public Group	2024-09-24 17:06:51	Learn more Send Message Delete

2. 在弹出的确认解散对话框中,单击确认。 解散后群组所有信息将被删除,且不能恢复。



回调配置

最近更新时间:2025-06-23 11:07:28

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Webhook configuration,您可以根据实际业务需求配置回调 URL 以及启用哪些回调。

配置回调 URL

1. 在回调配置页面,单击回调 URL 配置区域的编辑。

2. 在弹出的回调 URL 配置对话框中, 输入回调 URL。

说明:

新的回调地址在修改成功2分钟后生效。

回调 URL 必须以 http:// 或 https:// 开头。

若您暂未申请域名,可直接配置 IP,例如 http://123.123.123.123/imcallback 。

只能使用英文字母(a~z , 不区分大小写)、数字(0~9)以及连接符(-)。不支持使用空格及以下字符: ! \$&?等。

连接符(-)不能连续出现、不能单独注册,也不能放在开头和结尾。

域名长度不超过63个字符。

回调 URL中 Chat 默认带80/443端口,更换回调 URL 时涉及端口变化,需避免更换前后的端口互为前缀,比如避免 把 https://xxx:443 更改为 https://xxx:4433 或者 https://xxx 更改为 https://xxx:4433。

3. 单击**确定**保存配置。

配置事件回调

1. 在回调配置页面,单击第三方回调配置区域的编辑。在弹出的编辑弹窗部分选择要配置的功能。

说明:

部分回调功能需前往控制台开启对应功能后才能触发,具体请参见回调事件开启操作说明。 2. 单击**确定**保存配置。

后续操作

配置回调 URL 并启用相应的事件回调后,您可以参考 回调配置 使用相应的回调功能,实时了解用户信息和操作。

回调事件开启操作说明



直播群成员在线状态回调

若需要开启直播群成员在线状态回调,需首先在功能配置里打开群成员在线状态开关。

用户资料变更后回调

若需要开启用户资料变更后回调,需首先在功能配置里打开用户资料变更订阅开关。

在线状态回调

若需要开启在线状态回调,需首先在功能配置里打开用户状态查询及状态变更通知配置开关。



用量统计

最近更新时间:2025-05-19 17:05:39

即时通信 Chat 控制台为您提供了数据统计和分析功能,您可以查看该应用的用户规模、消息活跃、群组规模以及实时监控等相关数据。

说明:

正常情况下,每天上午10:00左右更新数据,如遇数据为0或未及时更新,请先检查该 SDKAppID 在统计段内是否产生相关数据(例如,是否有新注册用户等),若已产生相关数据但未及时更新请耐心等待。

每日数据统计

登录即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Statistics。

用户规模

1. 在每日数据统计页面,选择用户规模页签。

2. 在总览区域, 您可以查看以下相关数据:

本月峰值 DAU:截至昨日该 SDKAppID 的当月峰值 DAU 数据,每月1日该数据为0。

截至昨日累计用户数:截至昨日该 SDKAppID 累计注册的 UserID 数。

昨日新注册用户数:昨日该 SDKAppID 新注册的 UserID 数。

3. 选择7天、14天、30天或指定时间段。

4. 在数据趋势区域,您可以查看所选时间段内 DAU和新增或累计注册的趋势图。

5. 在数据详情区域,您可以查看所选时间段内每一天的详细数据,包括 DAU、DAU-日环比、累计用户数、累计用户数-日环比、新注册用户数和新注册用户数-日环比。您也可以单击**导出CSV文件**导出数据表。

消息活跃

1. 在每日数据统计页面,选择消息活跃页签。

2. 在总览区域, 您可以查看以下相关数据:

昨日单聊消息量:昨日该 SDKAppID 下 C2C 聊天上行消息总条数。

昨日群消息量:昨日该SDKAppID下,私有群、公开群和聊天室的聊天上行消息总条数。

3. 选择7天、14天、30天或指定时间段。

4. 在数据趋势区域,您可以查看所选时间段内C2C或普通群的消息量趋势图。

5. 在数据详情区域,您可以查看所选时间段内每一天的详细数据,包括消息量 (条)、消息量-日环比、发消息用户数、发消息用户数-日环比、离线推送量和离线推送量-日环比。您也可以单击**导出CSV文件**导出数据表。



群组规模

1. 在每日数据统计页面,选择群组规模页签。

2. 在总览区域,您可以查看**本月峰值群组数**相关数据,即截至昨日该 SDKAppID 的当月峰值群组数据,每月1日该数 据为0。

3. 选择7天、14天、30天或指定时间段。

4. 在数据趋势区域,选择数据项,您可以查看所选时间段内新建、累计或活跃的群组数趋势图。

5. 在数据详情区域,您可以查看所选时间段内每一天的详细数据,包括新建群数、新建群数-日环比、有消息活跃的 群数、有消息活跃的群数-日环比、峰值群组和峰值群组-日环比。您也可以单击**导出CSV文件**导出数据表。

实时监控

说明:

实时监控功能全线公测中,我们还在不断迭代更新,可以提交工单给我们反馈建议和意见。

1. 在左侧导航栏,选择 Chat > Monitor。

2. 在总览区域, 您可以**查看当前在线用户数、今日单聊消息量、今日普通群消息量、今日直播群消息量**。

3. 在详细监控数据区域,时间轴默认展示自然日24小时数据。鼠标指向数据图表区域时,通过滚动滚轮可以放大时间轴查看细节,左右拖动时间轴可以查看前后时间的数据,单击时间轴下方图例可以隐藏或展示图表中对应数值。 在登录情况监控区域,您可以查看各个端的登录次数和登录成功率。

说明:

目前仅支持展示4.8.10或以上的 iOS、Android、Windows 及 Mac 终端上报登录数据,建议您升级至 最新版 SDK。

在消息情况监控区域,您可以查看各个端发送单聊/群聊消息的次数和成功率。

说明:

目前仅支持展示4.8.10或以上的 iOS、Android、Windows 及 Mac 终端上报登录数据,建议您升级至 最新版 SDK。 Web端暂不支持按聊天类型统计消息量。

在回调情况监控区域,您可以查看回调次数和成功率。

在 REST API 调用情况监控区域,您可以查看 REST API 请求数和请求成功率。

在离线推送情况监控区域,您可以查看离线推送次数和推送成功率。



资源包查看指南

最近更新时间:2025-06-09 16:47:12

社群消息量资源包查看

针对目前 Chat 包月套餐包赠送 的社群下行消息量,可前往 即时通信 Chat 控制台,选择目标应用进入应用管理页面,在左侧导航栏选择 Chat > Configuration > Group Configuration,在 Group feature configuration > Community 下进行查看。

插件资源包查看

针对云端搜索、语音转文字等插件开通后,赠送的用量查看,可前往即时通信 Chat 控制台,左侧导航栏选择 Chat > Advanced Features,点击任意插件卡片,进入插件详情进行查看。



实时监控

最近更新时间:2025-05-19 17:06:10

实时监控

说明:

实时监控功能全线公测中,我们还在不断迭代更新,可以提交工单给我们反馈建议和意见。

1. 在左侧导航栏,选择 Chat > Monitor。

2. 在总览区域, 您可以**查看当前在线用户数、今日单聊消息量、今日普通群消息量、今日直播群消息量**。

3. 在详细监控数据区域,时间轴默认展示自然日24小时数据。鼠标指向数据图表区域时,通过滚动滚轮可以放大时间轴查看细节,左右拖动时间轴可以查看前后时间的数据,单击时间轴下方图例可以隐藏或展示图表中对应数值。 在登录情况监控区域,您可以查看各个端的登录次数和登录成功率。

说明:

目前仅支持展示4.8.10或以上的 iOS、Android、Windows 及 Mac 终端上报登录数据,建议您升级至 最新版 SDK。

在消息情况监控区域,您可以查看各个端发送单聊/群聊消息的次数和成功率。

说明:

目前仅支持展示4.8.10或以上的 iOS、Android、Windows 及 Mac 终端上报登录数据,建议您升级至 最新版 SDK。 Web端暂不支持按聊天类型统计消息量。

在回调情况监控区域,您可以查看回调次数和成功率。

在 REST API 调用情况监控区域,您可以查看 REST API 请求数和请求成功率。

在离线推送情况监控区域,您可以查看离线推送次数和推送成功率。



开发辅助工具

最近更新时间:2025-03-10 15:00:58

离线推送自查

离线推送定位工具

您可以使用该工具自助查询收不到离线消息的问题。

1. 登录即时通信 Chat 控制台,选择目标应用进入应用管理页面。

2. 在左侧导航栏选择 Chat > Push > Access test。

3. 在**离线推送定位工具**区域,输入 UserID。

4. 单击获取设备状态,查看该 UserID 目前已经上报的证书 ID、设备 Token 等信息。

说明

若该 UserID 未上报任何证书 ID、设备 Token 等信息,则无法进行下一步。

5. 选择该 UserID 上报的任意一个证书 ID, 单击开始检测, 查看发送结果。

若提示成功推送,说明您在控制台填写的证书信息无误、调用 SDK 接口上报 Token 成功。您可以通过 用户客户端 状态检查工具 进一步排查。

若提示失败,您可以查看具体的失败原因以及解决方案。



When an exception occurs when you connect to the native manufacturer's offline push, you can use this tool to troubleshoot.						
Username (UserID)	Enter					
	Get token binding status					
Results						
Certificate ID	Please select ~					
	Send a push test					
Results						
(i) Access to the te	esting tool. Currently, it only supports testing whether the					
push is sent suc the Token is suc	ccessfully and calling the SDK interface to report whether ccessful. It does not currently support testing jumps,					

用户状态检查工具

您可以使用该工具自动获取用户客户端状态,检查用户是否处于可接收离线推送状态。

1. 登录 即时通信 Chat 控制台,选择目标应用进入应用管理页面。

2. 在左侧导航栏选择 Chat > Push > Access test。

3. 在**用户状态检查工具**区域,输入 UserID。



4. 单击获取状态,您可以查看该 UserID 当前的状态、登录的客户端类型等信息。

若提示初步判定该 UserID 当前可以接收离线推送,您可以在其他设备上登录其他 UserID 作为发送方,给当前 UserID 发送单聊文本消息,检查是否能收到。

This tool is used to init state that can receive	tially determine whether the current user client is in a offline push.				
Username (UserID)	Enter				
	Get Status				
Results					
 Users cannot re 	ceive offline push when they are not logged in and the				
current status is process on the o	current status is Offline. Users can switch to the background or kill the				
receive it, , please check the token binding status.					

UserSig 生成&校验

签名(UserSig)生成工具

系统将会自动获取当前应用的密钥,您只需要填写用户名(UserID)即可使用该工具快速生成签名(UserSig)用于本地跑通 Demo 以及功能调试。如需用于正式业务,请采用 服务端计算 UserSig 方式。

- 1. 登录 即时通信 Chat 控制台 > Development Tools。
- 2. 在签名(UserSig)生成工具区域,输入用户名。
- 3. 单击**生成签名(UserSig)**即可生成签名,签名有效期默认为180天。



4. 单击复制签名(UserSig)即可粘贴保存签名。

Cverview	← UserSig Tools	
Applications		To unid on its international second state was information and optical
Usage Statistics	Too haven uprovided a payment medido. We will sospend the service for your account after you use up your in	ree resources. To avoid service interroption, piease comprete your information and refresh.
 ⊘ Data Monitoring 	Gimmetrine (HearCie) Conservator	Cianature (LaseCie) Varifier
	Signature (Usersig) Generator	Signature (Osersig) vermer
📋 Package Management 🗸	This tool can quickly generate a UserSig, which can be used to run through demos and to debug features.	This tool is used to verify the validity of the UserSig you use.
Relevant Services ^	Application (SDKAppID) Username (UserID) ①	Application (SDKAppID) Username (UserID) ①
Beauty AR	-first_app v user	Select an application v Set the user name
A Development Tools ^	SDKSecretKey	SDKSecretKey
UserSig Tools	06854¢	Auto-generated after you select an application
RTMP Address Generator	Generate	UserSig
		Please enter
	Generate result	
	elyrVi klmiZ kbmL uvu4:	Verify

签名(UserSig)校验工具

系统将会自动获取当前应用的密钥,您只需要填写 UserID 和 UserSig 即可使用该工具快速校验 UserSig 的有效性。 1. 登录 即时通信 Chat 控制台,选择目标应用进入应用管理页面。

2. 在左侧导航栏选择 Chat > Auxiliary Tools > UserSig Tools。

3. 在签名(UserSig)校验工具区域,输入UserID和UserSig。



4. 单击开始校验,查看校验结果信息。

腾讯云

若校验成功,您可以查看该 UserSig 对应的 SDKAppID、UserID、生成时间、有效期和过期时间。

Verify result

腾讯云

Verify result: Verified successfully Generate ticket parameters: SDKAppid: Identifier: user Creation time: Sep 24, 2024 17:33:47 (UTC+08:00) Validity: 180day 0hr 0min 0sec Expiration time: Mar 23, 2025 17:33:47 (UTC+08:00)

若校验失败,您可以查看具体的失败原因以及解决方案。

Verify result

Verify result: Failed to verify Failure reason: UserID not match Solution: make sure UserID in request match to the one in UserSig

