

云通信

服务能力综述

产品文档



腾讯云

【版权声明】

©2013-2018 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】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【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模式的承诺或保证。

文档目录

服务能力综述

服务能力综述

单聊消息综述

批量发消息与全员消息推送

群组系统

群组管理综述

群组消息综述

用户资料和好友关系综述

接入服务系统（服务端集成）

在线状态综述

消息缓存和存储

多终端同时在线

离线推送

服务能力综述

服务能力综述

最近更新时间：2018-05-31 15:51:40

单聊消息

应用场景	App 内双人聊天
	管理员后台发送消息
	模拟系统消息
发送单聊消息	通过 ImSDK 发送单聊消息
	通过 REST API 发送消息
接收单聊消息	ImSDK 内接收单聊消息
	ImSDK 离线消息
	ImSDK 本地消息
单聊权限控制	App 内任意两个用户之间发送单聊消息
	App 管理员发送单聊消息
	只允许给好友发送消息
	拒绝来自某人的消息
获取聊天记录	发消息之后回调
	消息记录下载
多终端同步	
单聊消息的离线推送	APNs 离线推送
	Android 离线推送
单聊消息中携带发送者资料	
单聊离线消息处理流程	

批量发消息与全员消息推送

应用场景	
向一批用户发送消息	
向 App 中的所有用户发送消息	
向 App 中满足特定条件的用户发送消息	
ImSDK 接收推送消息	

群组系统

群组系统介绍

群组系统简介	
群组形态介绍	
群组管理简介	
群组数据结构	群基础资料
	群成员资料
自定义群组 ID	
自定义字段	
自定义群组形态	
自定义回调	

群组消息综述

发送群消息	通过 ImSDK 发送群消息
	通过 REST API 发送群消息
	通过 REST API 发送系统通知
在群消息中携带发送者相关信息	

接收群消息	App 接收群实时消息
	App 获取群离线/历史消息
	App 获取群消息中携带的发送者信息
离线消息与历史消息	
群消息接收选项	
收发消息权限	
消息发送控制	禁言
	脏字过滤
	群内发言之前回调
	云通信后台进行定制化开发
	消息优先级与频率控制
成员变更通知	
群消息的 APNs 推送	
App 后台获取群消息记录	
消息删除	
群消息的 SEQ 机制	
群组离线消息处理流程	

群组管理综述

创建群组	通过 ImSDK 创建群组
	通过 REST API 创建群组
	自定义群组 ID
	创建群组的配额限制
	创建群组之后的 ImSDK 系统消息
	创建群组相关的第三方回调
转让群组	通过 ImSDK 转让群组

	通过 REST API 转让群组
	转让群组之后的 ImSDK 系统消息
	转让群组相关的第三方回调
解散群组	通过 ImSDK 解散群组
	通过 REST API 解散群组
	解散群组之后的 ImSDK 系统消息
	群组解散相关第三方回调
群组资料管理	通过 ImSDK 获取/修改群组资料
	通过 REST API 获取/修改群组资料
	群资料被修改之后的 ImSDK 事件消息
	修改群组资料相关第三方回调
获取群成员资料	通过 ImSDK 获取群组成员列表
	通过 REST API 获取成员列表
修改群成员资料	通过 ImSDK 修改群成员资料
	通过 REST API 修改群成员资料
邀请加群/拉人入群	通过 ImSDK 邀请加群/拉人入群
	通过 REST API 拉人入群
	拉人入群相关的 ImSDK 事件消息
	邀请加群相关的 ImSDK 系统通知
	拉人入群/邀请加群相关第三方回调
申请加群	通过 ImSDK 申请加群
	处理申请加群
	申请加群相关的第三方回调
删除群组成员	通过 ImSDK 删除群组成员
	通过 REST API 删除群组成员

	删除群组成员相关的 ImSDK 事件消息/系统通知
	删除群组成员相关的第三方回调
主动退群	通过 ImSDK 主动退群
	主动退群相关的 ImSDK 事件消息/系统通知
	主动退群相关的第三方回调
获取用户所加入的群组	通过 mSDK 取用户自身所加入的群组
	通过 REST API 取用户所加入的群组
群未决信息列表	
群组发现	依照群组 ID 精确查找
	获取 App 中的所有群组

用户资料与关系链

用户资料介绍	用户基础资料
	用户自定义资料
	自定义资料字段的申请
	自定义资料字段的命名规范
关系链介绍	关系链特性
	标配好友字段
	自定义好友字段
设置用户资料	设置昵称
	设置好友验证方式
	设置头像
	设置个性签名
	设置自定义资料
设置好友资料	设置好友备注

	设置好友自定义字段
获取资料	获取所有用户资料
	获取关系链相关资料
好友关系链操作	好友关系链操作网络异步调用
	好友关系链本地存储
	好友分组
未决请求和推荐	应用场景
	操作接口
黑名单	应用场景
	黑名单操作
好友列表的更新	未启用资料存储更新时机
	启用资料存储更新时机
搜索用户	

接入服务系统

接入服务系统（服务端集成）	接入服务—— HTTP 头部字段说明
	接入服务——缓存控制

在线状态

术语约定	在线
	上线
	下线
查询用户在线状态	
用户在线状态变更通知	

消息缓存和存储

离线消息存储	概念
	消息接收
	Server 保存时长
	Server 删除时机
漫游消息存储	概念
	获取漫游消息
	漫游消息保存时长
最近联系人消息	概念
	获取最近联系人
	禁用最近联系人
	最近联系人限制
App 本地存储	本地存储获取
	本地消息删除
	禁用本地存储
Server 消息删除	

多终端同时在线

概念介绍	互踢
	同时在线
客户端接口说明	
功能和限制因素	

离线推送

概念	
应用场景	
iOS APNs 推送	推送格式说明
	基本接口说明
	Ext 扩展设置
	设置推送声音
	单条消息声音设置
Android 离线推送	
后台发送消息	

专项指引

云通信平滑迁移方案	
-----------	--

单聊消息综述

最近更新时间：2018-11-13 10:34:50

应用场景

App 内双人聊天

单聊消息适用于 App 内双人聊天，类似 QQ 好友、微信好友的聊天方式。

管理员后台发送消息

单聊消息可以由 App 管理员在后台发送消息，也可以模拟其他用户身份发送消息。

模拟系统消息

通过 App 管理员后台发送消息，可以模拟系统消息，以系统消息形式给用户下达通知。App 端收到 App 管理员自定义消息可做特殊处理。

发送单聊消息

通过 ImSDK 发送单聊消息

通过 ImSDK 发送消息，首先根据目标用户的 `identifier`，获取对应会话，再调用客户端接口 `sendMessage` 发送消息，目前 SDK 内部发送消息不会判断是否好友关系，有需要判断好友关系场景，需要 App 拉取好友列表或者通过检查是否好友关系接口进行判断后发送。另外可以通过云通信控制台相应 SDKAPPID【应用配置】>【功能配置】>【单聊消息检验关系链】设置开启，强制校验好友关系。详细请参阅以下接口描述：

- [Android 收发消息](#)
- [iOS 收发消息](#)
- [Windows 收发消息](#)
- [Web 收发消息](#)

通过 REST API 发送消息

使用 REST API 可以使用 App 管理员身份给用户发送消息，另外在特殊场景，也可以使用普通用户身份作为发送方发送消息。详细请参阅 [Rest API 发送消息](#)。

接收单聊消息

ImSDK 内接收单聊消息

ImSDK 接收消息都是通过统一回调接口 `onNewMessage` 返回，通过 `onNewMessage` 回调获得消息后，能根据消息获取相应的会话。详细请参阅以下接口描述：

- [Android 消息接收](#)
- [iOS 消息接收](#)
- [Windows 消息接收](#)
- [WebSDK 初始化新消息回调](#)

ImSDK 离线消息

发送者发消息时，如果接收方不在线，IM Server 会对消息进行暂存，存储时常默认为 7 天，当接收方 7 天内登录时，可以通过 `onNewMessage` 回调获取到离线消息。

ImSDK 的在线消息和离线消息，都是通过 `onNewMessage` 回调，在登录成功以后，SDK 会先拉取离线消息，通过 `onNewMessage` 回调，离线消息完成以后，会通过 `onRefresh` 回调通知。可在此时刷新页面。

ImSDK 本地消息

在 ImSDK 接收到消息后，默认情况下会对消息进行缓存和存储（如果不需要，可以调用 `disableStorage` 接口进行存储禁用，提升性能），各个终端通过调用 `getMessage` 接口可以获取本地的记录（开启漫游的情况下，还可以获取 Server 漫游的消息）。详细请参阅以下接口描述：

- [Android 获取消息](#)
- [iOS 获取消息](#)
- [Windows 获取消息](#)

单聊权限控制

App 内任意两个用户之间发送单聊消息

App 内不进行好友关系的判断，任意两个用户默认是可以随意发消息。若接入方关系链已经导入至腾讯云 IM 侧，也可以提工单配置默认发送消息权限为仅好友关系才能收发消息。

App 管理员发送单聊消息

App 内管理员可以给任意用户发送单聊消息。

只允许给好友发送消息

目前 ImSDK 以及 Server 默认情况下在发消息时不会对关系链进行判断，需要 App 在本地获取关系链或者通过接口判断是否好友决定是否发送。

若接入方关系链已经导入至腾讯云 IM 侧，也可以通过到云通信控制台相应 SDKAPPID【应用配置】>【功能配置】>【单聊消息检验关系链】设置开启，将发送消息权限为仅好友关系才能收发消息。

