即时通信
服务端集成
server v1

2024-08-30

服务端集成概述

更新时间:2024-08-30

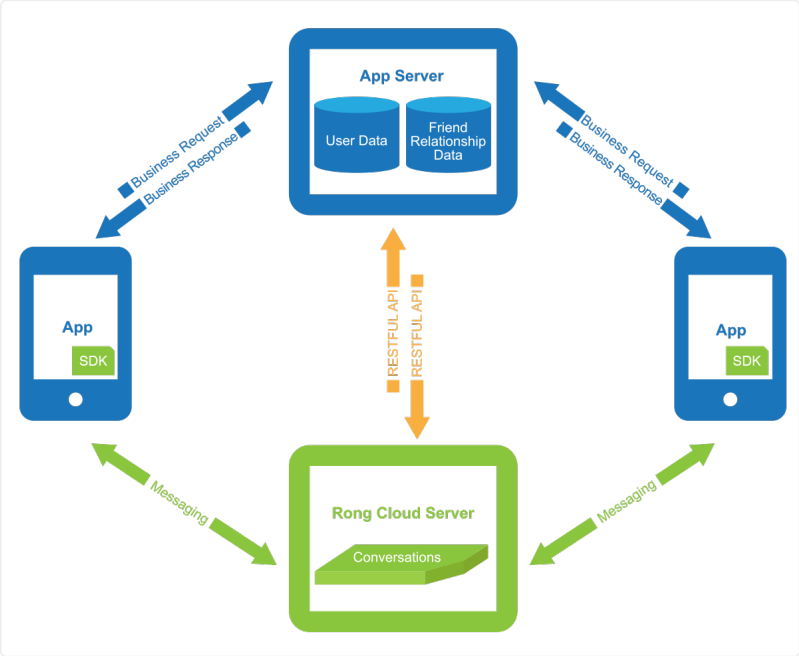
欢迎访问即时通讯服务端文档。您可以在您的应用服务端（App Server）集成即时通讯服务，构建完整和丰富的业务体验。

即时通讯服务端 API 支持从应用自身的服务端（App Server）向即时通讯服务端发送请求，例如将 App 用户接入即时通讯服务（注册用户）、发送消息。部分服务管理类接口、以及高级扩展特性仅在 IM Server API 中提供，例如 系统通知、在线用户广播、创建群组、用户封禁等。关于 API 的完整功能、调试工具、注意事项的详细描述，请参见 API 接口列表。

即时通讯提供多个服务端回调，支持将即时通讯服务中的部分数据和状态同步给指定的应用后端服务，例如用户在线状态变更、聊天室房间状态变更等。支持通过回调响应正文的参数控制消息是否发送给接收者。所有服务端回调均需要在控制台分别配置回调地址并启用。详见服务端回调。

交互关系图

> 提示
>
> 所有 Server API 接口必须通过 App Server 进行调用。切勿通过客户端直接调用 Server API 接口，避免 App Secret 泄漏等问题，



本文将以 注册用户 为例，帮助您快速了解服务端 API 接口的使用方式。

> 提示
>
> IM Server API 的主要功能之一是注册用户 。您必须从 App 后端必须调用该接口，使用 App 的用户 ID 换取 Token，App 用户才能接入即时通讯服务。

前提条件

在开始调用服务端 API 接口之前，请确保：

- 已在控制台创建应用，并获取了开发环境或者生产环境下有效的 App Key / App Secret。
- 已准备好自己的客户端，搭建相关的业务场景。例如，调用服务端 API 接口注册用户之后，可以使用获取到的 Token 在自己的客户端上建立 IM 连接。
- 推荐您提前了解 IM Server API 的默认行为与配置，以使整体接入体验更为顺畅。

调用方法

在控制台创建应用之后，即可创建 IM Server API 请求。建议使用 HTTPS 协议发出请求，以便对流量进行加密。

每个使用即时通讯业务的用户均需要持有即时通讯服务的身份验证令牌（Token）。以获取 Token 为例，请求如下：

```
POST /user/getToken.json HTTP/1.1
Host: api.rong-api.com
App-Key: your-own-app-key
Nonce: 14314
Timestamp: 1408710653000
Signature: 30be0bbca9c9b2e27578701e9fda2358a814c88f
Content-Type: application/x-www-form-urlencoded
Content-Length: 78

userId=jlk456j5&name=Ironman&portraitUri=http%3A%2F%2Fabc.com%2Fmyportrait.jpg
```