自助排障日志

即时通信 Chat 控制台为您提供了自助排障的功能,助力开发者高效定位、解决问题,您可以查询即时通信 Chat 三天内后台日志信息。

1. 登录 即时通信 Chat 控制台 , 单击目标应用卡片。

2. 在左侧导航栏选择辅助工具 > 自助排障日志。

3. 配置以下查询条件,检索日志信息。

事件名:可选择待查询事件,非必填。

用户ID:可输入用户名(UserID),其中发消息为发送方 UserID,非必填。

接收方/群 ID:可输入目标会话 ID,单聊为接收方 UserID,群聊为 GroupID,非必填。

错误码:可输入错误码,错误码对应描述请参见错误码,非必填。

时间范围:选择日志时间范围,最多可查询最近三天的日志信息,必填。

4. 点击查询,检索当前条件下的日志信息。



访问管理 授予子账号操作控制台权限

最近更新时间:2025-03-10 15:48:42

简介

本文将为您介绍两种授权方式快速解决下述问题,详细操作步骤如下。如果您希望进行更多复杂权限配置策略,请参见访问管理 > 自定义策略。

子帐号使用即时通信 Chat 服务时,需要主帐号授予 访问控制台 以及进行配置操作的权限,否则控制台应用列表将 无法显示应用,如下图所示:

	×
Auth	orization is required for the current operation, please add the following permissions to continue operation
	How to create a permission policy based on permissionless information? 🛂
o per	mission information details:
o per	mission information details:
o per 1 2	mission information details: you are not authorized to perform operation (im:DescribeIMAppList) resource (gcs::im::uin/1
o per 1 2	mission information details: you are not authorized to perform operation (im:DescribelMAppList) resource (qcs::im::uin/1sdkappid/*) has no permission

子帐号有标签访问授权,但当前控制台应用标签和子帐号标签权限不符时,子帐号无法查看新建应用。

方案一、全局授权操作步骤



步骤1:进入授权

使用主帐号,进入控制台访问管理>用户列表,单击子用户左侧的授权按钮,将弹出"关联策略"选择框。

Cloud Access Management	User List						
Dashboard Users ^		How to view more information? CAM protects the security of your subscription. You can also dick the	sensitive information. You can click	the drop-down button [▶] on the left side of	f the list to view more information about t	he user, such as identity security stat	us, groups the user has joined, and message
User List User Settings		Create User More				Search by username/	D/SecretId/mobile/email/remarks (se
User Groups Policies		Username	User Type	Account ID	Creation Date	Associated Info	Action
Roles		Tencent_testtttttttta	Root Account		2019-06-24 13:21:02	L 🖸	Authorize More 🔻
Identity Providers Access Key *		dms_test_sub01	Sub-user		2020-10-10 16:06:11]. 🖸	Authorize More 🔻
		► re_billing_test	Sub-user		2020-09-04 17:27:30	G	Authorize More 🔻

步骤2:选择策略

在策略筛选框,搜索"即时通信",勾选需要授权的选项,单击确定即完成授权。



Associate Policy

		•	Q		Policy Name	Policy Type
Policy Name	Policy Type 🔻					
QcloudAVCFullAccess Full read-write access to Instant Messagin	Preset Policy		•			
QcloudIMAGEFullAccess Full read-write access to Image	Preset Policy			÷		
QcloudIMFGFullAccess Full read-write access to Intelligent Manuf	Preset Policy					
QcloudIMReadOnlyAccess Read-only access to Instant Messaging (IM)	Preset Policy					
QcloudAccessForGSERoleInIMAGEUse	Dreset Doliny		•			

说明

读写访问权限:既能访问控制台又可以修改配置。 **只读访问权限**:只能访问控制台,不可以做其他操作。

步骤3:完成授权

右上角提示"关联策略成功"即完成关联操作。



Cloud Access	User List				Vie	w API Inspector 🗙	
Management Dashboard							Policy associ
Users ^		How to view more information? CAM protects the security of your sen: subscription. You can also click the use	itive information. You can click the mame to view or edit the user info	e drop-down button [▶] on the left side o prmation.	f the list to view more information about th	e user, such as identity security status,	groups the user has joined, and message
User List User Settings		Create User More *				Search by username/ID/	'Secretid/mobile/email/remarks (se Q
User Groups		- Username	User Type	Account ID	Creation Date	Associated Info	Action
Policies							
Roles		Tencent_testtttttttttta	Root Account		2019-06-24 13:21:02	10 M	Authorize More 🕈
Identity Providers Access Key *		dms_test_sub01	Sub-user		2020-10-10 16:06:11	. 🖸	Authorize More 🔻

方案二、按标签授权操作步骤

本方案针对需要通过标签授权管理子帐号的客户,子帐号仅能访问及操作授权标签下的应用。

注意

为子帐号分配标签策略后,子帐号无法访问及操作标签为空的应用。由于子帐号在 Chat 控制台 新建应用标签为空,因此需要主帐号更改该应用标签为已授权标签,子帐号方能使用。

如需将已有应用按标签授权给子帐号,请确保将要授权的应用已配置相应标签,否则将无法通过标签进行授权。 如应用未配置标签请前往即时通信 Chat 控制台中 基本配置 页面进行配置,详情请参见 标签配置。

或进入标签列表将应用批量绑定到标签下,详情请参照绑定资源。

步骤1:进入授权

使用主帐号,进入控制台 访问管理 >策略,单击顶部新建自定义,将弹出"选择创建策略方式"弹框。

Sencent Cloud	Overview	Products - CDN +			8 <mark>99+</mark>
Cloud Access Management	Policy				
Dashboard					
Users *		① Associate users or user groups with policies to grant permissions.			
User Groups					
Dolicios		Create Custom Policy Delete	All Policies	Preset Policy	Custom
PUILIES					



步骤2:选择标签

ľ

选择"按标签授权",进入"标签策略生成器"。

5	Select Pol	licy Creation Method	×
	C	Create by Policy Generator Select service and actions from the list to auto-generate policy syntax	>
		Create by Policy Syntax Write policy syntax to generate policies	>
		Authorize by Tag Grant permissions of resources under certain tags to users or user groups	>

步骤3:生成策略

在"标签策略生成器"填入需要授权的子帐号、标签等信息,单击**下一步**进入检查页面。



Cloud Access Management	← Autł	orize by Tag	
Dashboard		_	
Users ~		1 Tag Policy	Generator > 2 Check and Finish
User Groups			
Policies		Authorized Users	Please select
Roles		User Groups	Please select
Identity Providers		Tag Keys 🛈	
Access Key 👻		2.7.0	Please select
		Tag Values 🛈	Please select
		Resources	Manage Permission
		Next	

说明

如标签选择列表为空,则需要主帐号先去标签控制台新建标签。

Tag			View API Inspector 🗙
Resource Tag	lags: All tag keys		
Tag List	Invite		
	inquite		
	Create Delete		
	Tag key	Tag value	Operation
	tke-name	prom-aeq4ld7a/grafana	Delete View bound resource

步骤4:完成授权



检查无误,单击完成,结束标签授权流程。

Cloud Access Management	thorize by Tag
Dashboard	
Users *	Check and Finish
User Groups	
Policies	Policy Name * policygen-20210205095508
Roles	Authorized Users Tencent_testtttttttta
Identity Providers	Authorized User Groups test2
Access Key *	Policy Content
	2 "version": "2.0", 3 "statement": [
	4 { 5 "effect": "allow",
	6 "action": "*",
	8 "condition": {
	9 "for_any_value:string_equal": { 10 "qcs:tag": [
	11 "tke-lb-serviceuuid&4663ccfc-bb6f-4492-93b7-5c7c345fb311"
	14 } 15 }
	Back Done



预设策略

最近更新时间:2025-03-10 15:49:00

注意:

本文档主要介绍___即时通信 Chat___ 访问管理功能的相关内容,其他产品访问管理相关内容请参见 支持 CAM 的产品。

Chat 访问管理实质上是将子账号与策略进行绑定,或者说将策略授予子账号。开发者可以在控制台上直接使用预设 策略来实现一些简单的授权操作,复杂的授权操作请参见自定义策略。

Chat 目前提供了以下预设策略:

策略名称	策略描述
QcloudAVCFullAccess	Chat 全读写访问权限
QcloudIMReadOnlyAccess	Chat 只读访问权限

预设策略使用示例

新建拥有即时通信权限的子账号

1. 以腾讯云 主账号 的身份访问 CAM 控制台的用户列表, 单击新建用户。

2. 在"新建用户"页面选择自定义创建,进入"新建子用户"页面。

说明:

请根据 CAM 自定义创建子用户 的操作指引完成"设置用户权限"之前的步骤。

3. 在"设置用户权限"页面:

1. 搜索并勾选预设策略 Instant Messaging 。

2. 单击**下一步**。

3. 在"审阅信息和权限"分栏下单击**完成**,完成子用户的创建,在成功页面下载并保管好该子用户的登录链接和安全凭证,其中包含的信息如下表:

信息	来源	作用	是否必须保存	
登录链接	在页面中复制	方便登录控制台,省略填写主账号的步骤	否	
用户名	安全凭证 CSV 文件	登录控制台时填写	是	
密码	安全凭证 CSV 文件	登录控制台时填写	是	
SecretId	安全凭证 CSV 文件	调用服务端 API 时使用,详见 访问密钥	是	
SecretKey	安全凭证 CSV 文件	调用服务端 API 时使用,详见 访问密钥	是	



4. 将上述登录链接和安全凭证提供给被授权方,后者即可使用该子用户对 Chat 做所有操作,包括访问 Chat 控制 台、请求 Chat 服务端 API 等。

将即时通信权限授予已存在的子账号

1. 以腾讯云 主账号 的身份访问 CAM 控制台的用户列表,单击想要进行授权的子账号。

2. 单击"用户详情"页面权限栏的**添加策略**,如果子账号的权限非空,则单击关联策略。

3. 选择**从策略列表中选取策略关联**,搜索并勾选预设策略 Instant Messaging 。后续按页面提示完成授权流程 即可。

解除子账号的即时通信权限

1. 以腾讯云 主账号 的身份访问 CAM 控制台的用户列表, 单击想要解除授权的子账号。

2. 在"用户详情"页面权限栏找到预设策略 Instant Messaging ,单击右侧的**解除**。按页面提示完成解除授权流程即可。



自定义策略

最近更新时间:2025-03-10 15:49:00

注意:

本文档主要介绍___即时通信 Chat___ 访问管理功能的相关内容,其他产品访问管理相关内容请参见 支持 CAM 的产品。

在 Chat 访问管理中使用 预设策略 来实现授权虽然方便,但权限控制粒度较粗,不能细化到 Chat 应用和 云 API 粒 度。如果开发者要求精细的权限控制能力,则需要创建自定义策略。

自定义策略创建方法

自定义策略有多种创建方法,下方表格展示各种方法的对比,具体操作流程请参考下文。

创建入口	创建方法	效力 (Effect)	资源 (Resource)	操作 (Action)	灵活 性	难 度
CAM 控制台	策略生成器	手动选择	语法描述	手动选择	中	中
CAM 控制台	策略语法	语法描述	语法描述	语法描述	高	盲
CAM 服务端 API	CreatePolicy	语法描述	语法描述	语法描述	高	盲

说明:

Chat ___ 不支持___ 按产品功能或项目来创建自定义策略。

___手动选择___指用户在控制台所展示的候选项列表中选择对象。

___语法描述___指通过 授权策略语法 来描述对象。

授权策略语法

资源语法描述

如上文所述, Chat 权限管理的资源粒度是应用。应用的策略语法描述方式遵循 CAM 资源描述方式。在下文的示例中,开发者的主账号 ID 是12345678,开发者创建了三个应用:SDKAppID 分别是1400000000、1400000001和 1400000002。

即时通信 Chat 所有应用的策略语法描述

```
"resource": [
"qcs::im::uin/12345678:sdkappid/*"
```



]

单个应用的策略语法描述

```
"resource": [
"qcs::im::uin/12345678:sdkappid/140000001"
]
```

多个应用的策略语法描述

```
"resource": [
"qcs::im::uin/12345678:sdkappid/140000000",
"qcs::im::uin/12345678:sdkappid/140000001"
]
```

操作语法描述

如上文所述,实时音视频权限管理的操作粒度是云 API。在下文的示例中,以 DescribeAppStatList (获取应 用列表)、 DescribeSdkAppInfo (获取应用信息)等云 API 为例。 即时通信 Chat 所有云 API 的策略语法描述

```
"action": [
"name/im:*"
]
```

单个云 API 操作的策略语法描述

```
"action": [
"name/im:DescribeAppStatList"
]
```

多个云 API 操作的策略语法描述

```
"action": [
"name/im:DescribeAppStatList",
"name/im:DescribeTrtcAppAndAccountInfo"
]
```

自定义策略使用示例

使用策略生成器

在下文示例中,我们将创建一个自定义策略。该策略允许对1400000001这个即时通信 Chat 应用进行任何操作。 1. 以腾讯云 主账号 的身份访问 CAM 控制台的策略,单击新建自定义策略。