拒绝来自某人的消息

如果想要拒绝来自某人的消息，可以用黑名单进行控制。加用户为黑名单之后，对方发消息会显示成功，但消息会被丢弃，接收方无法接收消息。另外，在加用户黑名单时，如果之前是好友关系，会自动解除好友。目前提供以下两种设置黑名单方式：

- [通过 ImSDK 设置黑名单](#)。
- [通过 REST API 设置黑名单文档](#)。

获取聊天记录

发消息之后回调

如果要获取实时聊天记录，可以通过消息回调，选择在单发消息后回调，获取消息并记录。详细请参阅 [单发消息后回调](#)。

消息记录下载

如不需要实时获取记录，可以通过接口定时下载记录，消息记录接口和存储时长，可参见接口说明中的描述。详细请参阅 [消息记录下载](#)。

多终端同步

注：该功能目前处于内测中。

默认情况下，无法同时登录多个终端，通过提单配置后，可以允许多平台同时登录，会在多个终端尽量保持消息的完整性，另外，在默认情况下，用户在多个终端之间切换登录，也无法保证所有终端消息的一致性，需要提需求工单进行配置。

配置后，可实现在多个终端切换，或者同时登录多个终端的情况下，保持消息内容的一致性（由于 Server 存储消息的时长以及空间问题，只能保证大多数情况下一致性）。

单聊消息的离线推送

APNs 离线推送

ImSDK 默认支持 APNs 离线推送，在控制台上传对应推送证书，客户端上报 Token 后，可以收到苹果推送。通过自定义 TIMCustomElem 的 Ext 字段可以指定推送内容的 Ext 字段，方便用户单击后跳转；通过 sound 字段，可以自定义某条消息的声音。可参见 appledoc 说明。

如果需要自定义接收声音，可以调用接口设置单聊推送声音。APNs 离线推送形式为：昵称：内容。详细请参阅[iOS 客户端推送设置](#)。

Android 离线推送

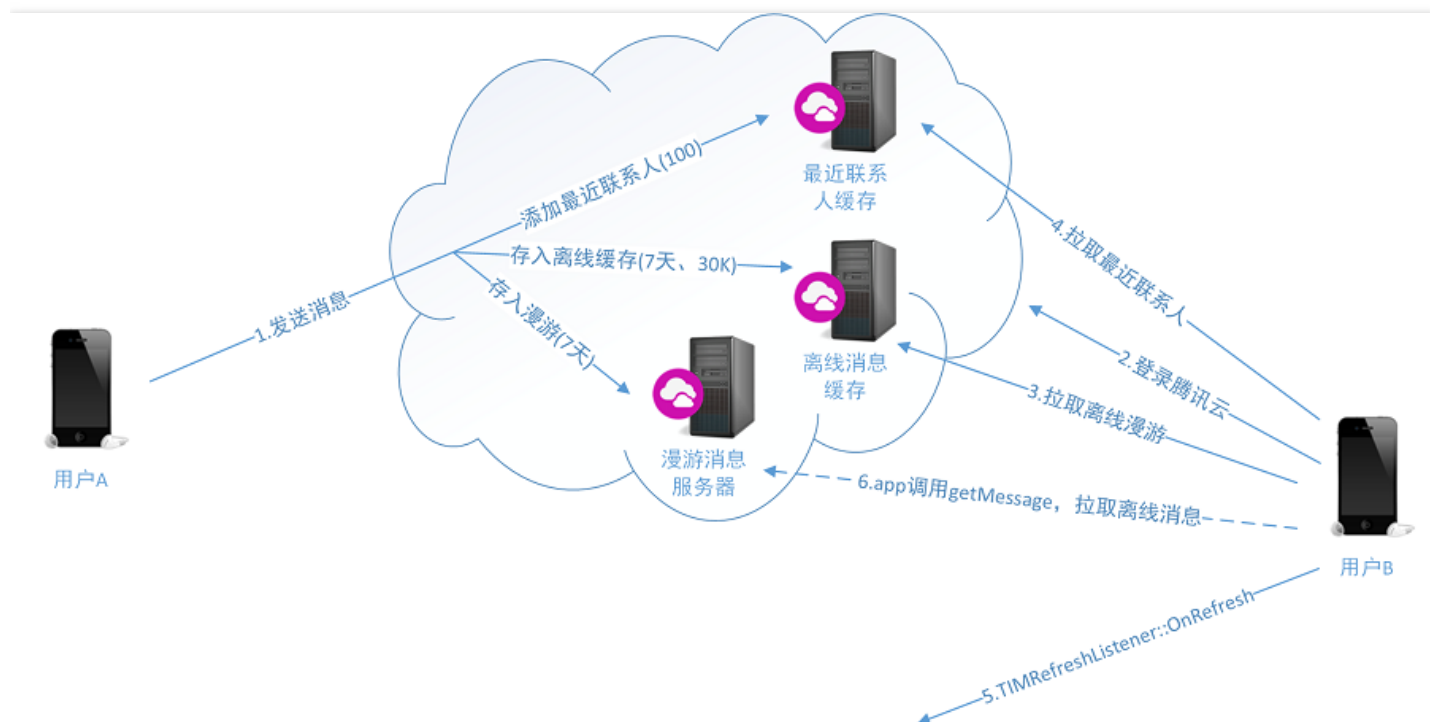
在 Android 1.8.0 以后版本，ImSDK 支持离线推送。

单聊消息中携带发送者资料

目前单聊消息中并未携带发送者昵称，如果需要可通过方法 `GetUserProfile` 获取昵称和头像等资料，如果是好友关系，可以通过 `GetFriendProfile` 获取好友相关的资料，比如备注、分组等信息。详细请参阅以下接口描述：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows 用户资料](#)
- [Web SDK 用户资料](#)

单聊离线消息处理流程



单聊消息离线及漫游处理流程：

1. 用户 A 调用 `sendMessage` 给用户 B 发送消息，用户 B 处于下线状态。
 - 1.1 把用户 A 添加进用户 B 的最近联系人，缓存大小为 100 条。
 - 1.2 把消息存入离线缓存中，缓存大小 30K，时间限制 7 天。
 - 1.3 把消息存入漫游服务器中，时间限制 7 天。
2. 用户 B 调用 `login` 接口登录腾讯云。
3. SDK 自动拉取离线缓存中的消息，通过 `OnNewMessage` 抛出。
4. SDK 自动拉取最近联系人，通过 `OnNewMessage` 接口抛出。
5. 同步消息过程完成，通过 `OnRefresh` 接口通知用户已完成消息同步。
6. 用户调用 `getMessage`，如果本地消息不完整，SDK 自动拉漫游服务器。

批量发消息与全员消息推送

最近更新时间：2018-05-31 15:52:15

应用场景

在 App 的日常运营过程中，经常需要向一大批用户甚至全部用户发送消息。针对不同的应用场景，我们提供了如下几种解决方案：

- 向一批用户发送消息（指定帐号）。
- 向 App 中的所有用户发送消息。
- 向 App 中符合特定条件的用户发送消息。

注：批量发消息的接口均不会触发第三方回调。

向一批用户发送消息

详细请参阅 [REST API：批量发单聊消息](#)。

向 App 中的所有用户发送消息

详细请参阅 [消息推送服务](#)。

向 App 中满足特定条件的用户发送消息

详细请参阅 [消息推送服务](#)。

ImSDK 接收推送消息

ImSDK 接收消息都是通过统一回调接口 `onNewMessage` 返回，通过 `onNewMessage` 回调获得消息后，能根据消息获取相应的会话。详细请参阅[请下接口中的描述](#)：

- [Android 新消息通知](#)
- [iOS 新消息通知](#)

-
- [Windows 新消息通知](#)
 - [WebSDK 初始化新消息回调](#)

群组系统

群组管理综述

最近更新时间：2018-05-31 15:52:36

本文档主要介绍了群组的创建、群资料管理、群成员管理。群组的整体综述参见[群组系统介绍](#)，群消息相关内容参见[群组消息综述](#)。

创建群组

通过 ImSDK 创建群组

- [Android 创建群组](#)
- [iOS 创建群组](#)
- [Windows C++ 创建群组](#)
- [Web 创建群组](#)

通过 REST API 创建群组

- [REST API：创建群组](#)。

自定义群组 ID

创建群组时如果不指定群组 ID，则云通信后台会默认分配一个群组 ID。默认分配的群组 ID 以 @TGS# 开头。如果指定了群组 ID，且该群组 ID 尚未被占用，则会分配成功。如果该群组 ID 已被占用，会创建失败。自定义群组 ID 必须为可打印 ASCII 字符（0x20-0x7e），最长 48 个字节，且前缀不能为 @TGS#（避免与默认分配的群组 ID 混淆）。

创建群组的配额限制

云通信目前限制每个 App 每天最多净消耗 5000 个群组，App 可以通过提需求工单来突破这一限制。所谓净消耗，是指创建群组的数量减去解散群组的数量。例如，创建 100 个群组，解散 20 个，则净消耗为 80 个。目前云通信后台没有限制每个用户可以创建的群组的数量。

创建群组之后的 ImSDK 系统消息

创建群组之后，群主所在终端会收到创建群组系统消息。该系统消息的目的在于保持多终端群组同步（一个终端创建群组之后，所有终端都能立即感知到该群已被创建）。

参见：

- [Android 创建群系统消息](#)