在构建 API 请求时，请关注以下内容：

- **HTTP 方法**：IM Server API 当前版本全部接口为 POST 请求方式
- **Host**：IM Server API 域名。即时通讯服务为国内应用提供两个域名，建议您自行实现域名切换机制。详见 Server API 域名。
- **App-Key**：应用的开发环境或生产环境的 App Key，可在控制台创建、管理应用，并获取 App Key。
- **Nonce**：随机数，不超过 18 个字符。
- **Timestamp**：时间戳，从 1970 年 1 月 1 日 0 点 0 分 0 秒开始到现在的毫秒数。
- **Signature**：需要通过计算得到。登录控制台，获取与您应用/环境的 App Key（即上方的 App-Key 字段）对应的 **App Secret**，按顺序拼接 `App Secret` + `Nonce` + `Timestamp` 为一个字符串，进行 SHA1 哈希计算。详见 API 请求签名。
- **Content-Type**：一般均为 `application/x-www-form-urlencoded`，有部分接口例外。详见数据格式。

发出请求后，将返回 `application/json` 格式的响应数据，其中包括：

- **状态码**：指示成功或失败的 HTTP 状态码。200 表示请求成功。有关 HTTP 错误代码的详细信息，请参阅状态码。
- **结果数据**：您请求的数据或操作的结果。对于某些操作，响应消息可以为空。

数据格式

IM Server API 接口一般使用 `application/x-www-form-urlencoded` 格式发送数据。

请注意，以下接口使用 application/json 格式发送数据：

- **发送单聊模板消息**：`/message/private/publish_template.json`
- **发送单个用户系统通知模板消息**：`/message/system/publish_template.json`
- **发送超级群消息**：`/message/ultragroup/publish.json`
- **为单个用户设置标签**：`/user/tag/set.json`
- **为多个用户设置标签**：`/user/tag/batch/set.json`
- **不落地通知**：`/push.json`
- **单个用户不落地通知**：`/push/user.json`
- **推送 Plus**：`/push/custom.json`

频率限制

更新时间:2024-08-30

即时通讯服务限制每个帐户（App Key）的最大 API 请求频率，确保所有客户可享受一致的、高质量的服务。

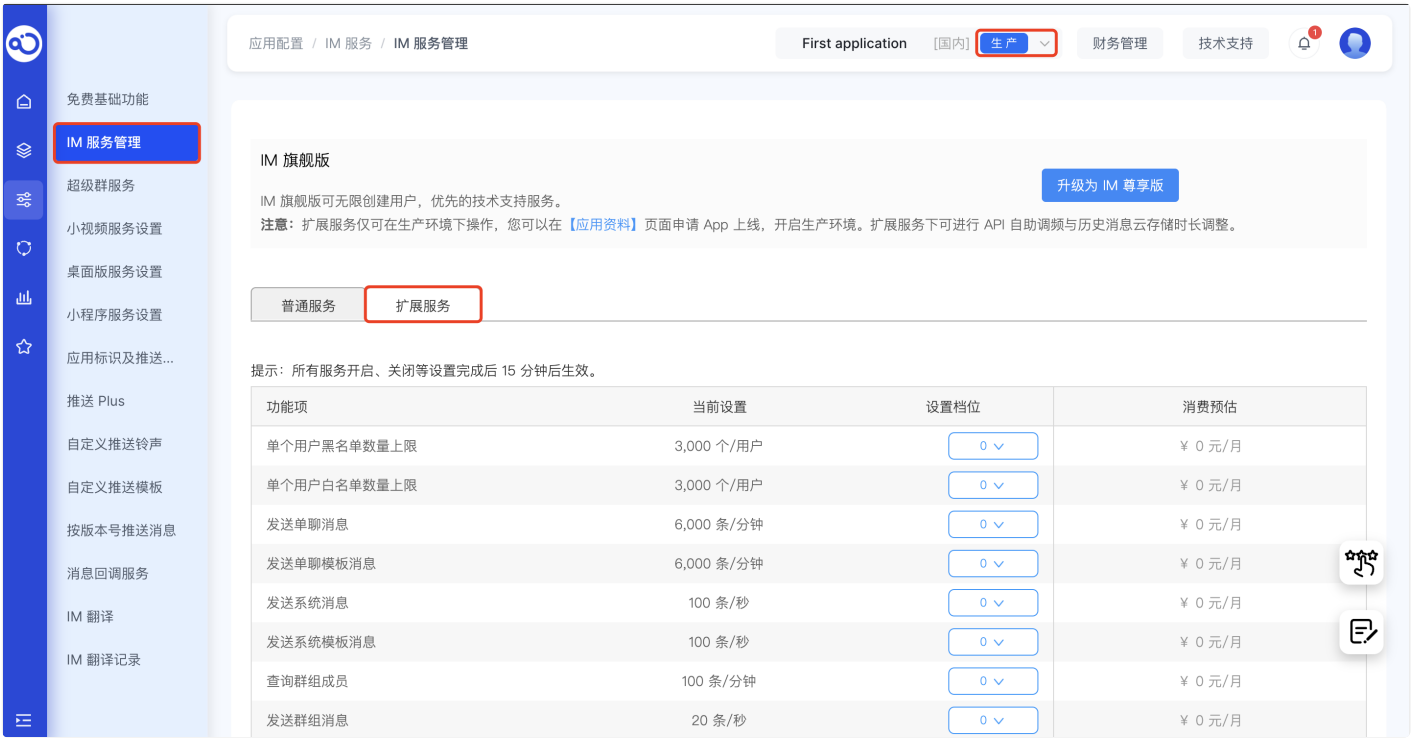
- API 接口列表中统一列出了各个接口的具体限频，大部分接口的频率限制一般为 **100 次/秒**，部分 API 接口不适用该限制。
- API 接口列表中标注「可调频」的接口支持自助调整频率。