2. 选择按策略生成器创建,进入策略创建页面。

3. 选择服务和操作。

效果(Effect)配置项选择允许。

服务(Service)配置项选择即时通信 Chat。

操作(Action)配置项勾选所有项。

资源(Resource**)**配置项按照 资源语法描述 说明填写 qcs::im::uin/12345678:sdkappid/1400000001 。

条件(Condition)配置项无需配置。

单击添加声明,页面最下方会出现一条"允许对即时通信 Chat 应用1400000001进行任何操作"的声明。

4. 在同个页面中继续添加另一条声明。

效果(Effect)配置项选择拒绝。

服务(Service)配置项选择即时通信 Chat。

操作(Action)配置项勾选 RemoveUser (可通过搜索功能快速查找)。

资源(Resource)配置项按照 资源语法描述 说明填写 qcs::im::uin/12345678:sdkappid/1400000001 。 **条件(Condition)**配置项无需配置。

单击**添加声明**,页面最下方会出现一条"拒绝对即时通信 Chat 应用1400000001进行 RemoveUser 操作"的声明。 5.单击**下一步**,按需修改策略名称(也可以不修改)。

6. 单击完成完成自定义策略的创建。

后续将该策略授予其他子账号的方法同 将 Chat 全读写访问权限授予已存在的子账号。

使用策略语法

在下文示例中,我们将创建一个自定义策略。该策略允许对1400000001和1400000002这两个即时通信 Chat 应用进 行任何操作。

1. 以腾讯云 主账号 的身份访问 CAM 控制台的策略,单击新建自定义策略。

2. 选择按策略语法创建,进入策略创建页面。

3. 在**选择模板类型**框下选择空白模板。

说明:

策略模板指新策略是现有策略(预置策略或自定义策略)的一个拷贝,然后在此基础上做调整。在实际使用中,开 发者可以根据情况选择合适的策略模板,降低编写策略内容的难度和工作量。

4. 单击**下一步**,按需修改策略名称(也可以不修改)。

5. 在编辑策略内容编辑框中填写策略内容。本示例的策略内容为:

```
{
    "version": "2.0",
    "statement": [
    {
        "effect": "allow",
        "action": [
            "name/im:*"
    ],
        "resource": [
```



```
"qcs::im::uin/12345678:sdkappid/140000001",
    "qcs::im::uin/12345678:sdkappid/140000002"
]
},
{
    "effect": "deny",
    "action": [
        "name/im:RemoveUser"
],
    "resource": [
        "qcs::im::uin/12345678:sdkappid/140000001"
]
}
```

说明:

策略内容需遵循 CAM 策略语法逻辑,其中资源和操作两个元素的语法请参见上文 资源语法描述 和 操作语法描述 所述。

6. 单击创建策略完成自定义策略的创建。

后续将该策略授予其他子账号的方法同将 Chat 全读写访问权限授予已存在的子账号。

使用 CAM 提供的服务端 API

对于大多数开发者来说,在控制台完成权限管理操作已经能满足业务需求。但如果需要将权限管理能力自动化和系统化,则可以基于服务端 API 来实现。

策略相关的服务端 API 属于 CAM,具体请参见 CAM 官网文档。此处仅列出几个主要接口:

创建策略

删除策略

绑定策略到用户

解除绑定到用户的策略



高级功能 云端搜索功能 效果展示

最近更新时间:2024-04-23 15:40:55

功能体验







使用场景

场景一:全局搜索

通过消息搜索功能,用户可全局搜索所有包含指定关键词的会话,还可在搜索栏中自由选择搜索的时间范围。用户 单击搜索到的会话,可以直接定位至对应聊天位置,并高亮显示该消息。操作演示如下:





场景二:指定会话内搜索文件/图片/视频/其他

通过消息搜索功能,用户可直接搜索指定会话内的全部文件/图片/视频/其他消息,还可在指定的时间段内搜索。用户 点击搜索到的文件/图片/视频/其他,同样可以直接定位至对应聊天位置,并高亮显示该文件/图片/视频/其他消息。以 下演示以文件为例:





集成指引

Web&小程序&uni-app:集成指引

Android&iOS&Windows&Mac:集成指引



集成指引 Android&iOS&Windows&Mac

最近更新时间:2025-03-27 11:23:30

功能描述

消息云端搜索,提升 App 使用体验必备功能,可以帮助用户从纷繁复杂的信息中直接找到预期内容,快捷方便;也可以作为运营工具,增加相关内容的引导,简洁高效。

说明:

消息云端搜索功能仅 7.3.4358 及以上版本支持。

云端搜索功能仅对专业版 Plus 和企业版客户开放,购买专业版 Plus 和企业版 后可使用;体验版支持一定额度免费 试用,有效期一个月。

如果您没有开通该服务,调用接口会返回 60020 错误码。

消息搜索类介绍

消息搜索参数类

消息搜索参数类为 V2TIMMessageSearchParam (Android / iOS & Mac / Windows)。搜索消息时, SDK 会按 照该对象的设置,执行不同的搜索逻辑。

V2TIMMessageSearchParam 的参数说明如下:

参数	含义	说明	
keywordList	关键字列表	最多支持5个。当消息发送者以及消息类型均未指定时, 必须设置关键字列表;否则,关键字列表可以为空。	
keywordListMatchType	指定关键字列表匹 配类型	可设置为"或"关系搜索,或"与"关系搜索。取值分别为 V2TIM_KEYWORD_LIST_MATCH_TYPE_OR 和 V2TIM_KEYWORD_LIST_MATCH_TYPE_AND 。默认为 "或"关系搜索。	
senderUserIDList	指定 userID 发送 的消息	最多支持5个。	
messageTypeList	指定搜索的消息类 型集合	传空表示搜索支持的全部类型消息(V2TIMFaceElem 和 V2TIMGroupTipsElem 不支持搜索)。其他类型取 值参考 V2TIMElemType (Android / iOS & Mac / Windows)。	



conversationID	搜索 "全部会话" 还是搜索 "指定的 会话"	conversationID 为空,搜索全部会 话; conversationID 不为空,搜索指定会话。
searchTimePosition	搜索的起始时间点	默认为0(从现在开始搜索)。UTC时间戳,单位:秒。
searchTimePeriod	从起始时间点开始 默认为 0 (不限制时间范围)。24 x 60 x 60 代表过去的过去时间范围 天,单位:秒。	
searchCount	搜索的数量	搜索的数量,最大支持100。
searchCursor	搜索的游标	搜索的起始位置,第一次填写空字符串,续拉时填写上一次 返回的 V2TIMMessageSearchResult 中的 searchCursor。

消息搜索结果类

消息搜索结果类为 V2TIMMess	ageSearchResult	(Android / iOS & Mac / Windows)。参数说明如下:	
参数	含义	说明	
totalCount	搜索结果总数	如果搜索指定会话,返回满足搜索条件的消息总数; 如果搜索全部会话,返回满足搜索条件的消息所在的所 有会话总数量。	
messageSearchResultItems	搜索结果列表	如果搜索指定会话,返回结果列表只包含该会话结果; 如果搜索全部会话,会对满足搜索条件的消息根据会话 ID 分组,分页返回分组结果。	
searchCursor 续拉的游标		调用搜索接口续拉时需要填的游标	

其中 messageSearchResultItems 是个列表,内含 V2TIMMessageSearchResultItem (Android/iOS & Mac/Windows)对象,参数说明如下:

参数	含义	说明
conversationID	会话 ID	-
messageCount	消息数量	当前会话一共搜索到了多少条符合要求的消息。
messageList	满足搜索条件的消 息列表	如果搜索指定会话, messageList 中装载的是本会话 中所有满足搜索条件的消息列表。 如果搜索全部会话, messageList 中装载的消息条数 会有如下两种可能: 如果某个会话中匹配到的消息条数 > 1, 则 messageList 为空, 您可以在 UI 上显示



"{ messageCount }条相关记录"。
如果某个会话中匹配到的消息条数 = 1,则
messageList 为匹配到的那条消息,您可以在 UI 上显
示之,并高亮匹配关键词。

搜索全部会话的消息

当用户在搜索框输入关键字搜索消息时,您可以调用 searchCloudMessages (Android / iOS & Mac /

Windows)搜索消息。

如果您希望在全部会话范围内搜索,只需要将 V2TIMMessageSearchParam 中的 conversationID 设置为 空或者不设置即可。

示例代码如下:

Android

iOS & Mac

Windows

```
List<String> keywordList = new ArrayList<>();
keywordList.add("abc");
keywordList.add("123");
V2TIMMessageSearchParam searchParam = new V2TIMMessageSearchParam();
// 设置搜索关键字
searchParam.setKeywordList(keywordList);
// 搜索20条数据
searchParam.setSearchCount(20);
// 从最新的会话开始搜索
searchParam.setSearchCursor("");
// 从当前时间开始搜索
searchParam.setSearchTimePosition(0);
// 搜索10分钟之内的消息
searchParam.setSearchTimePeriod(600);
V2TIMManager.getMessageManager().searchCloudMessages(searchParam,newV2TIMValueCallb
    @Override
   public void onSuccess(V2TIMMessageSearchResult v2TIMMessageSearchResult) {
     // 搜索成功
    }
    @Override
   public void onError(int code, String desc) {
     // 搜索失败
    }
});
```



```
V2TIMMessageSearchParam *param = [[V2TIMMessageSearchParam alloc] init];
// 设置搜索关键字
param.keywordList = @[@"abc", @"123"];
param.messageTypeList = nil;
param.conversationID = nil;
param.searchTimePosition = 0;
param.searchTimePeriod = 0;
// 搜索20条数据
param.searchCount = 20;
// 从最新的会话开始搜索
param.searchCursor = @"";
[V2TIMManager.sharedInstance searchCloudMessages:param
succ:^(V2TIMMessageSearchResult *searchResult) {
    // 搜索成功, searchResult 中返回搜索结果
} fail:^(int code, NSString *desc) {
   // 搜索失败
}];
template <class T>
class ValueCallback final : public V2TIMValueCallback<T> {
public:
    using SuccessCallback = std::function<void(const T&)>;
    using ErrorCallback = std::function<void(int, const V2TIMString&)>;
   ValueCallback() = default;
    ~ValueCallback() override = default;
   void SetCallback(SuccessCallback success_callback, ErrorCallback error_callback
     success_callback_ = std::move(success_callback);
      error_callback_ = std::move(error_callback);
    }
    void OnSuccess(const T& value) override {
     if (success_callback_) {
        success_callback_(value);
      }
   void OnError(int error_code, const V2TIMString& error_message) override {
     if (error_callback_) {
        error_callback_(error_code, error_message);
      }
    }
private:
    SuccessCallback success_callback_;
    ErrorCallback error_callback_;
```



```
};
V2TIMMessageSearchParam searchParam;
// 设置搜索关键字
searchParam.keywordList.PushBack("abc");
searchParam.keywordList.PushBack("123");
// 搜索20条数据
searchParam.searchCount = 20;
// 从最新的会话开始搜索
searchParam.searchCursor = "";
auto callback = new ValueCallback<V2TIMMessageSearchResult>{};
callback->SetCallback(
    [=](const V2TIMMessageSearchResult& messageSearchResult) {
     // 搜索成功
     delete callback;
    },
    [=](int error_code, const V2TIMString& error_message) {
     // 搜索失败
     delete callback;
    });
V2TIMManager::GetInstance()->GetMessageManager()->SearchCloudMessages(searchParam,
```

搜索指定会话的消息

```
当用户在搜索框输入关键字搜索消息时,您可以调用 searchCloudMessages (Android / iOS & Mac /
Windows)搜索消息。
示例代码如下:
Android
iOS & Mac
Windows
 List<String> keywordList = new ArrayList<>();
 keywordList.add("abc");
 keywordList.add("123");
 V2TIMMessageSearchParam searchParam = new V2TIMMessageSearchParam();
 // 搜索与用户 user1 的单聊消息
 searchParam.setConversationID("c2c_user1");
 // 设置搜索关键字
 searchParam.setKeywordList(keywordList);
 // 搜索20条数据
 searchParam.setSearchCount(20);
```



```
// 从最新的会话开始搜索
searchParam.setSearchCursor("");
// 从当前时间开始搜索
searchParam.setSearchTimePosition(0);
// 搜索10分钟之内的消息
searchParam.setSearchTimePeriod(600);
V2TIMManager.getMessageManager().searchCloudMessages(searchParam,newV2TIMValueCallb
    @Override
   public void onSuccess(V2TIMMessageSearchResult v2TIMMessageSearchResult) {
     // 搜索成功
   }
   QOverride
   public void onError(int code, String desc) {
     // 搜索失败
   l
});
V2TIMMessageSearchParam *param = [[V2TIMMessageSearchParam alloc] init];
// 设置搜索关键字
param.keywordList = @[@"abc", @"123"];
param.messageTypeList = nil;
// 搜索与用户 user1 的单聊消息
param.conversationID = @"c2c_user1";
param.searchTimePosition = 0;
param.searchTimePeriod = 0;
// 搜索20条数据
param.searchCount = 20;
// 从最新的会话开始搜索
param.searchCursor = @"";
[V2TIMManager.sharedInstance searchCloudMessages:param
succ:^(V2TIMMessageSearchResult *searchResult) {
   // 搜索成功, searchResult 中返回搜索结果
} fail:^(int code, NSString *desc) {
   // 搜索失败
}];
template <class T>
class ValueCallback final : public V2TIMValueCallback<T> {
public:
   using SuccessCallback = std::function<void(const T&)>;
   using ErrorCallback = std::function<void(int, const V2TIMString&)>;
   ValueCallback() = default;
   ~ValueCallback() override = default;
```



```
void SetCallback(SuccessCallback success_callback, ErrorCallback error_callback
     success_callback_ = std::move(success_callback);
     error_callback_ = std::move(error_callback);
    }
   void OnSuccess(const T& value) override {
     if (success_callback_) {
       success callback (value);
     }
    }
   void OnError(int error_code, const V2TIMString& error_message) override {
     if (error_callback_) {
       error_callback_(error_code, error_message);
     }
    }
private:
   SuccessCallback success_callback_;
    ErrorCallback error_callback_;
};
V2TIMMessageSearchParam searchParam;
// 搜索与用户 user1 的单聊消息
searchParam.conversationID = "c2c_user1";
// 设置搜索关键字
searchParam.keywordList.PushBack("abc");
searchParam.keywordList.PushBack("123");
// 搜索20条数据
searchParam.searchCount = 20;
// 从最新的会话开始搜索
searchParam.searchCursor = "";
auto callback = new ValueCallback<V2TIMMessageSearchResult>{};
callback->SetCallback(
    [=](const V2TIMMessageSearchResult& messageSearchResult) {
     // 搜索成功
     delete callback;
    },
    [=](int error_code, const V2TIMString& error_message) {
     // 搜索失败
     delete callback;
    });
V2TIMManager::GetInstance()->GetMessageManager()->SearchCloudMessages(searchParam,
```