- [iOS 创建群系统消息](#)
- [Windows C++ 创建群系统消息](#)
- [Web 创建群系统消息](#)

创建群组相关的第三方回调

- [创建群组之前回调](#)：云通信后台在收到创建群组请求之后，向 App 后台发起第三方回调，App 后台可以据此干预用户创建群组的动作（例如拒绝用户的建群请求）；
- [创建群组之后回调](#)：云通信后台完成一个群组的创建之后，向 App 后台发起第三方回调，App 后台可以据此进行必要的同步。

转让群组

通过 ImSDK 转让群组

只有群主才具备转让群组的权限，参见：

- [Android转让群组](#)
- [iOS转让群组](#)
- [Windows C++转让群组](#)
- [Web 转让群组](#)（文档完善中）

通过 REST API 转让群组

- [REST API：转让群组](#)

转让群组之后的 ImSDK 系统消息

文档完善中

转让群组相关的第三方回调

目前转让群组暂时无法触发第三方回调。

解散群组

通过 ImSDK 解散群组

- [Android 解散群组](#)
- [iOS 解散群组](#)
- [Windows C++ 解散群组](#)
- [Web 解散群组](#)

通过 REST API 解散群组

- [REST API：解散群组](#)。

解散群组之后的 ImSDK 系统消息

当群组被解散之后，群组原有成员均会收到解散群组的系统消息，参见：

- [Android 解散群系统消息](#)
- [iOS 解散群系统消息](#)
- [Windows C++ 解散群系统消息](#)
- [Web 解散群系统消息](#)

群组解散相关第三方回调

- [群组解散之后回调](#)：群组解散之后，云通信后台会给 App 后台发起回调，App 后台可以据此进行必要的数据同步。

群组资料管理

所谓群组资料，是指单个群组维度的属性，包括群名称、简介、公告、群主等，以及 App 的群组维度自定义字段。群成员相关信息参见群成员管理。

通过 ImSDK 获取/修改群组资料

通过 ImSDK 获取群组资料，参见：

- [Android 获取群组资料](#)
- [iOS 获取群组资料](#)
- [Windows C++ 获取群组资料](#)
- [Web 获取群组资料](#)

通过 ImSDK 修改群组资料，参见：

- [Android 修改群组资料](#)
- [iOS 修改群组资料](#)
- [Windows C++ 修改群组资料](#)
- [Web 修改群组资料](#)

（注意：非群成员不能调用获取群组资料相关接口，如果非群成员需要精确搜索一个群组，参见[依照群组 ID 精确查找](#)）

通过 REST API 获取/修改群组资料

- [REST API：获取群组详细资料](#)
- [REST API：修改群组基础资料](#)

群资料被修改之后的 ImSDK 事件消息

群名称、简介、公告、头像URL变更之后，云通信后台会给 ImSDK 推送事件消息，ImSDK 可以据此即时更新本地展示信息。参见：

- [Android 群资料变更事件消息](#)
- [iOS 群资料变更事件消息](#)
- [Windows C++ 群资料变更事件消息](#)
- [Web 群资料变更事件消息](#)

修改群组资料相关第三方回调

- [群组资料修改之后回调](#)：群基础资料变更之后，云通信后台会向 App 后台发起回调，App 可以据此进行必要的同步。

目前 App 可以在控制台上自助配置群名称、简介、公告、头像URL的变更回调。如果需要启用其他群资料（包括群维度的自定义字段）的变更回调，请提需求工单。

获取群成员资料

群成员管理包括以下几个方面：

1. 获取/修改其他群成员的信息，包括群成员的身份、入群时间、最后发消息时间、群名片，以及群成员维度的自定义资料；
2. 获取/修改自己在群组中的信息，这些信息仅仅可以由用户自己获取/设置，例如消息接收选项等。

通过 ImSDK 获取群组成员列表

- [Android 获取群成员列表](#)
- [iOS 获取群成员列表](#)
- [Windows C++ 获取群成员列表](#)
- [Web 获取群成员列表](#)

通过 REST API 获取成员列表

- [REST API：获取群组成员详细资料。](#)

修改群成员资料

修改群成员资料包括两方面：

1. 群主或者管理员修改**其他**群成员的资料，包括修改群内身份（设置/取消管理员）、禁言、修改群名片、群成员维度的自定义字段等；
2. 群成员主动修改**自己**在群内的资料，包括消息接收选项、群名片、群成员维度的自定义字段等。

通过 ImSDK 修改群成员资料

- [Android 修改群成员资料](#)
- [iOS 修改群成员资料](#)
- [Windows C++ 修改群成员资料](#)
- [Web 修改群成员资料](#)

通过 REST API 修改群成员资料

- [REST API：修改群成员资料](#)

邀请加群/拉人入群

邀请加群和拉人入群都是希望将其他用户拉入某一群组中。不同之处在于，拉人入群可以不经目标用户同意直接将其拉入群组，而邀请加群则需要用户同意。

默认情况下，私有群可以直接拉人入群；对于公开群、聊天室，只有群主和管理员能够邀请其他用户加群。

通过 ImSDK 邀请加群/拉人入群

参见：

- [Android 邀请入群](#)
- [iOS 邀请入群](#)
- [Windows C++ 邀请入群](#)
- [Web 邀请入群](#)

通过 REST API 拉人入群

App 管理员可以无需用户同意直接将其加入群组。参见：

- [REST API：增加群组成员](#)

拉人入群相关的 ImSDK 事件消息

拉人入群成功之后，所有群成员（包括刚刚被拉入的用户）都会收到用户被拉入群组的事件消息，参见：
成员变更通知。

邀请加群相关的 ImSDK 系统通知

被邀请的用户会收到邀请加群的系统通知，参见：

- [Android 邀请加群系统通知](#)
- [iOS 邀请加群系统通知](#)
- [Windows C++ 邀请加群系统通知](#)
- [Web 邀请加群系统通知](#)

被邀请的用户处理邀请加群的方法，参加：

- [Android处理邀请加群](#)
- [iOS处理邀请加群](#)
- [Windows C++处理邀请加群](#)

拉人入群/邀请加群相关第三方回调

- [拉人入群之前回调](#)：App 后台可以据此监控/拦截用户拉人入群/邀请加群的操作；
- [新成员入群之后回调](#)：如果直接拉人入群成功，云通信后台将发起该回调，App 后台可以据此进行必要的数据同步。

申请加群

所谓申请加群，即用户通过 ImSDK 主动加入某一群组的动作。申请加群可能遇到的结果如下：

1. 私有群不允许申请加群，会直接返回错误；
2. 对于公开群和聊天室，申请加群的处理结果由群资料中的 ApplyJoinOption 字段决定的：
 - i. DisableApply 表示禁止任何人申请加入；
 - ii. NeedPermission 表示需要群主或管理员审批（公开群默认配置如此）；
 - iii. FreeAccess 表示允许无需审批自由加入群组（聊天室默认配置如此）。

通过 ImSDK 申请加群

参见：

- [Android 申请加群](#)
- [iOS 申请加群](#)
- [Windows C++ 申请加群](#)
- [Web 申请加群](#)

处理申请加群

管理员处理加群申请，参见：

- [Android 处理申请加群](#)
- [iOS 处理申请加群](#)
- [Windows C++ 处理申请加群](#)

申请加群相关的第三方回调

- [申请入群之前回调](#)：在用户发起申请加群请求之后、云通信进行后续处理（直接入群或者给群组管理员下发申请加群通知）之前，云通信可以先将该请求发送给 App 后台，App 后台可以据此干涉用户的加群请求（例如拒绝掉本次申请加群请求）。
- [新成员入群之后回调](#)：如果群组允许任何人自由进入（例如聊天室默认配置如此），则云通信后台会在用户成功加群之后将该事件同步给 App 后台，App 后台可以据此进行必要的同步。

删除群组成员

所谓删除群组成员，是指群主或者群管理员将群成员从群组中移除的操作。App 后台管理员亦可通过 REST API 强制将某一用户从某一群组中删除。

通过 ImSDK 删除群组成员

参见：

- [Android 删除群组成员](#)
- [iOS 删除群组成员](#)
- [Windows C++ 删除群组成员](#)
- [Web 删除群组成员](#)

通过 REST API 删除群组成员

参见：

- [REST API：删除群组成员](#)。

删除群组成员相关的 ImSDK 事件消息/系统通知

当管理员或群主将用户从群组中删除之后：

被删除的用户会收到被踢出群组的**系统消息**，参见：

- [Android 被踢出群组系统消息](#)
- [iOS 被踢出群组系统消息](#)

- Windows C++ 被踢出群组系统消息
- Web 被踢出群组系统消息

群内其他成员会收到用户被踢出群组的事件消息，参见：
[成员变更通知](#)。

删除群组成员相关的第三方回调

- [群成员离开之后回调](#)：成员主动退群之后，云通信后台可以将该事件同步给 App 后台，App 后台可以据此进行必要的数据库同步。

主动退群

通过 ImSDK 主动退群

参见：

- [Android 退出群组](#)
- [iOS 退出群组](#)
- [Windows C++ 退出群组](#)
- [Web 退出群组](#)