如果使用 API 时超出这些限制，Server API 将返回 HTTP 429 Too Many Requests 错误。

自助调整频率限制

IM 旗舰版或 IM 尊享版客户自助调整部分 API 接口在生产环境中的调用频率上限。上调限频为增值服务，具体费用详见计费细则 。

访问控制台 IM 服务管理 页面，切换到应用的生产环境，再切换到扩展服务标签，可以查看所有可自助调整频率的 API 接口。



注意事项：

- 调整扩展服务页面发送全量用户通知频率，会同时调整以下接口的限频：
  - `/message/broadcast.json`：发送全量用户落地通知
  - `/push.json`：发送全量用户不落地通知、发送标签用户通知、发送应用包名通知
- 如需调整推送 Plus 专用接口 `/push/custom.json` 的调用频率上限，请联系销售专员，电话 13161856839。

已知问题

1. 设置单群聊消息扩展与删除单群聊消息扩展会额外消耗发送单聊普通消息（默认 6000 条每分钟）或发送群聊消息（默认 20 条每秒）的限频配额。由于发送群聊消息 API 默认限频为 20 条/秒，因此可能容易受到该问题影响。建议您在发送群聊消息后，不要立即高频设置或删除消息扩展，以免触发该问题。

API 接口列表

更新时间:2024-08-30

本文档列出了即时通讯（IM）服务端提供的 API 接口、调试工具、及注意事项。

> 提示
>
> - IM Server API 当前为 v1 版本，全部接口均使用 **POST** 请求方式。
> - 如果您首次接入 IM Server API，强烈推荐先了解 API 调用方法。

API 默认行为与配置

集成 IM Server API 需要注意以下默认行为：

1. 应用服务端可调用 IM Server API 直接发送消息。如果以用户身份向群组、聊天室发送消息，不要求已加入群组或聊天室。
2. 应用服务端调用 IM Server API 的行为不会在北极星的连接信息中生成记录。但通过 Server API 发送的消息均可通过北极星的消息流转查询。
3. 如果应用配置了敏感词过滤、消息回调服务、第三方审核，请注意调用 Server API 发送的消息默认不进行过滤。如有需要，您需要在控制台免费基础功能页面启用 **Server API 发送消息过滤敏感词**。
4. 如果应用启用了全量消息路由，请注意调用 Server API 发送的消息默认不进行路由。如有需要，您需要在控制台免费基础功能页面启用 **Server API 发送消息实时路由**。
5. Server API 的部分接口为即时通讯的高级、扩展特性或付费增值提供的接口。需要为开通服务后才能使用。您可以在控制台的免费基础功能页面与 IM 服务管理页面找到大部分服务配置开关，也可以在阅读相关 API 接口文档时找到具体细节。