搜索典型场景示例

普通的 IM 聊天软件, 搜索界面的展示通常分这几种场景:

图 1: 搜索聊天记录	图 2: 搜索更多聊天记录	图 3:搜索指定会话的消息

下文我们将依次向您展示如何利用 IM SDK 的搜索 API 实现上图的典型场景。

展示最近几个活跃的会话

```
如图1所示,最下方是搜索到的消息所属的最近3个会话列表,实现方式如下:
1. 设置搜索参数 V2TIMMessageSearchParam
conversationID 设置为 null , 表示搜索所有会话的消息。
searchCursor 设置为"",表示搜索最新的数据。
searchCount 设置为3,表示返回最近的会话数量,UI上一般显示3条。
2. 处理搜索回调结果 V2TIMMessageSearchResult
totalCount 表示匹配到的消息所属的所有会话数量。
messageSearchResultItems 列表为最近3(即入参 searchCount )个会话信息。其中元素
V2TIMMessageSearchResultItem 的 messageCount 表示当前会话搜索到的消息总数量;
搜索到的消息条数 > 1,则 messageList 为空,您可以在 UI 上显示 "4 条相关聊天记录",其中的 4 为
messageCount 。
搜索到的消息条数 = 1,则 messageList 为匹配到的那条消息,您可以在 UI 上显示消息内容并高亮搜索关键
词,例如搜索典型场景示例图中的"test"。
示例代码如下:
Android
iOS & Mac
Windows
 List<String> keywordList = new ArrayList<>();
 keywordList.add("test");
 V2TIMMessageSearchParam v2TIMMessageSearchParam = new V2TIMMessageSearchParam();
 // conversationID 设置为 null 表示搜索所有会话中的消息,结果会按照会话分类
 v2TIMMessageSearchParam.setConversationID(null);
 v2TIMMessageSearchParam.setKeywordList(keywordList);
 v2TIMMessageSearchParam.setSearchCursor("");
 v2TIMMessageSearchParam.setSearchCount(3);
 V2TIMManager.getMessageManager().searchCloudMessages(v2TIMMessageSearchParam, newV2
```



```
@Override
   public void onSuccess(V2TIMMessageSearchResult v2TIMMessageSearchResult) {
     // 匹配到的消息所属的所有会话数量
     int totalCount = v2TIMMessageSearchResult.getTotalCount();
     // 最近3个根据消息会话分类的信息
     List<V2TIMMessageSearchResultItem> resultItemList = v2TIMMessageSearchResult.
     for (V2TIMMessageSearchResultItem resultItem : resultItemList) {
       // 会话 ID
       String conversationID = resultItem.getConversationID();
       // 该会话匹配到的所有消息数量
       int totalMessageCount = resultItem.getMessageCount();
       // 消息列表:如果 totalMessageCount > 1, 该列表为空;如果 totalMessageCount = 1,
       List<V2TIMMessage> v2TIMMessageList = resultItem.getMessageList();
     }
    }
   @Override
   public void onError(int code, String desc) {
});
V2TIMMessageSearchParam *param = [[V2TIMMessageSearchParam alloc] init];
param.keywordList = @[@"test"];
// conversationID 设置为 nil 表示搜索所有会话中的消息,结果会按照会话分类
param.conversationID = nil;
param.searchCursor = @"";
param.searchCount = 3;
[V2TIMManager.sharedInstance searchCloudMessages:param succ:^(V2TIMMessageSearchRes
   // 匹配到的消息所属的所有会话数量
   NSInteger totalCount = searchResult.totalCount;
   // 最近3个根据消息会话分类的信息
   NSArray<V2TIMMessageSearchResultItem *> *messageSearchResultItems = searchResul
   for (V2TIMMessageSearchResultItem *searchItem in messageSearchResultItems) {
     // 会话 ID
     NSString *conversationID = searchItem.conversationID;
     // 该会话匹配到的所有消息数量
     NSUInteger messageCount = searchItem.messageCount;
     // 消息列表
     NSArray<V2TIMMessage *> *messageList = searchItem.messageList ?: @[];
} fail:^(int code, NSString *desc) {
   // fail
}];
template <class T>
class ValueCallback final : public V2TIMValueCallback<T> {
```



```
public:
    using SuccessCallback = std::function<void(const T&)>;
    using ErrorCallback = std::function<void(int, const V2TIMString&)>;
   ValueCallback() = default;
    ~ValueCallback() override = default;
    void SetCallback(SuccessCallback success callback, ErrorCallback error callback
     success_callback_ = std::move(success_callback);
      error callback = std::move(error callback);
    }
    void OnSuccess(const T& value) override {
     if (success_callback_) {
     success_callback_(value);
     }
    }
   void OnError(int error_code, const V2TIMString& error_message) override {
     if (error_callback_) {
     error_callback_(error_code, error_message);
     }
    }
private:
    SuccessCallback success_callback_;
    ErrorCallback error_callback_;
};
V2TIMMessageSearchParam searchParam;
// conversationID 设置为空表示搜索所有会话中的消息,结果会按照会话分类
searchParam.conversationID = "";
searchParam.keywordList.PushBack("test");
searchParam.searchCursor = "";
searchParam.searchCount = 3;
auto callback = new ValueCallback<V2TIMMessageSearchResult>{};
callback->SetCallback(
    [=] (const V2TIMMessageSearchResult& messageSearchResult) {
      // 匹配到的消息所属的所有会话数量
      uint32_t totalCount = messageSearchResult.totalCount;
      // 最近3个根据消息会话分类的信息
     V2TIMMessageSearchResultItemVector messageSearchResultItems =
     messageSearchResult.messageSearchResultItems;
      for (size_t i = 0; i < messageSearchResultItems.Size(); ++i) {</pre>
       // 会话 ID
       V2TIMString conversationID = messageSearchResultItems[i].conversationID;
       // 该会话匹配到的所有消息数量
```



```
uint32_t messageCount = messageSearchResultItems[i].messageCount;
    // 消息列表:如果 messageCount > 1, 该列表为空;如果 messageCount = 1, 该列表元素为
    V2TIMMessageVector messageList = messageSearchResultItems[i].messageList;
    }
    delete callback;
    },
    [=](int error_code, const V2TIMString& error_message) {
        // 搜索失败
        delete callback;
});
```

```
V2TIMManager::GetInstance()->GetMessageManager()->SearchCloudMessages(searchParam,
```

展示所有搜索到的消息所属会话列表

点击搜索典型场景示例图 1 中的"更多聊天记录", 会跳转到图 2, 展示所有搜索到的消息所属的会话列表。搜索参数 和搜索结果描述跟上文的场景类似。

为了防止内存膨胀,我们强烈建议您分页加载会话列表。举个例子,分页加载,每页展示10条会话结果,搜索参数 V2TIMMessageSearchParam 可以参考如下设置:

```
1. 首次调用:设置参数 searchCount = 10, searchCursor = ""。调用 searchCloudMessages 获取消息搜索结果,解析并展示到首页,并且从结果回调中获取会话总数量 totalCount 以及下次请求的游标 searchCursor 。
```

```
2. 当界面滑动快到底部后根据上一次请求结果中的游标 searchCursor 继续拉取下一页的数据。
```

示例代码如下:

Android

iOS & Mac

Windows

```
// 记录搜索游标
String searchCursor = "";
.....
private void searchConversation(String cursor) {
   List<String> keywordList = new ArrayList<>();
   keywordList.add("test");
   V2TIMMessageSearchParam v2TIMMessageSearchParam = new V2TIMMessageSearchParam()
   v2TIMMessageSearchParam.setConversationID(null);
   v2TIMMessageSearchParam.setKeywordList(keywordList);
   v2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.netSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.netSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.netSearchCursor(cursor);
   V2TIMMessageSearchParam.netSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.netSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchCursor(cursor);
   V2TIMMessageSearchParam.setSearchParam.setSearchCursor(cursor);
   V2TIMessageSearchParam.setSearchParam.setSearchParam
```



```
// 匹配到的消息所属的所有会话数量
       int totalCount = v2TIMMessageSearchResult.getTotalCount();
       // 下一页的游标
       searchCursor = v2TIMMessageSearchResult.getSearchCursor();
       // 该页的根据消息会话分类的信息
       List<V2TIMMessageSearchResultItem> resultItemList = v2TIMMessageSearchResul
       for (V2TIMMessageSearchResultItem resultItem : resultItemList) {
         // 会话 ID
         String conversationID = resultItem.getConversationID();
         // 该会话匹配到的所有消息数量
         int totalMessageCount = resultItem.getMessageCount();
         // 消息列表:如果 totalMessageCount > 1, 该列表为空;如果 totalMessageCount = 1
         List<V2TIMMessage> v2TIMMessageList = resultItem.getMessageList();
        }
     }
     @Override
     public void onError(int code, String desc) {
     }
   });
  }
// 当需要加载下一页时
public void loadMore() {
   searchConversation(searchCursor);
}
. . . . . .
// 记录搜索游标
NSString *searchCursor = @"";
. . . . . .
- (void) searchConversation: (NSString *) cursor {
   V2TIMMessageSearchParam *param = [[V2TIMMessageSearchParam alloc] init];
   param.keywordList = @[@"test"];
   param.conversationID = nil;
   param.searchCursor = cursor;
   param.searchCount = 10;
    [V2TIMManager.sharedInstance searchCloudMessages:param succ:^ (V2TIMMessageSearc
     // 匹配到的消息所属的所有会话数量
     NSUInteger totalCount = searchResult.totalCount;
     // 下一页的游标
     searchCursor = searchResult.searchCursor;
     // 该页的根据消息会话分类的信息
     NSArray<V2TIMMessageSearchResultItem *> *messageSearchResultItems = searchRes
     for (V2TIMMessageSearchResultItem *searchItem in messageSearchResultItems) {
```



```
// 会话 ID
       NSString *conversationID = searchItem.conversationID;
       // 该会话匹配到的所有消息数量
       NSUInteger totalMessageCount = searchItem.messageCount;
       // 消息列表:如果 totalMessageCount > 1, 该列表为空;如果 totalMessageCount = 1,
       NSArray<V2TIMMessage *> *messageList = searchItem.messageList ?: @[];
     }
    } fail:^(int code, NSString *desc) {
    // fail
   }];
}
// 当需要加载下一页时
- (void)loadMore {
    [self searchConversation:searchCursor];
}
template <class T>
class ValueCallback final : public V2TIMValueCallback<T> {
public:
    using SuccessCallback = std::function<void(const T&)>;
   using ErrorCallback = std::function<void(int, const V2TIMString&)>;
   ValueCallback() = default;
   ~ValueCallback() override = default;
   void SetCallback(SuccessCallback success_callback, ErrorCallback error_callback
     success_callback_ = std::move(success_callback);
     error_callback_ = std::move(error_callback);
    }
   void OnSuccess(const T& value) override {
   if (success_callback_) {
     success_callback_(value);
     }
    }
   void OnError(int error_code, const V2TIMString& error_message) override {
    if (error_callback_) {
     error_callback_(error_code, error_message);
     }
    }
private:
   SuccessCallback success_callback_;
   ErrorCallback error_callback_;
};
```



```
// 记录搜索游标
V2TIMString searchCursor = "";
void SearchConversation(V2TIMString cursor) {
   V2TIMMessageSearchParam searchParam;
   searchParam.keywordList.PushBack("test");
    searchParam.searchCursor = cursor;
    searchParam.searchCount = 10;
   auto callback = new ValueCallback<V2TIMMessageSearchResult>{};
     callback->SetCallback(
      [=](const V2TIMMessageSearchResult& messageSearchResult) {
       // 匹配到的消息所属的所有会话数量
       uint32_t totalCount = messageSearchResult.totalCount;
       // 下一页的游标
       searchCursor = messageSearchResult.searchCursor;
       // 该页的根据消息会话分类的信息
       V2TIMMessageSearchResultItemVector messageSearchResultItems =
       messageSearchResult.messageSearchResultItems;
       for (size_t i = 0; i < messageSearchResultItems.Size(); ++i) {</pre>
         // 会话 ID
         V2TIMString conversationID = messageSearchResultItems[i].conversationID;
         // 该会话匹配到的所有消息数量
         uint32_t messageCount = messageSearchResultItems[i].messageCount;
         // 消息列表:如果 messageCount > 1, 该列表为空;如果 messageCount = 1, 该列表元素
         V2TIMMessageVector messageList = messageSearchResultItems[i].messageList;
        }
       delete callback;
     },
      [=](int error_code, const V2TIMString& error_message) {
     // 搜索失败
     delete callback;
    });
   V2TIMManager::GetInstance()->GetMessageManager()->SearchLocalMessages(searchPar
}
// 当需要加载下一页时
void LoadMore() { SearchConversation(searchCursor); }
```