主动退群相关的 ImSDK 事件消息/系统通知

用户主动退群之后：

主动发起退群操作的用户会收到主动退群系统消息，目的在于保持多终端同步，参见：

- [Android 主动退群退群系统消息](#)
- [iOS 主动退群退群系统消息](#)
- [Windows C++ 主动退群退群系统消息](#)
- [Web 主动退群退群系统消息](#)

群内其他成员会收到用户退出群组的事件消息，参见：
[成员变更通知](#)。

主动退群相关的第三方回调

- [群成员离开之后回调](#)。

获取用户所加入的群组

通过 ImSDK 获取用户自身所加入的群组

- [Android 获取加入的群组列表](#)
- [iOS 获取加入的群组列表](#)
- [Windows C++ 获取加入的群组列表](#)
- [Web 获取加入的群组列表](#)

通过 REST API 获取用户所加入的群组

- [REST API：获取用户所加入的群组。](#)

群未决信息列表

- [Android 处理群未决信息](#)
- [iOS 处理群未决信息](#)
- [Windows C++ 处理群未决信息](#)

群组发现

依照群组 ID 精确查找

通过 ImSDK 精确查找群组，参见：

- [Android 获取群组公开资料](#)
- [iOS 获取群组公开资料](#)
- [Windows C++ 获取群组公开资料](#)
- [Web 获取群组公开资料](#)

（注意：非群成员不能调用获取群组资料相关接口）

通过 REST API 精确查找群组，参见：

- [REST API：获取群组详细资料](#)

获取 App 中的所有群组

仅 App 后台管理员才能获取 App 中的所有群组，普通用户无权限。参见：

- [REST API：获取 App 中的所有群组。](#)

群组消息综述

最近更新时间：2018-11-13 14:55:03

发送群消息

通过 ImSDK 发送群消息

用户在 App 内向自己所加入的群组发送群消息，群内所有用户均可接收到消息。即使用户不在线，上线之后仍可以拉取到历史消息。

参见：

- [Android 消息收发](#)
- [iOS 消息收发](#)
- [Windows C++ 消息收发](#)
- [Web 消息收发](#)

通过 REST API 发送群消息

App 管理员在群组中发送普通消息，同时可以指定发送者。适用于通过 App 后台向群组发送消息的场景。

参见：

- [REST API：在群组中发送普通消息。](#)

通过 REST API 发送系统通知

App 管理员在群组中发送系统通知，只有群内在线成员才会收到该消息。不在线用户即使上线也无法收到该消息。

参见：

- [REST API：在群组中发送系统通知。](#)

系统通知只有 App 后台管理员才能发送。系统通知的可靠性弱于普通群消息，若无特殊需求，请尽量发送群消息而非系统消息。更多细节参见 [群组内普通消息与系统消息的区别](#)。

在群消息中携带发送者相关信息

群消息在下发给客户端时，除了会携带消息内容本身，还会携带消息发送者的信息，默认会携带如下字段：发送者的昵称、发送者的头像、发送者在群内的群名片。如果您的 App 在云通信中配置了用户维度的自定义字段、[群维度的自定义字段](#)、[群成员维度的自定义字段](#)，亦可在消息中携带这些信息，但自定义字段不会默认携带，需要提需求工单。

如果要在消息中携带发送者昵称、头像，必须将这两个信息导入云通信的用户资料。

通过 App 设置用户资料，参见：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows C++ 用户资料](#)
- [Web 用户资料](#)

通过 REST API 查询/设置用户资料，参见：

- [REST API：拉取用户资料](#)
- [REST API：设置用户资料](#)

除此之外，使用独立模式的 App 还可以在通过 REST API 导入帐号的同时设置用户资料，参见：

- [REST API：独立模式帐号同步](#)

接收群消息

App 接收群实时消息

ImSDK 接收消息都是通过统一回调接口 onNewMessage 返回，通过 onNewMessage 回调获得消息后，能根据消息获取相应的会话。

参见：

- [Android 新消息通知](#)
- [iOS 新消息通知](#)
- [Windows 新消息通知](#)
- [WebSDK 初始化新消息回调](#)

App 获取群离线/历史消息

在 ImSDK 接收到消息后，默认情况下会对消息进行缓存和存储（如果不需要，可以调用 disableStorage 接口进行存储禁用，提升性能），各个终端通过调用 getMessage 接口可以聊天记录。另外，私有群和公开群的群消息默认有 7 天漫游功能，getMessage 接口同时会通过网络获取漫游的历史消息。

参见：

- [Android 获取消息](#)
- [iOS 获取消息](#)
- [Windows 获取消息](#)

App 获取群消息中携带的发送者信息

默认在群消息中带有发送方的资料以及群成员资料，在 1.9.0 以后版本可以通过接口获取相关资料。

- [Android 获取发送者信息](#)
- [iOS 获取发送者信息](#)
- [Windows 获取发送者信息](#)

离线消息与历史消息

云通信后台默认会对私有群和公开群产生的消息保存 7 天。在 7 天之内，群成员可以获取历史消息（即使用户已经读取过群消息，其依然可以拉取历史消息）。如果 App 希望延长群消息的保存时间，可以提需求工单。

群消息接收选项

群消息支持三种消息屏蔽选项，同一个用户在不同的群组中可以设置不同的消息接收选项：

1. 接收并提示（AcceptAndNotify）；
2. 接收不提示（AcceptNotNotify）；
3. 屏蔽消息（Discard）。

接收并提示消息为默认情况，如果设置了接收不提示，对于 iOS 终端来说，会去掉 APNs 推送；进入 App 内部之后，接收不提示跟接收提示行为一致，需要 App 根据用户设置的群成员资料进行判断并过滤（为了避免 App 层面不知道已经收到了新消息）。屏蔽消息指此群消息会在 Server 端进行屏蔽，不下发到客户端。

通过 App 修改消息接收选项，参见：

- [Android 修改群消息接收选项](#)
- [iOS 修改群消息接收选项](#)
- [Windows C++ 修改群消息接收选项](#)
- [Web 修改群消息接收选项（文档完善中）](#)

通过 App 获取消息接收选项，参见：

- [iOS 获取群消息接收选项](#)

App 后台通过 REST API 获取/修改某个用户在某个群内的消息接收选项，参见：

- [REST API：获取群成员详细资料](#)
- [REST API：修改群成员资料](#)

收发消息权限

一般情况下，只有群成员才能在群内收发消息，如果非群成员在群组中发送消息，会返回 10007 错误。但是存在特例：

1. App 管理员可以在不加入群组的前提下，通过[REST API：在群组中发送普通消息](#)向任意群组发送群消息；
2. App 管理员通过[REST API：在群组中发送普通消息](#)向群组中发消息时，如果“伪造”了消息发送者（From_Account），即使“伪造”的发送者不是群成员，消息依然会下发。

消息发送控制

如果 App 希望可以在群内下发的消息进行控制，有如下几种方式：

1. 禁言：禁止某个用户在群组内发言；
2. 脏字过滤：导入一批敏感词到云通信，云通信会拒绝下发包含敏感词的消息；
3. 发消息之前回调：在下发消息之前，云通信后台先去 App 后台询问是否允许下发消息，如果不允许，则拒绝下发消息；
4. 云通信后台进行定制化开发：给云通信提需求工单，由云通信后台协助 App 进行消息控制。

下面将依次进行介绍。

禁言

所谓禁言，即禁止某一用户一段时间内在群内发言。单个群组内的禁言仅针对单个群组生效。如果用户退群再重新入群，只要禁言时间没有过期，禁言依旧有效。

群主、管理员可以通过 App 进行禁言，参见：

- [Android 对群成员进行禁言](#)
- [iOS 对群成员进行禁言](#)
- [Windows C++ 对群成员进行禁言](#)
- [Web 对群成员进行禁言](#)

禁言之后的 ImSDK 相关事件通知（文档完善中）。

App 后台亦可通过 REST API 进行禁言，参见：

- [REST API：批量禁言和取消禁言](#)
- [REST API：获取群组被禁言用户列表](#)

脏字过滤

所谓脏字过滤，是指 App 可以在云通信后台配置一批敏感词，如果云通信后台在用户发送的消息中检测到这些敏感词，则拒绝下发该消息，并向发送者返回错误码 80001。

脏字的增删查改参见：

- [REST API : 查询 App 自定义脏字](#)
- [REST API : 添加 App 自定义脏字](#)
- [REST API : 删除 App 自定义脏字](#)

注意：

1. 云通信已经默认配置了政治类、色情类的脏字，能够满足这两方面的大多数过滤需求，App 只需要配置业务场景相关的脏字即可。例如，电商类 App 只需要配置“假货”、“刷单”等商业类脏字即可。
2. 默认情况下，云通信后台只会对消息格式中的文本消息进行脏字过滤（TIMTextElem，参见 [消息格式描述](#)），如果 App 使用自定义消息格式（TIMCustomElem，参见 [消息格式描述](#)），云通信后台无法进行过滤。因此，App 最好在设计之初就确保将需要进行脏字过滤的文本放到 TIMCustomElem 中。对于 App 已经使用了自定义消息格式，且已经上线，可以联系云通信客服，将消息格式提供给云通信的技术人员。云通信后台可以据此解析 App 的自定义消息中的文本信息，并进行脏字过滤。