API 调试工具

> 提示
>
> 本文档已列出了 API 接口在控制台「北极星」调试地址（需登录开发者账号）。

- 控制台「北极星」开发者工具箱的 IM Server API 调试 页面提供了大部分 API 接口的调试功能。请注意区分开发环境与生产环境。

用户管理

> 提示
>
> IM Server API 的主要功能之一是注册用户 。您需要使用 App 的用户 ID 换取 Token，App 用户才能接入即时通讯服务。

功能/文档页面	API URL	频率限制	北极星 API 调试地址
注册用户	/user/getToken.json	200 次/每秒，可调频	API 调试
作废 Token	/user/token/expire.json	100 次/每秒	API 调试
获取用户信息	/user/info.json	100 次/每秒	API 调试
修改用户信息	/user/refresh.json	100 次/每秒	API 调试
注销用户	/user/deactivate.json	100 用户/每秒	暂不支持
查询已注销用户	/user/deactivate/query.json	100 次/每秒	暂不支持
重新激活用户 ID	/user/reactivate.json	100 用户/每秒	暂不支持
封禁用户	/user/block.json	100 次/每秒，可调频。	API 调试
解除用户封禁	/user/unblock.json	100 次/每秒，可调频。	API 调试
获取封禁用户列表	/user/block/query.json	100 次/每秒，可调频。	API 调试
用户状态	/user/checkOnline.json	100 次/每秒，可调频。	API 调试
设置用户单聊禁言	/user/chat/fb/set.json	100 次/每秒	API 调试
查询单聊禁言用户列表	/user/chat/fb/querylist.json	100 次/每秒	API 调试
设置用户标签	/user/tag/set.json	100 次/每秒	API 调试
批量设置用户标签	/user/tag/batch/set.json	10 次/每秒	暂不支持
获取用户标签	/user/tags/get.json	100 次/每秒	API 调试

用户黑/白名单服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
添加用户到黑名单	/user/blacklist/add.json	100 次/每秒，可调频。	API 调试
从黑名单中移除用户	/user/blacklist/remove.json	100 次/每秒，可调频。	API 调试
获取某用户的黑名单列表	/user/blacklist/query.json	100 次/每秒，可调频。	API 调试
为用户开启白名单	/user/whitesetting/set.json	100 次/每秒	API 调试
查询用户白名单服务状态	/user/whitesetting/query.json	100 次/每秒	API 调试
添加用户到白名单	/user/whitelist/add.json	100 次/每秒，可调频。	API 调试
从用户白名单中移除用户	/user/whitelist/remove.json	100 次/每秒，可调频。	API 调试
获取用户的白名单列表	/user/whitelist/query.json	100 次/每秒，可调频。	API 调试

消息管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
发送单聊普通消息	/message/private/publish.json	6000 条消息/每分钟，按收件人数量计算条数，可调频。	API 调试
发送单聊模板消息	/message/private/publish_template.json	6000 条消息/每分钟，按收件人数量计算条数，可调频。	API 调试

功能/文档页面	API URL	频率限制	北极星 API 调试地址
发送单聊状态消息	/statusmessage/private/publish.json	6000 条消息/每分钟，按收件人数量计算条数。	API 调试
发送群聊消息	/message/group/publish.json	20 条/每秒，按目标群组数量计算条数，可调频。	API 调试
发送群聊状态消息	/statusmessage/group/publish.json	20 条/每秒，按目标群组数量计算条数	API 调试
发送超级群消息	/message/ultragroup/publish.json	100 条/每秒，按目标群组数量计算条数；单个频道限 20 条/每秒，可调频	API 调试
发送聊天室消息	/message/chatroom/publish.json	100 条/每秒，按目标聊天室数量计算条数，可调频。	API 调试
发送全体聊天室广播消息	/message/chatroom/broadcast.json	1 次/每秒，可调频。	API 调试
设置单群聊消息扩展	/message/expansion/set.json	100 次/每秒，其中群聊消息扩展最多 20 次。	API 调试
删除单群聊消息扩展	/message/expansion/delete.json	100 次/每秒，其中群聊消息扩展最多 20 次。	API 调试
获取单群聊消息扩展	/message/expansion/query.json	100 次/每秒	API 调试
撤回消息	/message/recall.json	100 次/每秒	API 调试
获取历史消息日志	/message/history.json	100 次/每秒	API 调试
删除历史消息日志	/message/history/delete.json	100 次/每秒	API 调试
清除消息	/conversation/message/history/clean.json	100 次/每秒	API 调试