展示搜索指定会话的消息

与搜索典型场景示例图 2 展示会话列表不同的是,图 3 所示在指定会话中搜索到的消息列表。为了防止内存膨胀, 我们强烈建议您分页加载消息。举个例子,分页加载,每页展示 10 条消息结果:

1. 搜索参数 V2TIMMessageSearchParam 可以参考如下设置:



```
设置搜索参数 V2TIMMessageSearchParam 的 conversationID 为搜索的会话 ID。
首次调用:设置参数 searchCount = 10, searchCursor = ""。调用 searchCloudMessages 获取消息
搜索结果,解析并展示到首页,并且从结果回调中获取会话总数量 totalCount 以及下一页的游标
searchCursor 。
再次调用:更新参数 searchCursor 的值为上一步调用结果中的返回值。
2. 处理搜索结果 V2TIMMessageSearchResult :
totalCount 表示该会话匹配到的所有消息数量。
messageSearchResultItems 列表只有该会话的结果。列表中的元素
V2TIMMessageSearchResultItem 的 messageCount 为该分页的消息数量, messageList 为该分页
的消息列表。
searchCursor 表示下一页搜索的起始游标。
Android
iOS & Mac
Windows
  . . . . . .
 // 记录搜索游标
 String searchCursor = "";
 . . . . . .
 private void searchMessage(String cursor) {
     List<String> keywordList = new ArrayList<>();
     keywordList.add("test");
     V2TIMMessageSearchParam v2TIMMessageSearchParam = new V2TIMMessageSearchParam()
     v2TIMMessageSearchParam.setConversationID(conversationID);
     v2TIMMessageSearchParam.setKeywordList(keywordList);
     v2TIMMessageSearchParam.setSearchCount(10);
     v2TIMMessageSearchParam.setSearcuCursor(cursor);
     V2TIMManager.getMessageManager().searchCloudMessages(v2TIMMessageSearchParam,ne
       @Override
       public void onSuccess(V2TIMMessageSearchResult v2TIMMessageSearchResult) {
       // 该会话匹配到的所有消息数量
       int totalMessageCount = v2TIMMessageSearchResult.getTotalCount();
       // 下一页的游标
       searchCursor = v2TIMMessageSearchResult.getSearchCursor();
       // 该页消息信息
       List<V2TIMMessageSearchResultItem> resultItemList = v2TIMMessageSearchResult.
       for (V2TIMMessageSearchResultItem resultItem : resultItemList) {
         // 会话 ID
         String conversationID = resultItem.getConversationID();
         // 该页的消息数量
         int totalMessageCount = resultItem.getMessageCount();
         // 该页的消息数据列表
```



```
List<V2TIMMessage> v2TIMMessageList = resultItem.getMessageList();
     }
    }
     QOverride
     public void onError(int code, String desc) {
     }
   });
}
// 当需要加载下一页时
public void loadMore() {
    searchMessage(searchCursor);
}
. . . . . .
// 记录搜索游标
NSString *searchCursor = @"";
. . . . . .
- (void) searchMessage: (NSString *) cursor {
 V2TIMMessageSearchParam *param = [[V2TIMMessageSearchParam alloc] init];
 param.keywordList = @[@"test"];
  // conversationID 是要搜索的会话 ID
 param.conversationID = conversationID;
 param.searchCursor = cursor;
 param.searchCount = 10;
  [V2TIMManager.sharedInstance searchCloudMessages:param succ:^(V2TIMMessageSearchR
    // 该会话匹配到的所有消息数量
   NSUInteger totalMessageCount = searchResult.totalCount;
   // 下一页的游标
    searchCursor = searchResult.searchCursor;
    // 该页消息信息
   NSArray<V2TIMMessageSearchResultItem *> *messageSearchResultItems = searchResul
    for (V2TIMMessageSearchResultItem *searchItem in messageSearchResultItems) {
     // 会话 ID
     NSString *conversationID = searchItem.conversationID;
     // 该页的消息数量
     NSUInteger totalMessageCount = searchItem.messageCount;
     // 该页的消息数据列表
     NSArray<V2TIMMessage *> *messageList = searchItem.messageList ?: @[];
     }
    } fail:^(int code, NSString *desc) {
     // fail
    }];
}
```



```
// 当需要加载下一页时
- (void)loadMore {
    [self searchMessage:searchCursor];
}
template <class T>
class ValueCallback final : public V2TIMValueCallback<T> {
public:
    using SuccessCallback = std::function<void(const T&)>;
   using ErrorCallback = std::function<void(int, const V2TIMString&)>;
   ValueCallback() = default;
    ~ValueCallback() override = default;
   void SetCallback(SuccessCallback success_callback, ErrorCallback error_callback
     success_callback_ = std::move(success_callback);
      error_callback_ = std::move(error_callback);
    }
   void OnSuccess(const T& value) override {
      if (success_callback_) {
       success_callback_(value);
      }
   void OnError(int error_code, const V2TIMString& error_message) override {
     if (error_callback_) {
       error_callback_(error_code, error_message);
      }
    }
private:
    SuccessCallback success_callback_;
   ErrorCallback error_callback_;
};
// 记录搜索游标
V2TIMString searchCursor = "";
void SearchConversation(V2TIMString cursor) {
   V2TIMMessageSearchParam searchParam;
    searchParam.conversationID = conversationID;
    searchParam.keywordList.PushBack("test");
    searchParam.searchCursor = cursor;
    searchParam.searchCount = 10;
   auto callback = new ValueCallback<V2TIMMessageSearchResult>{};
```



```
callback->SetCallback(
      [=](const V2TIMMessageSearchResult& messageSearchResult) {
       // 该会话匹配到的所有消息数量
       uint32_t totalCount = messageSearchResult.totalCount;
       // 下一页的游标
       searchCursor = messageSearchResult.searchCursor;
       // 该页消息信息
       V2TIMMessageSearchResultItemVector messageSearchResultItems =
       messageSearchResult.messageSearchResultItems;
       for (size t i = 0; i < messageSearchResultItems.Size(); ++i) {</pre>
         // 会话 ID
         V2TIMString conversationID = messageSearchResultItems[i].conversationID;
         // 该页的消息数量
         uint32_t messageCount = messageSearchResultItems[i].messageCount;
         // 该页的消息数据列表
         V2TIMMessageVector messageList = messageSearchResultItems[i].messageList;
        }
       delete callback;
    },
    [=] (int error_code, const V2TIMString& error_message) {
       // 搜索失败
       delete callback;
   });
   V2TIMManager::GetInstance()->GetMessageManager()->SearchCloudMessages(searchPar
}
// 当需要加载下一页时
void LoadMore() { SearchConversation(SearchCursor); }
```

搜索自定义消息

通常情况下,如果您使用接口 createCustomMessage(data) (Android/iOS & Mac/Windows)创建自定义 消息,该消息无法被搜到,因为 SDK 将该自定义消息保存为二进制数据流。 如果您希望自定义消息可以被搜到,需要使用接口 createCustomMessage(data, description, extension) (Android/iOS & Mac/Windows)来创建并发送自定义消息,把需要搜索的文本放到 description 参数中。 如果您配置了离线推送功能,设置参数 description 后,自定义消息也会有离线推送且通知栏展示该参数内 容。如果不需要离线推送可以用发消息接口 sendMessage (Android/iOS & Mac/Windows)的参数 V2TIMOfflinePushInfo 中的 disablePush 来控制。 如果推送的通知栏内容不想展示为被搜索的文本,可以用参数 V2TIMOfflinePushInfo 中的 desc 来另外设 置推送内容。



搜索富媒体消息

富媒体消息包含文件、图片、语音、视频消息。

对于文件消息,界面通常显示文件名。如果调用 createFileMessage 创建文件消息时传入 fileName 参数, fileName 会作为文件消息被搜索的内容,与搜索关键词进行匹配。如果未设置 fileName , SDK 则会自动从 filePath 提取文件名作为搜索内容。 fileName 和 filePath 信息会保存到本地和服务器,换设备 拉取相关信息后均可搜索。

对于图片、语音、视频消息,并没有类似 fileName 这种名称,界面通常显示缩略图或时长,此时指定 keywordList 搜索无效。如果您希望搜索出此类消息,可以指定 messageTypeList 为

V2TIM_ELEM_TYPE_IMAGE / V2TIM_ELEM_TYPE_SOUND / V2TIM_ELEM_TYPE_VIDEO 做分类搜索,此时 会搜索出所有指定类型的消息。

交流与反馈

加入Telegram 技术交流群组或 WhatsApp 交流群,享有专业工程师的支持,解决您的难题。



Web & uni-app & 小程序

最近更新时间:2025-03-27 11:23:30

功能体验

含 UI 集成

快速集成消息云端搜索

Web&H5 Vue2&Vue3 Uniapp Vue2&Vue3

步骤1: 集成TUIKit

@tencentcloud/chat-uikit-vue ≥ 2.0.0,如未集成,请务必先根据 Vue2 & Vue3 TUIKit 快速集成指引进行集成。

步骤2:控制台开通云端搜索插件

注意:

每个插件限免费试用 1 次,有效期 7 天,试用结束后将停服,请提前购买。试用时,仅支持搜索开通云端搜索功能 后产生的消息内容,不支持历史消息搜索;购买插件后,将自动同步历史消息,支持历史消息搜索。 云端搜索功能仅对**专业版 Plus 和企业版**客户开放,购买专业版 Plus 和企业版 后可使用;体验版支持一定额度免费 试用,有效期一个月。

步骤3:搜索您的第一条消息

在完成 Vue2 & Vue3 TUIKit 快速集成指引 - 步骤6:发送您的第一条消息后,搜索您刚才发送的消息。

步骤1: 集成TUIKit

@tencentcloud/chat-uikit-uniapp ≥ 2.0.6,如未集成,请请务必先根据 uniapp TUIKit 快速集成指引进行集成。

步骤2:控制台开通云端搜索插件

注意:



每个插件限免费试用 1 次, 有效期 7 天, 试用结束后将停服, 请提前购买。试用时, 仅支持搜索开通云端搜索功能 后产生的消息内容, 不支持历史消息搜索; 购买插件后, 将自动同步历史消息, 支持历史消息搜索。

步骤3:搜索你的第一条消息

在完成 Uniapp TUIKit 快速集成指引 - 步骤6: 发送您的第一条消息后, 搜索您刚才发送的消息。

独立引入消息云端搜索

说明:

以上 快速集成消息云端搜索 中已默认包含消息云端搜索全部功能, 无需重复引入。

如果您想独立引入 <TUISearch> 消息云端搜索,请参考以下教程。

Web&H5 Vue2&Vue3

Uniapp Vue2&Vue3

前提条件

@tencentcloud/chat-uikit-vue ≥ 2.0.0,如未集成,请务必先根据 Vue2 & Vue3 TUIKit 快速集成指引进行集成。

引入 <TUISearch>

在您所需要使用 消息云端搜索 功能的 .vue 界面,引入 <TUISearch>。

<TUISearch> 参数说明

参数名	类型	说明
searchType String	global:全局搜索(default)	
	conversation:会话内搜索	

<TUISearch> 效果展示

<tuisearch searchtype="global"></tuisearch>	<tuisearch searchtype="conversation"></tuisearch>

使用 TUISearch

```
import { TUISearch } from "@tencentcloud/chat-uikit-vue";
// 全局搜索
<TUISearch searchType="global" />
```



// 会话内搜索

<TUISearch searchType="conversation" />

删除默认引入的 TUISearch

TUIKit 中默认集成 <TUISearch> ,如您不按照默认集成方式使用,可在 TUIKit/index.vue 中,注释掉 <TUISearch> 即可。

Uniapp TUISearch 支持两种方式引入:组件方式引入与界面方式引入。

前提条件

@tencentcloud/chat-uikit-uniapp ≥ 2.0.6,如未集成,请请务必先根据 uniapp TUIKit 快速集成指引进行集成。 组件方式引入

界面方式引入

在您所需要使用 消息云端搜索 功能的 .vue 界面,引入 <TUISearch>。

<TUISearch> 参数说明

参数名	类型	说明
searchType String	global:全局搜索	
	conversation:会话内搜索(default)	

<TUISearch> 效果展示

<tuisearch searchtype="global"></tuisearch>	<tuisearch searchtype="conversation"></tuisearch>

使用 TUISearch

```
// 以下路径仅为示例路径,请根据您项目自身路径进行调整
import { TUISearch } from "/TUIKit/components/TUISearch/index.vue";
// 全局搜索
<TUISearch searchType="global" />
// 会话内搜索
<TUISearch searchType="conversation" />
```