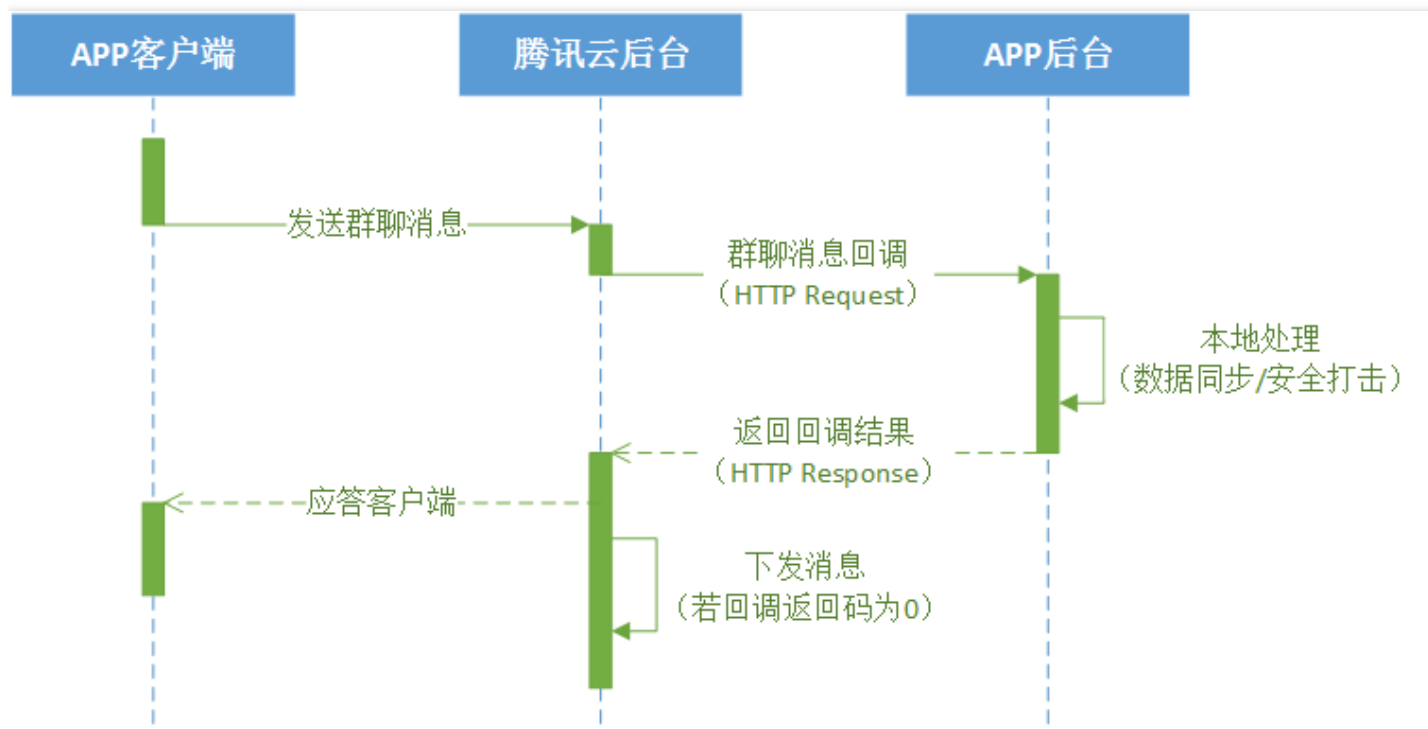
群内发言之前回调

所谓群内发言之前回调，即云通信后台在收到用户的发消息请求之后、将该消息下发给群成员之前，向 App 后台发起 http/https 回调，并根据 App 后台的应答结果来判定是否应当下发该消息。

除此之外，App 后台在收到回调请求之后，亦可对用户发送的消息内容进行修改并返回给云通信，云通信将使用修改之后的信息进行下发。

注意：在发起第三方回调时，云通信要求 App 后台必须在2秒之内给出应答，如果没有收到应答，云通信不会进行重试，会直接将消息下发给群成员。

整体逻辑如下图所示：



参见：

- [第三方回调简介](#)
- [第三方回调接入指引](#)
- [群内发言之前回调](#)

云通信后台进行定制化开发

如果上述方案均无法满足您的需求，或者 App 已经上线，发现难以解决的问题，则可以给云通信提需求工单，对 App 的消息进行定制化的控制（例如依照特定正则表达式进行过滤）。

消息优先级与频率控制

群消息优先级

优先级选择

群消息分为 4 个优先级：High/Normal/Low/Lowest。如果某个群的发消息请求量超过了频率限制，后台会优先下发优先级相对更高的消息。用户应根据消息的重要程度来选择优先级，比如：

- 红包消息和礼物消息使用 High 优先级；
- 普通文本消息使用 Normal 优先级；
- 点赞消息使用 Low 优先级；
- 其它最不重要的使用 Lowest 优先级；

优先级设置

- [Android SDK 设置群消息优先级](#)
- [iOS SDK 设置群消息优先级](#)
- [通过 REST API 发送群内普通消息设置优先级](#)

群消息频控

总消息数频控

总消息数频控是指单个群每秒最多能发送的消息数限制，默认值为 40 条/秒。频控对象包括客户端发送的群消息和[通过 REST API 发送的普通消息](#)，但不包括[通过 REST API 发送的系统通知](#)。

当在一个频控周期内单个群的发消息请求超过了频控限制时，后台优先下发优先级相对较高的消息，同等优先级的消息随机排序。

被频控策略所限制的消息不会下发；不产生 [群消息 seq](#)；不会进入[历史消息](#)；会触发[群内发言之前回调](#)；不会触发[群内发言之后回调](#)；发送方会得到发送成功的返回包。

优先级频控

优先级频控是指单个群每秒最多能发送多少条某优先级的消息，发消息请求只有在通过总消息数频控之后，才会进入优先级频控。

App 管理员/群主/群管理员发送的消息不受优先级频控限制；High 优先级的消息不受优先级频控限制，其余 3 个优先级可以独立配置，默认都为 40 条/秒。

一个 sdkappid 下的每个[群组形态](#)都有自己的总消息数频控和优先级频控配置。如果需要修改默认值，请根据[工单模板](#)提交工单进行申请。

成员变更通知

成员变更通知是指：当群组中有成员加入（包括申请加入和被邀请加入）或者退出（包括被踢出群和主动退群）时，云通信后台会给群组中的所有成员广播一条消息来通知这一事件。

对于用户主动申请加群和被邀请加群，群内用户会收到 ImSDK 事件消息，参见：

- [Android 用户加群通知](#)
- [iOS 用户加群通知](#)
- [Windows C++ 用户加群通知](#)
- [Web 用户加群通知](#)

对于用户主动退群，群内用户会收到 ImSDK 事件消息，参见：

- [Android 用户退群通知](#)
- [iOS 用户退群通知](#)
- [Windows C++ 用户退群通知](#)

- [Web 用户退群通知](#)

对于用户被踢出群，ImSDK 相关事件通知参见：

- [Android 用户被踢出群组通知](#)
- [iOS 用户被踢出群组通知](#)
- [Windows C++ 用户被踢出群组通知](#)
- [Web 用户被踢出群组通知](#)

不论是哪一种群组形态，ImSDK 中接收成员变更通知的接口都是相同的。但是，不同群组形态的成员变更通知却有着不同的表现形式。云通信目前支持三种成员变更通知形式：

1. 无提醒：群成员变更时，不下发成员变更通知（聊天室默认配置如此）；
2. 弱提醒：群成员变更时，以系统消息的形式广播成员变更通知：此类通知只会触达群组中的在线成员，它们不占用消息 SEQ，不保存于消息记录，不计入用户的未读消息计数，因此不在线用户重新上线之后也看不到该消息；
3. 强提醒：群成员变更时，以普通消息的形式广播成员变更通知：群组后台为其分配消息 SEQ，会将其保存在后台消息记录中，并计入用户的未读消息计数中，不在线成员重新上线之后可以看到该消息（私有群、公开群默认如此）。

群消息的APNs推送

当用户在群内的消息接收选项为“接收并提示”时，如果群内有新消息，iOS 客户端会收到 APNS 推送。基本的展示形式是：

发送者昵称/群名片(群名称)：消息内容

其中群名片的展示优先级大于昵称。下面给出几个具体示例：

示例 1：群名称为“xxx业主群”，发送者昵称为“张三”，未设置群名片：张三(xxx业主群)：消息内容

示例 2：群名称为“xxx业主群”，发送者昵称为“张三”，设置群名片为“2栋308业主”：2栋308业主(xxx业主群)：消息内容

示例 3：群名称为“xxx业主群”，发送者ID为ID_yyy，发送者未设置昵称，也未设置群名片：ID_yyy(xxx业主群)：消息内容

有关离线推送的详细介绍，请参考 [离线推送 \(ISO SDK\)](#)。

App 后台获取群消息记录

App 后台可以通过如下三种方式获取群组消息：

1. 所谓消息记录下载，是指云通信会将 App 在一段时间内产生的所有消息打包成文件，App 可以通过 [REST API：消息记录下载](#) 来获取消息记录。该方案适用于 App 定期备份消息记录的场景。
2. 所谓 [群内发言之后回调](#)，是指云通信后台会在完成群消息下发之后，将该消息同步给 App 后台。与 [群内发言之前回调](#) 相比，该回调不会干涉 App 正常的消息下发流程。
3. 可以通过 [REST API：拉取群漫游消息](#) 来获取单个群组的历史消息。

建议：

1. 如果 App 只是需要定期备份群消息，则只需要使用 [REST API：消息记录下载](#)；
2. 如果 App 需要实时获取 App 中的群消息，需要开通 [群内发言之后回调](#)；
3. 如果 App 需要快速获取指定群组的历史消息，建议使用 [REST API：拉取群漫游消息](#)。

消息删除

所谓消息删除，是指一旦 App 后台发现有人在群内发布了恶意信息（并且部分成员已经看到），但为了降低影响，可以将该消息从历史消息中删除，从而确保尚未看到该消息的用户无法看到该消息。

参见：

- [REST API：删除指定用户发送的消息](#)。

群消息的 SEQ 机制

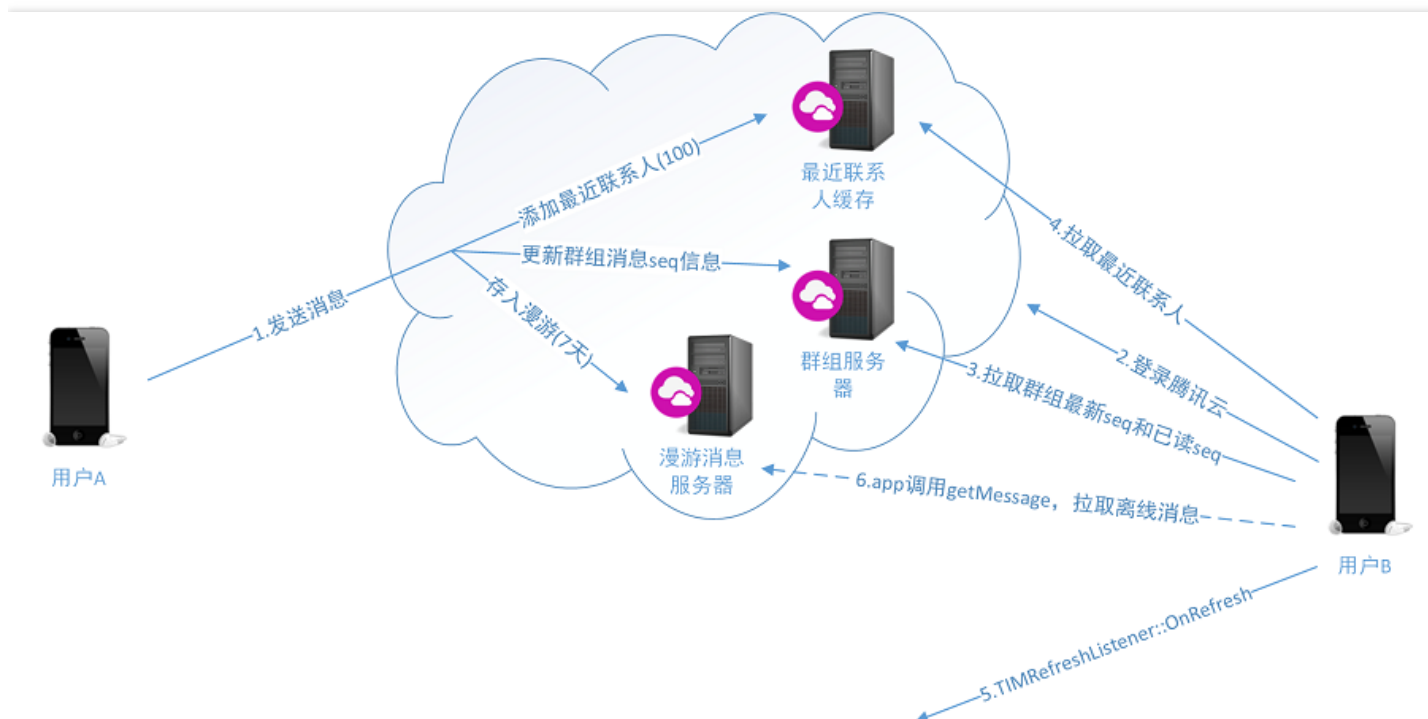
云通信会为每个群维护一个消息 SEQ。SEQ 的初始值为 1。群内每产生一条普通消息，云通信后台会将当前 SEQ 的值作为该消息的 SEQ，并且将该 SEQ 自增 1。

群组 ID + SEQ，相当于是是一条消息的唯一标识。

特别注意的是：

云通信只会为普通消息产生 SEQ，不会为系统消息产生 SEQ；

群组离线消息处理流程



群组离线消息处理流程：

1. 用户 A 调用 sendMessage 给群组 C 发送消息，用户 B 处于下线状态。
 - 1.1 把群组 C 添加进用户 B 的最近联系人，缓存大小为 100 条。
 - 1.2 用户更新群组的消息信息，包括群组最新消息 seq。
 - 1.3 把消息存入漫游服务器中，时间限制 7 天。
2. 用户B调用 login 接口登录腾讯云。
3. SDK 自动拉取所有群组的消息 seq 信息，包括最新消息 seq 和未读计数。
4. SDK 自动拉取最近联系人，通过 OnNewMessage 接口抛出。
5. 同步消息过程完成，通过 OnRefresh 接口通知用户已完成群组数据同步。
6. 用户调用 getMessage，SDK 自动拉漫游服务器。

用户资料和好友关系综述

最近更新时间：2018-05-31 16:26:06

用户资料介绍

用户基础资料

默认目前用户资料基础字段支持四项：

- 昵称：用户昵称，只有自己和管理员可修改。
- 头像 URL：用户头像的 URL，目前 IM 通讯云不保存用户头像，需要上传至其他存储服务，通过 ImSDK 设置 URL。
- 加好友验证方式：分三种方式，需要确认才能添加好友、所有人可添加好友、任何人不可以添加好友。默认为所有人可添加。
- 个性签名：用户个性签名，只有自己和管理员可修改。

用户自定义资料

另外，默认字段无法满足所有用户的需求，另外有添加自定义字段的功能，自定义资料字段是各 App 根据各自业务需要而设置的用户数据。通过自定义资料，各 App 可以将一些额外数据附加到用户资料上，并通过现有接口进行读写操作。目前此功能为内测功能，需要提工单增加。

自定义资料字段的申请

各 App 需要申请自定义资料字段时，可以向腾讯云客服提交需求工单，工单内容可参考：[新增用户维度的自定义字段](#)。需求工单提交后，腾讯云将在三个工作日内完成配置并生效。

申请自定义资料字段时，需要为每一个自定义资料字段提交如下资料：

1. 自定义资料字段的名称（Key）：详情可参见 [自定义资料字段的命名规范](#)；
2. 自定义资料字段的类型（Value）：详情可参见 [资料字段](#)。

自定义资料字段的命名规范

自定义资料字段的命名规则如下：

1. 自定义资料字段的名称分为三部分：前缀、SDKAppId 和关键字；
2. 自定义资料字段的前缀是：Tag_Profile_Custom；
3. 关键字：必须是英文字母，且长度不得超过8字节，建议用一个英文单词或该英文单词的缩写；
4. 示例：某 App 申请的 SDKAppId 是 888888，关键字是 Test，则自定义资料字段的名称是：
Tag_Profile_Custom_888888_Test。

关系链介绍

关系链特性

目前云通信支持加好友（收发消息并不进行好友关系判断），还支持对好友的备注和分组。如果不想收到某人的单聊消息，黑名单：

- 好友数量限制 1000 个；
- 黑名单数量限制 1000 个；

标配好友字段

目前云通信支持的标配好友字段如下：

- 好友分组（同一个好友可以有多个不同的分组）
- 好友备注
- 加好友来源（申请加好友时填写）
- 加好友附言（申请加好友时填写）

自定义好友字段

自定义好友字段是各 App 根据各自业务需要而设置的好友数据。通过自定义好友字段，各 App 可以将一些额外数据附加到好友上，并通过现有的接口进行读写操作。

各 App 需要申请自定义好友字段时，可以向腾讯云客服提交需求工单，工单内容可参考：[新增用户维度的自定义字段](#)。需求工单提交后，腾讯云将在三个工作日内完成配置并生效。

自定义好友字段的命名规则如下：

1. 自定义好友字段的名称分为三部分：前缀、SDKAppId 和关键字；
2. 自定义好友字段的前缀是：Tag_SNS_Custom；
3. 关键字：必须是英文字母，且长度不得超过8字节，建议用一个英文单词或该英文单词的缩写；
4. 示例：某 App 申请的 SDKAppId 是 888888，关键字是 Test，则自定义关系字段的名称是：
Tag_SNS_Custom_888888_Test。

申请自定义好友字段时，需要为每一个自定义好友字段提交如下资料：

1. 自定义好友字段的名称（Key）；
2. 自定义好友字段的类型（Value）：详情可参见 [关系链字段](#)。

设置用户资料

用户可以设置如昵称、头像等资料，当前只能设置用户自己的资料，Server 端使用管理员身份可以设置任意用户资料。

设置昵称

昵称为公开资料，用户可以设置自己的昵称，对所有用户可见。

参见：

- [Android 设置自己的昵称](#)
- [iOS 设置自己的昵称](#)
- [Windows 设置自己的昵称](#)
- [WebSDK 设置自己的昵称](#)

设置好友验证方式

用户可以设置加好友的方式，如，同意任何人加好友，需要验证，拒绝任何人加好友。默认情况为允许任何人加好友。设置完成后，用户申请自己加好友时，会根据设置做相应的操作。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)

设置头像

目前 ImSDK 不会存储头像，需要 App 把头像先上传到其他存储平台，然后在 ImSDK 中设置用户的头像 URL。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)