删除默认引入的 TUISearch





TUIKit 中默认集成 <TUISearch> , 如您不按照默认集成方式使用, 可在

TUIKit/components/TUIConversation/index.vue 中, 注释掉 <TUISearch> 即可。

在 pages.json 新增 TUISearch 页面

```
{
   "pages": [
   ...,
    {
        "path": "TUIKit/components/TUISearch/index",
        "style": {
            "navigationBarTitleText": "聊天记录"
        }
    }
    ],
    ...
}
```

跳转到 TUISearch 界面

```
uni.navigateTo({
    url: "/TUIKit/components/TUISearch/index",
});
```

高级指引

新增搜索消息类型

原"全局搜索"消息类型列表	新增后"全局搜索"消息类型列表

目录位置: src/TUIKit/components/TUISearch/search-type-list.ts

searchMessageTypeList 中包含了所有"搜索消息类型" Tab 定义,如需新增

searchMessageTypeList 未定义的搜索消息类型,请按照以下结构在 searchMessageTypeList 中进行 新增:

```
[keyName: string]: {
    key: string;// 消息搜索类型 key, 请保持唯一性
    label: string;// 消息搜索类型渲染 label
    value: Array<string>;// 消息搜索类型实际搜索列表
```



};

```
// 例如, 定义搜索自定义类型消息
export const searchMessageTypeList = {
    ...
    customMessage: {
        key: "customMessage",// 消息搜索类型 key, 请保持唯一性
        label: "自定义",// 消息搜索类型渲染 label
        value: [TUIChatEngine.TYPES.MSG_CUSTOM],// 消息搜索类型实际搜索列表
    }
};
```

因为 TUIKit 使用 i18next 支持国际化,如您声明新的 label,请在

```
src/TUIKit/locales/zh_cn/TUISearch.ts 以及 src/TUIKit/locales/en/TUISearch.ts 增加相
应的国际化词条进行翻译。
```

```
如需将已定义的 searchMessageTypeList 中某类型增加到全局搜索类型列表或者会话内搜索类型列表,仅需
将其 key 填入 globalSearchTypeKeys 或 conversationSearchTypeKeys 即可。
```

```
// 例如, 将以上新增的 自定义消息 customMessage 应用到"全局搜索"消息类型列表
export const globalSearchTypeKeys = [..., "customMessage"];
// 例如, 将以上新增的 自定义消息 customMessage 应用到"会话内搜索"消息类型列表
export const conversationSearchTypeKeys = [..., "customMessage"];
```

新增消息云端搜索时间范围

原"全局搜索"时间范围列表	新增后"全局搜索"时间范围列表

目录位置: src/TUIKit/components/TUISearch/search-time-list.ts

```
searchMessageTimeList 中包含了所有"搜索时间范围" Tab 定义,如需新增
searchMessageTimeList 未定义的搜索时间范围类型,请按照以下结构在 searchMessageTimeList 中
进行新增:
```

```
[keyName: string]: {
    key: string;// 消息搜索时间范围 key, 请保持唯一性
    label: string;// 消息搜索时间范围渲染 label
    value: {
        timePosition: number; // 消息搜索时间范围起始位置, 默认为 0, 从当前时间开始搜索
        timePeriod: number; // 从 timePosition 向前搜索的时间范围
    };
};
```



```
// 例如, 定义搜索"近两天"时间范围
export const searchMessageTimeList = {
    ...
    twoDay: {
        key: "twoDay",// 消息搜索时间范围 key, 请保持唯一性
        label: "近两天",// 消息搜索时间范围渲染 label
        value: {
            timePosition: 0,// 消息搜索时间范围起始位置, 默认为 0, 从当前时间开始搜索
            timePeriod: 2 * oneDay,// 从 timePosition 向前搜索的时间范围
        },
        },
    };
```

因为 TUIKit 使用 i18next 支持国际化,如您声明新的 label,请在

src/TUIKit/locales/zh_cn/TUISearch.ts 以及 src/TUIKit/locales/en/TUISearch.ts 增加相 应的国际化词条进行翻译。

无 UI 集成

以下为消息云端搜索无 UI 集成方案:

功能描述

搜索云端消息,提升 Chat SDK 使用体验,可以帮助用户从纷繁复杂的信息中直接找到预期内容,快捷方便;也可以 作为运营工具,增加相关内容的引导,简洁高效。

注意:

搜索云端消息功能 v3.1.0 起支持。

此接口本地限频 2 次/秒。

搜索**全部会话**的消息,如果匹配到的消息数量 messageCount > 1,则接口返回的 messageList 为[],您可以在 UI 上展示【 \${messageCount} 条相关记录】。如果您想高亮展示匹配到的消息,请参考 指定搜索,将接口返 回的 messageList 高亮展示。

搜索**全部会话**的消息,如果某个会话中匹配到的消息条数 = 1,则 messageList 为匹配到的那条消息。

社群、topic、直播群,不支持搜索云端消息。

搜索云端消息

接口

chat.searchCloudMessages(options);

参数



参数 options 为 Object 类型,包含的属性值如下:

Name	Туре	Description
keywordList	Array undefined	关键字列表,最多支持5个。 注意: 当消息发送者以及消息类型均未指定时,关键字列表必须非空;否则,头
keywordListMatchType	String undefined	关键字列表匹配类型 or:"或"关系搜索(默认) and:"与"关系搜索
senderUserIDList	Array undefined	指定 userID 发送的消息,最多支持 5 个。
messageTypeList	Array undefined	指定搜索的消息类型集合,默认搜索全部类型。 不传入时,表示搜索支持的全部类型消息 (TencentCloudChat.TYPES.MSG_FACE、TencentCloudChat 和 TencentCloudChat.TYPES.MSG_GRP_SYS_NOTICE 不支持) 传值时,取值详见TencentCloudChat.TYPES。
conversationID	String undefined	搜索"全部会话"还是搜索"指定的会话",不传入时,表示全部会话,默认 方式: C2C\${userID} (单聊) GROUP\${groupID} (群聊) 社群、topic、直播群,不支持搜索云端消息
timePosition	Number undefined	搜索的起始时间点。默认为0即代表从现在开始搜索,单位:秒。
timePeriod	Number undefined	从起始时间点开始的过去时间范围,单位秒。默认为0即代表不限制时能表过去一天。
cursor	String undefined	每次云端搜索的起始位置。第一次搜索时不要传入 cursor , 继续搜索 searchCloudMessages 接口返回的 cursor 的值 注意: 全量搜索时, cursor 的有效期为2分钟。

返回值

Promise

Name	Туре	Description
totalCount	Number	如果搜索指定会话,返回满足搜索条件的消息总数。 如果搜索全部会话,返回满足搜索条件的消息所在的所有会话总数量。
searchResultList	Array	满足搜索条件的消息根据会话 ID 分组。



cursor String 调用搜索接口续拉时需要填的游标。

其中 searchResultList 是个列表,内含搜索结果对象,参数说明如下:

Name	Туре	Description
conversationID	String	会话 ID。
messageCount	Number	当前会话一共搜索到了多少条符合要求的消息。
messageList	Array	 当前会话中所有满足搜索条件的消息列表 注意: 1.如果搜索指定会话, messageList 是本会话中所有满足搜索条件的消息列表。 2.如果搜索全部会话, messageList 中装载的消息条数会有如下两种可能: 如果匹配到的消息数量 messageCount >1,则接口返回的 messageList 为[],您可以在 UI 上展示【\${messageCount} 条相关 记录】。如果您想高亮展示匹配到的消息,请参考指定搜索,将接口返回的 messageList 高亮展示。 如果某个会话中匹配到的消息条数=1,则 messageList 为匹配到的 那条消息。

搜索全部会话的消息

不指定 conversationID ,指定关键字搜索。

示例

```
// 全量搜索, 指定关键字
// - 搜索消息里出现'你好'或'在哪里'
let promise = chat.searchCloudMessages({
  keywordList: ['你好', '在哪里']
});
promise.then(function(imResponse) {
// 搜索消息成功
  const { totalCount, cursor, searchResultList } = imResponse.data;
  console.log(totalCount); // 满足搜索条件的消息所在的所有会话总数量
  console.log(cursor); // 下一次云端搜索的起始位置, 如果没有表示搜索结果拉取完成
  console.log(searchResultList); // 满足搜索条件的消息根据会话 ID 分组, 分页返回分组结果
  for (let i = 0; i < searchResultList.length; i++) {</pre>
     const searchResultItem = searchResultList[i];
     const { conversationID, messageCount, messageList } = searchResultItem;
     console.log(conversationID); // 会话 ID
     console.log(messageCount); // 当前会话一共搜索到了多少条符合要求的消息
```



```
// 本次搜索【全部会话】, 那么 messageList 中装载的消息条数会有如下两种可能:
    // - 如果某个会话中匹配到的消息条数 > 1, 则 messageList 为空, 您可以在 UI 上显示" mess
    // - 如果某个会话中匹配到的消息条数 = 1, 则 messageList 为匹配到的那条消息,
    // 您可以在 UI 上显示之,并高亮匹配关键词。
    console.log(messageList);
    }
}).catch(function(imError) {
    console.error(imError); // 搜索消息失败
});
```

搜索指定会话的消息

指定 conversationID ,指定关键字搜索。

示例

```
// 指定会话,指定关键字搜索
// - 搜索在 'GROUPPublic001' 会话中, 消息里出现 '你好'或'在哪里'的消息。
let promise = chat.searchCloudMessages({
  keywordList: ['你好', '在哪里'],
  conversationID: 'GROUPPublic001'
});
promise.then(function(imResponse) {
  // 搜索消息成功
  const { totalCount, cursor, searchResultList } = imResponse.data;
  console.log(totalCount); // 满足搜索条件的消息总数量
  console.log(cursor); // 下一次云端搜索的起始位置, 如果没有表示搜索结果拉取完成
  console.log(searchResultList); // 当前会话搜索的消息结果
  const { conversationID, messageCount, messageList } = searchResultList[0];
  console.log(conversationID); // 会话ID
  console.log(messageCount); // 当前会话一共搜索到了多少条符合要求的消息
  console.log(messageList); // 本会话中所有满足搜索条件的消息列表
}).catch(function(imError); {
  console.error(imError); // 搜索消息失败
});
```

搜索自定义消息

1.使用接口 createCustomMessage 来创建自定义消息时,需要把搜索的文本放到 description 参数中。支持 关键词与 description 进行匹配。



2. 指定 messageTypeList 为 TencentCloudChat.TYPES.MSG_CUSTOM 做分类搜索,此时会搜索出所有 自定义消息。

示例

```
// 全量搜索, 指定关键字和消息类型搜索
// - 搜索消息类型为'自定义消息', 且消息的 description 里出现了'你好'或'在哪里'的消息
let promise = chat.searchCloudMessages({
  keywordList: ['你好', '在哪里'],
  messageTypeList: [TencentCloudChat.TYPES.MSG_CUSTOM],
});
promise.then(function(imResponse) {
// 搜索消息成功
  const { totalCount, cursor, searchResultList } = imResponse.data;
  console.log(totalCount); // 满足搜索条件的消息所在的所有会话总数量
  console.log(cursor); // 下一次云端搜索的起始位置, 如果没有表示搜索结果拉取完成
  console.log(searchResultList); // 满足搜索条件的消息根据会话 ID 分组, 分页返回分组结果
  for (let i = 0; i < searchResultList.length; i++) {</pre>
     const searchResultItem = searchResultList[i];
     const { conversationID, messageCount, messageList } = searchResultItem;
     console.log(conversationID); // 会话 ID
     console.log(messageCount); // 当前会话一共搜索到了多少条符合要求的消息
     // 本次搜索【全部会话】, 那么 messageList 中装载的消息条数会有如下两种可能:
     // - 如果某个会话中匹配到的消息条数 > 1, 则 messageList 为空, 您可以在 UI 上显示" mess
     // - 如果某个会话中匹配到的消息条数 = 1,则 messageList 为匹配到的那条消息,
     // 您可以在 UI 上显示之,并高亮匹配关键词。
     console.log(messageList);
   }
}).catch(function(imError) {
  console.error(imError); // 搜索消息失败
});
```

搜索富媒体消息

富媒体消息包含文件、图片、语音、视频消息。 1.对于文件消息,界面通常显示文件名。如果调用 createFileMessage 创建文件消息时传入 fileName 参 数, fileName 会作为文件消息被搜索的内容,与搜索关键词进行匹配。

2. 对于图片、语音、视频消息,并没有类似 fileName 这种名称,界面通常显示缩略图或时长,此时指定 keywordList 搜索无效。

3. 可以指定 messageTypeList 为

TencentCloudChat.TYPES.MSG_IMAGE / TencentCloudChat.TYPES.MSG_AUDIO / TencentCloudCh at.TYPES.MSG_FILE 做分类搜索,此时会搜索出所有指定类型的消息。