设置个性签名

用户可以设置个人签名，所有用户可见。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)

设置自定义资料

目前 ImSDK 只有几种内置资料，显然无法满足所有应用的需求，ImSDK 增加了自定义字段，用户可根据自身的需求设置自己想要的资料类型。（目前此功能还在内侧，需要提单修改后才能使用）。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)

设置好友资料

除了用户资料设置，在加了好友之后，可以设置好友相关属性，如好友备注，好友分组等。

设置好友备注

备注可以让用户自定义好友的显示名称，仅自己可见。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)

设置好友自定义字段

除了内置属性，用户还可以设置自定义字段，根据自己的需要申请 [自定义好友字段](#)。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)

获取资料

获取所有用户资料

通过接口可以获取一批用户的资料，不用关心是否好友。

在 iOS，Android，Windows 终端，1.9 版本之前，可以使用 GetFriendsProfile 获取，在 1.9 版本之后，需要调用 GetUsersProfile 获取。

参见：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows 用户资料](#)
- [WebSDK 用户资料](#)
- [Rest API](#)

获取关系链相关资料

通过 `GetUsersProfile` 只能获取用户的公开资料，而关系链相关的资料，如备注和分组，无法获取，在1.9版本之前，并没有接口获取指定用户的接口，在 1.9 版本之后，可以使用 `GetFriendsProfile` 指定一批好友获取用户的资料和关系链资料。

参见：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows 用户资料](#)

好友关系链操作

好友关系链操作网络异步调用

ImSDK 提供了加好友、删除好友、获取好友列表、检查是否好友的功能，所有接口都是异步网络操作。

参见：

- [Android 关系链](#)
- [iOS 关系链](#)
- [Windows 关系链](#)
- [WebSDK 关系链](#)
- [Rest API](#)

好友关系链本地存储

在 Android，iOS，Windows 终端，如果开启本地存储功能会在本地进行存储和缓存数据，可通过同步接口从缓存获取显示。因为效率考虑，此功能默认不开启，需要调用接口开启。

参见：

- [Android 关系链](#)
- [iOS 关系链](#)
- [Windows 关系链](#)

好友分组

好友分组相关操作，如添加分组，删除分组，分组改名，获取分组内好友，添加好友到某个分组，可参考对应平台文档：

参见：

- [Android 关系链](#)
- [iOS 关系链](#)
- [Windows 关系链](#)

未决请求和推荐

应用场景

当有用户申请加好友，并且当前用户设置了需要审核，会产生一条未决请求（未决即为需要用户处理，悬而未决的消息），未决请求可以通过系统消息通知感知到，1.9版本以后可以通过注册相关回调感知到。另有异步网络接口获取所有未决和已决消息。另外，用户可以通过server接口设置用户推荐（内测功能，未文档），设置了某个好友的推荐好友，客户端可以拉取到此信息。

操作接口

- [Android 未决请求](#)
- [iOS 未决请求](#)
- [Windows 未决请求](#)

黑名单

应用场景

如果用户想屏蔽某人的消息，可以使用黑名单功能，当加入某人为黑名单时，如果之前是好友关系，自动解除好友。加入黑名单后，对方发消息能够显示成功，但消息会被丢弃，接收方无法收到。

黑名单操作

用户可以调用接口获取黑名单列表，增加删除黑名单。

参见：

- [Android 设置资料](#)
- [iOS 设置资料](#)
- [Windows 设置资料](#)
- [WebSDK 设置资料](#)
- [Rest API](#)

好友列表的更新

未启用资料存储更新时机

客户端 SDK 1.9 版本之前，不支持本地存储用户资料和关系链，需要 App 层面进行缓存和存储，好友的更新通过系统消息来通知。收到系统消息时，可以调用接口重新获取关系链和资料进行更新。

参见：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows 用户资料](#)

启用资料存储更新时机

客户端 SDK 1.9 版本之后，支持本地存储，如果启用此功能（默认不开启），客户端本地会对关系链和资料进行存储，用户可以通过同步接口获取资料和关系链，同时可以设置变更回调，当收到变更回调时，选择刷新界面。

参见：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows 用户资料](#)

搜索用户

云通信支持按照用户的昵称进行模糊检索。用户设置昵称后，可以通过检索接口查询。

参见：

- [Android 用户资料](#)
- [iOS 用户资料](#)
- [Windows 用户资料](#)

接入服务系统（服务端集成）

最近更新时间：2018-04-04 11:39:14

接入服务—— HTTP 头部字段说明

App 通过 QAL SDK 发起 CS 请求之后，网络接入服务将该请求转发到 App 后台服务。App 后台收到的将是标准的 http/https 请求。为方便 App 后台获取请求方的各种信息，App 后台收到的 http 请求中，http 头部将会携带如下信息。

HTTP Header	说明	例子
X-Forwarded-For	用户的网关 IP	X-Forwarded-For: 14.17.43.51
QAL-Identifier	每个终端每个应用的唯一 ID	QAL-Identifier: @v_tls#tarqqnwfrcdx
QAL-Req-Seq-No	每个请求唯一序列号	QAL-Req-Seq-No: 39134
QAL-SDK-APP-Id	应用的 sdkAPPid	QAL-SDK-APP-Id: 1600190028
QAL-SDK-Version	QAL SDK 的版本	QAL-SDK-Version: 1.3.0
QAL-APP-Version	应用的版本	QAL-APP-Version: 5.3.1
QAL-APP-Channel	应用的渠道	QAL-APP-Channel: yingyongbao
QAL-APN	WiFi 网络值是 wifi，移动网络则是运营商的 APN	QAL-APN: wifi
QAL-RAT	无线网络接入技术(Radio Access Technology)，有效值：UNKNOWN、GPRS、EDGE、UMTS、CDMA、EVDO Rel. 0、EVDO Rev. A、1xRTT、HSDPA、HSUPA、HSPA、iDEN、EVDO Rev. B、LTE、eHRPD、HSPA+、CDMA1x、WCDMA、Undefine	QAL-RAT: LTE
QAL-RAT-Generation	第几代无线网络接入技术，有效值：2、3、4	QAL-RAT-Generation: 4
QAL-OS	操作系统，有效值：Android、iOS、OS X、Windows Phone	QAL-OS: Android

QAL-OS-Version	操作系统版本	QAL-OS-Version: 4.3
QAL-Device	设备型号	QAL-Device: Lenovo A788t
QAL-Brand	设备品牌	QAL-Brand: Lenovo
QAL-Screen-Width	屏幕宽度	QAL-Screen-Width: 480
QAL-Screen-Height	屏幕高度	QAL-Screen-Height: 854
QAL-IMEI	国际移动设备标识	QAL-IMEI: 865000026007591
QAL-IMSI	国际移动用户识别码	QAL-IMSI: 865000026007591

接入服务——缓存控制

网络接入服务支持标准的 HTTP 缓存机制，服务端通过 HTTP Header 相关字段控制终端缓存。

对于 GET 请求，如果服务端的 HTTP 响应里带了有效的 ETag、Last-Modified、Expires、Cache-Control、QAL-Cache-Control、QAL-Stale-Expires 头部，并且没有输出 Cache-Control: no-cache/no-store/must-revalidate、Pragma: no-cache 情况下，SDK 会将 HTTP 响应保存在终端本地，启用 HTTP 缓存机制。建议 HTTP 缓存机制仅对静态资源或用户无关的数据开启。

HTTP响应控制缓存方法：

1. 304 Not Modified

资源没有发生变更的情况下，无需重传资源，在节省流量的同时还能提升速度。

协议：

1) ETag

例子：

```
HTTP/1.1 200 OK
ETag: "568ddc6e-253"
```

2) Last-Modified

例子：

```
HTTP/1.1 200 OK
Last-Modified: Thu, 07 Jan 2016 03:33:02 GMT
```

QAL SDK 发起 HTTP 请求的时候会通过 If-None-Match、If-Modified-Since 将 ETag 或 Last-Modified 带上，服务端可实现 304。

相关标准：[RFC7232 - Hypertext Transfer Protocol \(HTTP/1.1\): Conditional Requests](#)

2. 缓存生命期

缓存在有效生命期内，将直接命中缓存并返回，不发起网络请求。

1) Cache-Control

设置缓存生命期为 60 秒，例子：

```
HTTP/1.1 200 OK
Cache-Control: max-age=60
```

2) Expires

指定缓存有效日期，例子：

```
HTTP/1.1 200 OK
Expires: Fri, 06 Jan 2017 03:35:08 GMT
```

相关标准：[RFC7234 - Hypertext Transfer Protocol \(HTTP/1.1\): Caching](#)

3. 缓存异步更新

在可容忍的生命期内，将先返回缓存，再异步发请求更新资源，下次请求将得到最新的资源。

1) Cache-Control

设置缓存生命期 15 秒可直接返回，60 秒内先返回缓存再异步更新资源，例子：

```
HTTP/1.1 200 OK
Cache-Control: max-age=60, stale-while-revalidate=15
```

2) QAL-Cache-Control (QAL扩展)

设置 15 秒内，可先返回缓存再异步更新资源，例子：

```
HTTP/1.1 200 OK
QAL-Cache-Control: max-stale-age=15
```

设置缓存生命期 15 秒可直接返回，60 秒内先返回缓存再异步更新资源，例子：

```
HTTP/1.1 200 OK
Cache-Control: max-age=15
QAL-Cache-Control: max-stale-age=60
```

3) QAL-Stale-Expires (QAL扩展)

指定日期内，可先返回缓存再异步更新资源，例子：

HTTP/1.1 200 OK

QAL-Stale-Expires: Wed, 04 Nov 2015 08:13:44 GMT

相关标准：[RFC5861 - HTTP Cache-Control Extensions for Stale Content](#)

在线状态综述

最近更新时间：2018-05-31 15:55:05

术语约定

在线

在线，是指客户端和云通信服务器保持有顺畅的 TCP 网络连接，客户端可以发消息给云通信服务端，也可以收到来自云通信服务端推送的消息。

App 启动后，客户端和云通信建立 TCP 长连接，云通信服务保存客户端的在线信息，如客户端的网络链路，客户端的平台版本等。App 长连接建立成功后，在运行过程中，IMSDK 会定时发送心跳来维持当前的在线状态。iOS 客户端和云通信的 TCP 链路断开后，还可以收到苹果推送的消息，但是还是处于不在线的状态。

上线

所谓上线，App 客户端和云通信服务建立 TCP 长链接的动作。

下线

所谓下线，包括两种场景：

1. App 客户端退出登录；
2. 云通信服务器检测到与 App 客户端的 TCP 长连接断开。服务器依赖心跳包超时判断，当客户端和服务端之间持续 400 秒没有心跳包时，云通信服务器认为该用户异常下线（Android 版本较多出现此情况）。

注意：在线、上线、下线这几个概念，唯一的判断标准是客户端与云通信后台是否保持网络链接。对于 iOS 平台，即使用户不在线，仍然有可能收到来自云通信后台的 APNs 推送。

查询用户在线状态

App 后台可以通过 [REST API：获取用户在线状态](#) 来查询一批用户的在线状态。

ImSDK 暂时无法获取用户的在线状态。

用户在线状态变更通知

云通信可以把用户上下线的事件通知给 App 后台，参见 [状态变更回调](#) 文档。

消息缓存和存储

最近更新时间：2018-05-31 15:55:22

离线消息存储

云通信支持离线消息，即当用户不在线时，有用户发送消息，当下次此用户登录时会收到离线消息，离线消息和在线消息通过相同的回调接口返回给 App 层，通过消息时间可判断消息产生时间。

消息接收

消息接收参考：

- [iOS 消息接收](#)
- [Android 消息接收](#)
- [Windows 消息接收](#)

Server 保存时长

离线消息默认保存 7 天，一定大小，即如果用户 7 天内没有登录过，再次登录的时候，不会拿到 7 天前的离线消息，另外由于会有一定大小限制，过多的离线消息会被冲掉。

Server 删除时机

默认情况下，当一个终端把离线消息拉回到本地，Server 便会删除消息。如果需要支持多终端的情况、或者更换终端还想要拿到未读的消息，需要自己管理未读，可参考各平台的未读计数管理部分，以及禁用自动已读上报：

- [iOS 未读计数](#)
- [Android 未读计数](#)
- [Windows 未读计数](#)

当禁用了已读上报，只有用户显示调用已读上报接口，Server 才会删除消息，否则保存到过期。这样，如果没有进行已读上报，更换终端，仍然可拿到未读的消息。

漫游消息存储

云通信支持消息漫游，即用户更换终端的情况下，也可以获取到跟某人的聊天记录，或者某个群内的消息。

获取漫游消息

漫游消息客户端通过 getMessage 接口获取。

参考：

- [iOS 获取会话漫游消息](#)
- [Android 获取会话漫游消息](#)
- [Windows 获取会话漫游消息](#)

漫游消息保存时长

默认情况下，单聊消息和群聊消息有 7 天漫游，超过漫游时长的消息会被删除。群聊消息漫游时长不支持修改，而延长单聊消息的漫游时长是增值服务，需要提交工单修改，可参考：[调整单聊消息漫游时长](#)。

最近联系人消息

概念

最近联系人即为跟用户联系过的人，产品形态类似 QQ 的最近联系人列表，可以展示最近跟用户联系过的用户以及最后一条消息。

获取最近联系人

客户端默认情况下会在登录时拉取最近联系人消息，如果本地之前没有存储过，会通过 onNewMessage 回调抛出来。

消息接收参考：

- [iOS 消息接收](#)
- [Android 消息接收](#)
- [Windows 消息接收](#)

禁用最近联系人

另外，使用最近联系人，登录时会消耗一些流量，以及在新终端可能会抛出会话的最后一条消息。如果不需要此功能，可以选择禁用：

- [iOS 禁用最近联系人](#)
- [Android 禁用最近联系人](#)
- [Windows 禁用最近联系人](#)

最近联系人限制

最近联系人默认存储最近 100 个联系人，但是保存时长跟最近联系人中的最后一条消息保存时间一致，比如默认如果超过 7 天跟联系人没有消息，最后一条消息会过期后便无法在最近联系人中获取到此用户。

App 本地存储

本地存储获取

默认情况下，ImSDK 内部会对收到的消息进行存储，无需用户进行存储。用户可调用接口获取本地消息（无网络操作）：

参考：

- [iOS 获取本地消息](#)
- [Android 获取本地消息](#)
- [Windows 获取本地消息](#)

另外，通过 getMessage 接口，也会获取本地消息，如果本地消息存在断层，会通过 [漫游消息](#) 补全。

本地消息删除

ImSDK 默认不会删除用户消息，如果有需要，需要用户显示调用接口删除：

删除会话参考：

- [iOS 删除会话](#)
- [Android 删除会话](#)
- [Windows 删除会话](#)

删除单条消息参考：

- [iOS 删除单条消息](#)
- [Android 删除单条消息](#)
- [Windows 删除单条消息](#)

禁用本地存储

使用本地存储会消耗磁盘以及 CPU 性能，在不需要存储的场景（如直播场景，更注重消息处理性能，也不关心历史消息），可选择禁用本地存储：

参考：

- [iOS 禁用本地存储](#)
- [Android 禁用本地存储](#)
- [Windows 禁用本地存储](#)

Server 消息删除

对于群消息，可以通过 Server 接口删除某人的消息，但只能删除当前 Server 的存储，如果用户已经把消息拉到本地，是没有办法删除用户客户端的消息存储的，只能保证删除 Server 消息之后再登录查看消息的成员看不到消息。

参考：

- [删除指定用户发送的消息](#)

多终端同时在线

最近更新时间：2018-05-31 15:55:42

概念介绍

互踢

默认情况，ImSDK 在同时登录多个终端（如同时登录 PC、Android）时，会进行互踢，只有最后一个登录的设备可以在线，之前登录的都会被踢下线，详细互踢逻辑可以参考文档：

[Android 用户状态变更](#)

[iOS 用户状态变更](#)

[Windows 用户状态变更](#)

同时在线

此功能目前内测阶段，需要提工单修改同时在线策略，可参考：[支持多终端同时在线](#)。通过配置可以做到 PC 端和手机端同时在线，或者 PC、iOS 和 Android 都可以同时在线，开启了同时在线登录不通终端不会互踢（但是两个相同终端，如两个 iOS 端登录仍然会互踢）。

客户端接口说明

如果有应用场景需要 PC 和手机同时在线（如 QQ 等）可以提交工单修改，提工单修改服务端配置后，即可进行多终端登录，无需客户端修改。

功能和限制因素

此功能仅能让多个终端登录，并不保证多端消息的完整性，如果有消息完整性方面的需求，可参阅：[消息缓存和存储](#)。

离线推送

最近更新时间：2018-07-26 11:03:56

概念

用户登录后退到后台，或者杀掉进程，当有用户给其发消息时，云通信支持离线推送，在 iOS 端会有苹果推送，Android 需要用户注册离线消息回调。

应用场景

在 App 退后台或者进程被杀的情况下，有新消息时需要提醒用户。可使用离线推送功能。

iOS APNs 推送

推送格式说明



上图为一条单聊消息和一条群聊消息的示例。

iOS APNs 推送格式详细说明可参考：[推送格式说明](#)

基本接口说明

支持 APNs 必须调用以下接口，具体参考 [iOS APNs 事件上报](#)：

- 设置 Token
- 切后台上报未读
- 切前台通知

Ext 扩展设置

有时应用需要根据情况设置推送的 Ext 扩展字段，方便用户单击跳转等操作，可以填写到 TIMCustomElem 中的 ext 字段，推送时 Server 会把该字段填入 Ext。参考 [扩展字段](#) 文档。

设置推送声音

有时应用需要根据情况设置单条消息的推送声音，方便特别提醒某类消息，可以把声音填写到 TIMCustomElem 中的 sound 字段，推送时 Server 会把该字段填入 Ext。参考 [设置推送声音](#) 文档。

单条消息声音设置

有时应用需要根据情况设置单条消息的推送声音，方便特别提醒某类消息，可以把声音填写到 TIMCustomElem 中的 sound 字段，推送时 Server 会把该字段填入 Ext。参考 [单条消息声音](#) 文档。

Android 离线推送

Android 在 1.8.0 以后版本支持了服务和进程分离，如果 App 进程被杀，服务仍然存活，可以收到离线推送功能。具体配置以及设置过程，可参考 [Android 离线推送](#) 文档。

后台发送消息

后台发送消息时，可以控制 APNs 推送的展示形式，详细可参考：[Apple Push Notification Service \(APNS \) 相关说明](#) 文档。