示例



```
// 全量搜索,只指定消息发送者和消息类型搜索(当指定发送者和消息类型时,关键字列表可以为空)
// - 搜索消息发送者为 'user1' 或 'user2', 且消息类型为 '图片消息'、 '语音消息' 或 '文件消息'
let promise = chat.searchCloudMessages({
  senderUserIDList: ['user1', 'user2'],
  messageTypeList: [
     TencentCloudChat.TYPES.MSG IMAGE,
     TencentCloudChat.TYPES.MSG_AUDIO,
     TencentCloudChat.TYPES.MSG_FILE
  1,
});
promise.then(function(imResponse) {
// 搜索消息成功
  const { totalCount, cursor, searchResultList } = imResponse.data;
  console.log(totalCount); // 满足搜索条件的消息所在的所有会话总数量
  console.log(cursor); // 下一次云端搜索的起始位置, 如果没有表示搜索结果拉取完成
  console.log(searchResultList); // 满足搜索条件的消息根据会话 ID 分组, 分页返回分组结果
  for (let i = 0; i < searchResultList.length; i++) {</pre>
     const searchResultItem = searchResultList[i];
     const { conversationID, messageCount, messageList } = searchResultItem;
     console.log(conversationID); // 会话 ID
     console.log(messageCount); // 当前会话一共搜索到了多少条符合要求的消息
     // 本次搜索【全部会话】, 那么 messageList 中装载的消息条数会有如下两种可能:
     // - 如果某个会话中匹配到的消息条数 > 1, 则 messageList 为空, 您可以在 UI 上显示" mess
     // - 如果某个会话中匹配到的消息条数 = 1,则 messageList 为匹配到的那条消息,
     // 您可以在 UI 上显示之,并高亮匹配关键词。
     console.log(messageList);
   }
}).catch(function(imError) {
  console.error(imError); // 搜索消息失败
});
```

搜索地理位置消息

1. 支持 latitude 、 longitude 、 description 与关键词进行匹配。

```
2.指定 messageTypeList 为 TencentCloudChat.TYPES.MSG_LOCATION 做分类搜索,此时会搜索出所 有地理位置消息。
```

示例

```
// 全量搜索,指定关键字和消息类型搜索
// - 搜索消息类型为 '地理位置',且消息的 latitude、longitude、description 里出现了 '你好' d
let promise = chat.searchCloudMessages({
    keywordList: ['你好', '在哪里'],
    messageTypeList: [TencentCloudChat.TYPES.MSG_LOCATION],
});
```


```
promise.then(function(imResponse) {
// 搜索消息成功
  const { totalCount, cursor, searchResultList } = imResponse.data;
  console.log(totalCount); // 满足搜索条件的消息所在的所有会话总数量
  console.log(cursor); // 下一次云端搜索的起始位置, 如果没有表示搜索结果拉取完成
  console.log(searchResultList); // 满足搜索条件的消息根据会话 ID 分组, 分页返回分组结果
  for (let i = 0; i < searchResultList.length; i++) {</pre>
     const searchResultItem = searchResultList[i];
     const { conversationID, messageCount, messageList } = searchResultItem;
     console.log(conversationID); // 会话 ID
     console.log(messageCount); // 当前会话一共搜索到了多少条符合要求的消息
     // 本次搜索【全部会话】, 那么 messageList 中装载的消息条数会有如下两种可能:
     // - 如果某个会话中匹配到的消息条数 > 1,则 messageList 为空,您可以在 UI 上显示" mess
     // - 如果某个会话中匹配到的消息条数 = 1, 则 messageList 为匹配到的那条消息,
     // 您可以在 UI 上显示之, 并高亮匹配关键词。
     console.log(messageList);
   }
}).catch(function(imError) {
  console.error(imError); // 搜索消息失败
});
```

搜索合并消息

1.使用接口 createMergerMessage 来创建合并消息时,需要把搜索的文本放到 title 或 abstractList 参数 中,支持 title 、 abstractList 与关键词进行匹配。

2.指定 messageTypeList 为 TencentCloudChat.TYPES.MSG_MERGER 做分类搜索,此时会搜索出所有 合并消息。

示例

```
// 全量搜索, 指定关键字和消息类型搜索
// - 搜索消息类型为'合并消息',且消息的 title 或 abstractList 里出现了'你好'或'在哪里'的
let promise = chat.searchCloudMessages({
  keywordList: ['你好', '在哪里'],
  messageTypeList: [TencentCloudChat.TYPES.MSG_MERGER],
});
promise.then(function(imResponse) {
// 搜索消息成功
  const { totalCount, cursor, searchResultList } = imResponse.data;
  console.log(totalCount); // 满足搜索条件的消息所在的所有会话总数量
  console.log(cursor); // 下一次云端搜索的起始位置, 如果没有表示搜索结果拉取完成
  console.log(searchResultList); // 满足搜索条件的消息根据会话 ID 分组, 分页返回分组结果
  for (let i = 0; i < searchResultList.length; i++) {</pre>
     const searchResultItem = searchResultList[i];
     const { conversationID, messageCount, messageList } = searchResultItem;
     console.log(conversationID); // 会话 ID
```



```
console.log(messageCount); // 当前会话一共搜索到了多少条符合要求的消息
    // 本次搜索【全部会话】, 那么 messageList 中装载的消息条数会有如下两种可能:
    // - 如果某个会话中匹配到的消息条数 > 1, 则 messageList 为空, 您可以在 UI 上显示" mess
    // - 如果某个会话中匹配到的消息条数 = 1, 则 messageList 为匹配到的那条消息,
    // 您可以在 UI 上显示之, 并高亮匹配关键词。
    console.log(messageList);
    }
}).catch(function(imError) {
    console.error(imError); // 搜索消息失败
});
```



文本消息翻译功能 效果展示

最近更新时间:2024-10-29 16:40:35

"集成前"与"集成后"效果对比



设置入口及语言选择页面如下所示: 目标语言设置入口 目标语言选择页



我	
Alice ID: 暂无个性签名	>
加我为好友时	需要验证 >
消息阅读状态 关闭后,您收发的消息均不带消息阅读状态,您 将无法看到对方是否已读,同时对方也无法看到 您是否已读。	
显示在线状态 开启后,您将可以在会话列表和通讯录中看到好 友在线或离线的状态提示。	
将消息翻译为	English >
显示通话记录	
月 二 消息 通讯录	上 我



<	将消息翻译为
简体中文	
繁體中文	
English	\checkmark
日本語	
한국어	
Français	
Español	
Italiano	
Deutsch	
Türkçe	
Русский	

注意:

TUITranslationPlugin 依赖于 TUIChat , 不能单独集成。

仅支持文本消息、文本类的引用或回复消息,图片、语音、视频、文件、表情、自定义消息等不支持翻译。 并不是任意源语言都能翻译成设定的目标语言。例如,英语可以翻译成印地语,但是中文不可以。目前已经支持的 翻译语言请参见文本翻译语言支持。如果翻译失败,请查询该文档切换源语言或目标语言。



集成指引 Android & iOS

最近更新时间:2025-03-27 15:35:28

插件功能

当您进入了聊天界面后,可以手动长按消息列表中的文本消息 item,在出现的菜单中,单击**翻译按钮**,翻译文本。 注意:

文本消息翻译功能由插件提供,使用需集成 TUITranslationPlugin , 7.2 及以上版本支持。

文本消息翻译功能仅对**专业版 Plus 和企业版**客户开放,购买专业版 Plus 和企业版 后可使用;体验版支持一定额度 免费试用,有效期一个月。

如果您不需要翻译功能,不集成该插件即可。长按文本消息时,不会显示翻译按钮。

如果您需要翻译功能,需集成 TUIChat 和 TUITranslationPlugin 。集成方法请参见集成基础功能 (Android/iOS)。集成后不需要进行任何设置,长按文本消息时,自动显示**翻译按钮**。

集成 TUITranslationPlugin 后,您还可以在**我**界面设置翻译的目标语言。默认目标语言是当前 TUIChat 所使用的语言。

注意:

TUITranslationPlugin 依赖于 TUIChat ,不能单独集成。

仅支持文本消息、文本类的引用或回复消息,图片、语音、视频、文件、表情、自定义消息等不支持翻译。 并不是任意源语言都能翻译成设定的目标语言。例如,英语可以翻译成印地语,但是中文不可以。目前已经支持的 翻译语言请参见 文本翻译语言支持。如果翻译失败,请查询该文档切换源语言或目标语言。



投票功能 效果展示

最近更新时间:2025-01-23 11:43:53

发起投票

集成 TUIPollPlugin 成功后, TUIChat 输入栏的+菜单中会自动展示投票插件入口。发起人**单击投票**后, 可以在接下来的发起投票界面,设置投票主题、选项,并进行相关设置,如下图所示:





くとなった。	己投票
周末聚餐吃什么?	投票主题
一 烧烤	选项设置
一 火锅	
+ 添加选项	
允许多选	
匿名投票	投票设置
公开结果	
ل ا لا	之布

参与人投票

发起人创建投票成功后, TUIChat 消息列表中会自动展示投票消息。群组里的其他成员, 收到投票消息后, 可以参与到投票中。

发起人投票

发起人自己也可参与投票。



投票数据统计

投票结束或本人完成投票后,可以看到当前的投票数据统计。单击某一选项可查看详细的投票成员列表,如下图所示:

票数据统计	+		选项成	员列表	
<	my friends			<	烧烤
	Candy Hi everyone			Car	ndy
	Alice			Alic	ce
	 □ 周末聚餐吃什么? [单选][公开结果][不匿名] 	1		Dav	vid
	1 烷烤 3 (1) 74% Candy; Alice; David 2 2 火锅 1 (2) 25%				已投3票,占比75%
	Bob 3 自助餐 0 票 0%				
	4 川菜 ○ 票 0% 5 粤菜 ○ 票 0%				
	应参与人数:4 实际参与人数:4]			
	已投票				
٩	() (+)			

投票设置

为了满足不同的投票需求, TUIPollPlugin 开放了几个投票设置。不同的设置项组合, 会产生不同的效果。例 如:

开启**允许多选**,一个参与人可投多个选项。

开启**匿名投票**,投票发起人无法看到投票者的具体信息,只能看到票数统计。

关闭公开结果,只有投票发起人能看到投票数据统计,参与人无法得知。

说明:

发起人不可以在发起投票后修改设置。





集成指引 Android & iOS

最近更新时间:2025-01-23 11:43:53

插件功能

从 7.1 版本开始,您可以集成投票插件 TUIPollPlugin 。投票插件是一个 UI 闭源库,依赖于 TUIChat 。 集成后,您可以直接在 TUIChat 中发起群组投票、查看投票统计结果、参与其他人发起的投票等。

Android 集成

TUIPollPlugin 是闭源插件,需要通过 gradle 集成。找到 App 的 build.gradle,在 dependencies 中添加投票插件的依赖。

经典版

}

```
dependencies {
    ...
# 集成聊天功能
api project(':tuichat')
    ...
# 集成投票插件,从 7.1 版本开始支持
api "com.tencent.imsdk:tuipoll-plugin:7.3.4358"
    ...
```

iOS 集成

```
TUIPollPlugin 和普通的 TUIKit 组件一样,都可以通过 CocoaPods 快速集成。
对于投票插件来说,您只需要向 Podfile 中添加一行即可完成:
经典版
```

```
# Uncomment the next line to define a global platform for your project
# ...
# 集成聊天功能
pod 'TUIChat/UI_Classic'
```



```
# 集成投票插件,从 7.1 版本开始支持
pod 'TUIPollPlugin'
...
end
```

注意:

TUIPollPlugin依赖于TUIChat, 单独集成TUIPollPlugin会使功能异常, 无法正常展示投票界面。TUIPollPlugin从 7.1 版本开始支持, 使用投票插件需要将TUIChat升级到 7.1 及以上版本。



Flutter

最近更新时间:2025-01-23 11:43:54

插件功能

从tencent_cloud_chat_sdk 5.1.5版本之后,您可以集成腾讯云即时通信官方提供的投票插件 tencent_cloud_chat_vote_plugin,此插件为闭源插件,集成后,您可以在群里(Community和 AVChatRoom 除外) 集成投票功能。投票功能包括发起(单选、多选)投票、查看投票结果,参与投票等。Flutter 端与 Native 的投票能 力互通。

环境与版本

本插件依赖插件以及环境 Fluter 3.10.0及以上 tencent_cloud_chat_sdk 5.1.5及以上

插件引入

通过 pub 可将投票插件 tencent_cloud_chat_vote_plugin 1.0.2 引入到项目中:

// 集成最新版本 pub add tencent_cloud_chat_vote_plugin

// 集成指定版本, 在项目pubspec.yaml中dependencies字段加入 tencent_cloud_chat_vote_plugin: "version"

核心组件

TencentCloudChatVoteCreate 创建投票 TencentCloudChatVoteMessage 投票消息解析 TencentCloudChatVoteDetail 投票详情展示

插件集成



初始化插件

// 放在登录IM后

```
await TencentCloudChatVotePlugin.initPlugin();
```

创建投票

用户单击投票按钮,可创建投票。

```
import 'package:example/config.dart';
import 'package:flutter/material.dart';
import 'package:tencent_cloud_chat_vote_plugin/components/vote_create/vote_create.d
class VoteCreateExample extends StatefulWidget {
 const VoteCreateExample({super.key});
 Coverride
 State<StatefulWidget> createState() => VoteCreateExampleState();
}
class VoteCreateExampleState extends State {
  Coverride
 Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
       title: const Text("创建投票"),
      ),
     body: Container(
        padding: const EdgeInsets.all(16),
        child: TencentCloudChatVoteCreate(
          groupID: ExampleConfig.testGruopID,
          onCreateVoteSuccess: () {
            Navigator.pop(context);
          },
        ),
     ),
    );
  }
}
```

TencentCloudChatVoteCreate 参数说明

参数	说明
groupID	需要创建投票的群 ID,同 IM 群 ID, Community和 AVChatRoom 除外。



即时通信 IM

onCreateVoteSuccess

创建投票成功回调。

投票消息解析

```
import 'package:example/config.dart';
import 'package:example/vote_detail_example.dart';
import 'package:flutter/material.dart';
import 'package:tencent_cloud_chat_sdk/enum/history_msg_get_type_enum.dart';
import 'package:tencent_cloud_chat_sdk/models/v2_tim_message.dart';
import 'package:tencent_cloud_chat_sdk/models/v2_tim_value_callback.dart';
import 'package:tencent_cloud_chat_sdk/tencent_im_sdk_plugin.dart';
import 'package:tencent_cloud_chat_vote_plugin/tencent_cloud_chat_vote_plugin.dart'
class VoteMessageExample extends StatefulWidget {
  const VoteMessageExample({super.key});
  @override
  State<StatefulWidget> createState() => VoteMessageExampleState();
}
class VoteMessageExampleState extends State {
 V2TimMessage? message;
 getTestV2TimMessage() async {
    V2TimValueCallback<List<V2TimMessage>> messageListRes =
        await TencentImSDKPlugin.v2TIMManager
            .getMessageManager()
            .getHistoryMessageList(
              count: 1,
              groupID: ExampleConfig.testGruopID,
              getType: HistoryMsgGetTypeEnum.V2TIM_GET_CLOUD_OLDER_MSG,
            );
    if (messageListRes.code == 0) {
      if (messageListRes.data != null) {
        if (messageListRes.data!.isNotEmpty) {
          setState(() {
            message = messageListRes.data!.first;
          });
        l
      }
    }
  }
 bool isEnd = false;
  @override
  void initState() {
    super.initState();
```



```
Future.delayed(
       const Duration(
         milliseconds: 300,
       ), () {
     setState(() {
       isEnd = true;
     });
   });
   // 页面运动结束再显示组件
   getTestV2TimMessage();
 }
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: const Text("投票消息体"),
     ),
     body: !isEnd
         ? Container()
          : message != null
              ? TencentCloudChatVoteMessage(
                 message: message!,
                 onTap: (
                    TencentCloudChatVoteDataOptoin option,
                   TencentCloudChatVoteLogic data,
                 ) {
                   print(data.voteData.toJson());
                   Navigator.push(
                     context,
                     MaterialPageRoute(
                       builder: (context) => VoteDetailExample(
                         option: option,
                         data: data,
                       ),
                      ),
                   );
                 },
                )
              : const Center(
                 child: Text("未获取到正确的Message实例"),
               ),
   );
 }
}
```



TencentCloudChatVoteMessage参数说明

参数	说明
message	投票消息, V2TimMessage 类型。
onTap	单击投票回调,当投票为公开且实名时,可以打开群 投票详情。

投票详情查看

```
import 'package:flutter/material.dart';
import 'package:tencent_cloud_chat_vote_plugin/tencent_cloud_chat_vote_plugin.dart'
class VoteDetailExample extends StatelessWidget {
 final TencentCloudChatVoteDataOptoin option;
 final TencentCloudChatVoteLogic data;
 const VoteDetailExample({
   super.key,
   required this.option,
   required this.data,
  });
  @override
 Widget build(BuildContext context) {
    return Scaffold(
     appBar: AppBar(
       title: Text(option.option),
     ),
     body: Padding(
        padding: const EdgeInsets.all(16),
        child: TencentCloudChatVoteDetail(
         option: option,
         data: data,
       ),
     ),
   );
  }
}
```

TencentCloudChatVoteDetail 参数说明

参数	说明
option	TencentCloudChatVoteDataOption 类型,投票详情数据,由



	TencentCloudChatVoteMessage 单击时获取。
data	TencentCloudChatVoteLogic 类型由 TencentCloudChatVoteMessage 点击时获取。



接龙功能 效果展示

最近更新时间:2025-01-23 11:43:53

发起群接龙

集成 TUIGroupNotePlugin 成功后, TUIChat 输入栏的+菜单中会自动展示群接龙插件入口。发起人单击 **接龙**后,可以在接下来的创建接龙界面,填写接龙主题、说明、截止时间、示例,设置接龙特性等,如下图所示:

接龙信	言息填写		接龙设	と置	
					_
	<	发送		<	发送
	接龙	设置		接龙	设置
	添加接龙主题			允许每人提交多份	
	添加接龙说明			设置截止时间	未设置 >
	截止时间			有新提交时在群里发通知	
	例 可填写接龙格式				
	\oplus				

参与人接龙



发起人创建接龙成功后, TUIChat 消息列表中会自动展示接龙消息。群组里的其他成员, 收到接龙消息后, 可以参与到接龙中。

发起人接龙

发起人自己也可参与接龙,如下图所示:

发起人	人新收到	间接龙消息		发起人	、准备	接龙	
	<	my friends			<		发送
		"Alice"创建讨论组				接龙	设置
		hel	lo			由 Alice 发起,已有 2 人参加	
		Candy Hi everyone			#下 活动	「 周打球谁参加? ^{时间周三}	
		■ 群接龙 下周打球谁参加?			例	张三	
		由 Alice 发起,已有 2 人参加			1	Candy	+1
		1 Candy 2 Bob			2	Bob	+1
		立即接龙			3	Alice	÷1
		Candy已参加群接龙 [<mark>下周打球</mark> <mark>谁参加</mark> ?]			Ð		
		Bob 已参加群接龙 [下周打球谁 参加?]					
	0		(:) $(+)$				

接龙设置

为了满足不同的接龙需求, TUIGroupNotePlugin 开放了几个设置。例如: 开启**允许每人提交多份**, 一个参与人可接龙多次。 设置**截止时间**, 超过截止时间, 所有人不能再参与接龙。 开启**有新提交时在群里发通知**, 有人新增接龙条目后, 群里会有通知消息提醒。 **说明:**



发起人可以在发起接龙后修改设置。



集成指引 Android & iOS

最近更新时间:2025-01-23 11:43:54

插件功能

从 7.1 版本开始,您可以集成群接龙插件 TUIGroupNotePlugin 。群接龙插件是一个 UI 闭源库,依赖于 TUIChat 。

集成后,您可以直接在 TUIChat 中发起群组接龙活动。

Android 集成

TUIGroupNotePlugin 是闭源插件,需要通过 gradle 集成。找到 App 的 build.gradle,在 dependencies 中添加 投票插件的依赖。

经典版

}

```
dependencies {
    ...
# 集成聊天功能
api project(':tuichat')
...
# 集成接龙插件,从 7.1 版本开始支持
api "com.tencent.imsdk:tuigroupnote-plugin:7.3.4358"
...
```

iOS 集成

TUIGroupNotePlugin 和普通的 TUIKit 组件一样,都可以通过 CocoaPods 快速集成。详细的集成步骤可以参见集成基础功能。

```
对于接龙插件来说,您只需要向 Podfile 中添加一行即可完成:
经典版
```

```
\# Uncomment the next line to define a global platform for your project \# \ldots
```



```
# 集成聊天功能
pod 'TUIChat/UI_Classic'
...
# 集成接龙插件,从 7.1 版本开始支持
pod 'TUIGroupNotePlugin'
...
end
```

注意:

TUIGroupNotePlugin 依赖于 TUIChat , 单独集成 TUIGroupNotePlugin 会使功能异常, 无法正常展示接龙界面。

TUIGroupNotePlugin 从 7.1 版本开始支持,使用接龙插件需要将 TUIChat 升级到 7.1 及以上版本。



会话标记功能 效果展示

最近更新时间:2025-01-23 11:43:53

标记会话

集成 TUIConversationMarkPlugin 成功后,在 TUIConversation 会话列表界面左滑 Cell 就能标记一个 会话,当会话被标记后,会在会话列表上方展示"标记"分组栏,如下图所示:

记会话		"标记"	分组	
	_			
腾讯云·IM	(\pm)		腾讯云·IM	÷
Q. 搜索			Q 搜索	
示例好友 【安全提示】本 APP 仅用于体验腾讯云即时	16:10 †通信 I		全部 标记 1	
2	11:51		yahaha-昵称 今天晚上在哪聚餐?	P.
yahaha-昵称 今天晚上在哪聚餐?	11:45			
在线客服示例 您好,欢迎查看在线客服示例。	11:40			
不显示				
标记				



取消标记会话

如果一个会话已经被标记,当对该会话 Cell 左滑时,可以取消标记会话。

注意:

如果从来没有标记过会话,"标记"分组默认不展示,如果标记过会话,即使该会话已经被取消标记,"标记"分组也会一直展示。



集成指引 Android & iOS

最近更新时间:2025-01-23 11:43:53

插件功能

从 7.3 版本开始, 您可以集成会话标记插件 TUIConversationMarkPlugin 。会话标记插件是一个 UI 闭源库, 依赖于 TUIConversation 。集成后, 您可以在 TUIConversation 中标记会话、取消标记会话。

Android 集成

TUIConversationMarkPlugin 是闭源插件,需要通过 gradle 集成。找到 App 的 build.gradle,在 dependencies 中添加标记插件的依赖。

经典版

}

```
dependencies {
    ...
    # 集成会话功能
    api project(':tuiconversation')
    ...
    # 集成会话标记插件,从 7.3 版本开始支持
    api "com.tencent.imsdk:tuiconversationmark-plugin:7.3.4358"
    ...
```

iOS 集成

TUIConversationMarkPlugin 和普通的 TUIKit 组件一样,都可以通过 CocoaPods 快速集成。详细的集成步骤可以参见 集成基础功能。

对于会话标记插件来说,您只需要向 Podfile 中添加一行即可完成: 经典版

```
\# Uncomment the next line to define a global platform for your project \# \ldots
```

集成会话功能



```
pod 'TUIConversation/UI_Classic'
...
# 集成会话标记插件,从 7.3 版本开始支持
pod 'TUIConversationMarkPlugin'
...
end
```

注意:

```
TUIConversationMarkPlugin依赖于TUIConversation单独集成TUIConversationMarkPlugin会使功能异常,无法正常展示会话标记界面。TUIConversationMarkPlugin从 7.3 版本开始支持,使用会话标记插件需要将TUIConversation到 7.3 及以上版本。
```



会话分组功能 效果展示

最近更新时间:2025-01-23 11:43:54

展示会话分组

集成 TUIConversationGroupPlugin 成功后,在 TUIConversation 会话列表上方会展示会话分组管理 View,里面由各个分组 Item 组成,每个分组 Item 包含了分组名和分组未读数信息。如果点击分组管 理 View 右边的 三 按钮,就会弹出分组列表界面,如下图所示:





管理会话分组

在会话分组列表界面,点击右上角的"管理"按钮,就可以跳转到分组管理界面,在分组管理界面您可以创建、删除、 编辑、隐藏分组,也能对分组进行重排序,如下图所示:

管理会	会话分组		创	建会	话分组		
	/	伊方			1		但左
		1本1子				新增分组	1木1子
	展示云话ガ组				分组名称★		
	全部				请输入分组名		
	未读	隐藏			包含会话★		
	@我	隐藏			+ 添加会话		
	e GroupA	0 =					
	e GroupB						
	+ 新增分组						
	隐藏会话分组						
	石十 4 岁µ						
	单聊	(展示)					

会话分组操作

在会话列表界面,您可以左滑会话 Cell 对其进行分组操作,例如添加会话到分组,把会话移动到其他分组,创建一个包含该会话的新分组等。

注意:

会话分组栏默认会展示"全部"、"未读"、"@我"三个常驻分组,如果不需要某个分组,可以在分组管理界面隐藏对应 分组。





集成指引 Android & iOS

最近更新时间:2025-01-23 11:43:53

插件功能

从 7.3 版本开始, 您可以集成会话分组插件 TUIConversationGroupPlugin 。会话分组插件是一个 UI 闭源 库, 依赖于 TUIConversation 。集成后, 您可以创建分组、删除分组、编辑分组、隐藏分组、对分组进行重排 序。

Android 集成

TUIConversationGroupPlugin 是闭源插件,需要通过 gradle 集成。找到 App 的 build.gradle,在 dependencies 中添加分组插件的依赖。

经典版

```
dependencies {
    ...
    # 集成会话功能
    api project(':tuiconversation')
    ...
    # 集成会话分组插件,从 7.3 版本开始支持
    api "com.tencent.imsdk:tuiconversationgroup-plugin:7.3.4358"
    ...
}
```

iOS 集成

TUIConversationGroupPlugin 和普通的 TUIKit 组件一样,都可以通过 CocoaPods 快速集成。详细的集成 步骤可以参见 集成基础功能。

对于会话分组插件来说,您只需要向 Podfile 中添加一行即可完成: 经典版

```
\# Uncomment the next line to define a global platform for your project \# \ldots
```



集成会话功能

```
pod 'TUIConversation/UI_Classic'
...
# 集成会话分组插件,从 7.3 版本开始支持
pod 'TUIConversationGroupPlugin'
...
end
```

注意:

TUIConversationGroupPlugin 依赖于 TUIConversation , 单独集成

TUIConversationGroupPlugin 会使功能异常,无法正常展示会话分组界面。

TUIConversationGroupPlugin 从 7.3 版本开始支持,使用会话分组插件需要将 TUIConversation 升级 到 7.3 及以上版本。