会话管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
会话置顶	/conversation/top/set.json	100 次/每秒	暂不支持

系统通知

> 提示
>
> 下表中频率限制一栏标注「共享」的项目均使用 /push.json 接口，共享该接口频率限额，即 2 次/每小时，3 次/每自然日，可调频。

功能/文档页面	API URL	频率限制	北极星 API 调试地址
发送系统通知普通消息	/message/system/publish.json	100 条/每秒，按收件人数量计算条数，可调频。	API 调试
发送系统通知模板消息	/message/system/publish_template.json	100 条/每秒，按收件人数量计算条数，可调频。	API 调试
撤回单条系统通知	/message/recall.json	100 次/每秒	API 调试

功能/文档页面	API URL	频率限制	北极星 API 调试地址
发送全量用户落地通知	/message/broadcast.json	2 次/每小时，3 次/每自然日，可调频。	API 调试
发送在线用户广播	/message/online/broadcast.json	60 次/每分钟	API 调试
发送全量用户不落地通知	/push.json	2 次/每小时，3 次/每自然日（共享），可调频。	API 调试
发送标签用户通知	/push.json	2 次/每小时，3 次/每自然日（共享），可调频。	API 调试
发送应用包名通知	/push.json	2 次/每小时，3 次/每自然日（共享），可调频。	API 调试
撤回全量用户落地通知	/message/broadcast/recall.json	2 次/每小时，3 次/每自然日	暂不支持
发送指定用户不落地通知	/push/user.json	100 条/每秒，按收件人数量计算条数，可调频。	API 调试

群组管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
创建群组	/group/create.json	100 次/每秒	API 调试
解散群组	/group/dismiss.json	100 次/每秒	API 调试
加入群组	/group/join.json	100 次/每秒	API 调试
退出群组	/group/quit.json	100 次/每秒	API 调试
查询群组成员	/group/user/query.json	100 次/每秒，可调频。	API 调试
同步用户所在群组	/group/sync.json	100 次/每秒	暂不支持
查询用户所在群组	/user/group/query.json	100 次/每秒。	API 调试
刷新群组信息	/group/refresh.json	100 次/每秒	API 调试

群组禁言服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
禁言指定群成员	/group/user/gag/add.json	100 次/每秒	API 调试
取消指定群成员禁言	/group/user/gag/rollback.json	100 次/每秒	API 调试
查询群成员禁言列表	/group/user/gag/list.json	100 次/每秒	API 调试
设置群组全体禁言	/group/ban/add.json	100 次/每秒	API 调试
取消群组全体禁言	/group/ban/rollback.json	100 次/每秒	API 调试
查询群组全体禁言	/group/ban/query.json	100 次/每秒	API 调试
加入群组全体禁言白名单	/group/user/ban/whitelist/add.json	100 次/每秒	API 调试
移出群组全体禁言白名单	/group/user/ban/whitelist/rollback.json	100 次/每秒	API 调试
查询群组全体禁言白名单	/group/user/ban/whitelist/query.json	100 次/每秒	API 调试

超级群管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
创建超级群	/ultragroup/create.json	100 次/每秒	API 调试
加入超级群	/ultragroup/join.json	100 次/每秒，可调频	API 调试
退出超级群	/ultragroup/quit.json	100 次/每秒	API 调试
解散超级群	/ultragroup/dis.json	100 次/每秒	API 调试
刷新超级群信息	/ultragroup/refresh.json	100 次/每秒	API 调试
创建频道	/ultragroup/channel/create.json	100 次/每秒	API 调试
删除频道	/ultragroup/channel/del.json	100 次/每秒	API 调试
查询频道列表	/ultragroup/channel/get.json	100 次/每秒	API 调试
获取指定超级群消息内容	/ultragroup/msg/get.json	5 次/每秒	暂不支持
修改超级群消息	/ultragroup/msg/modify.json	100 次/每分钟	暂不支持
搜索超级群消息	/ultragroup/hismsg/query.json	100 次/每分钟	暂不支持
搜索超级群消息上下文	/ultragroup/hismsg/msgid/query.json	100 次/每分钟	暂不支持
设置超级群消息扩展	/ultragroup/message/expansion/set.json	100 次/每秒，可调频	暂不支持
删除超级群消息扩展	/ultragroup/message/expansion/delete.json	100 次/每秒	暂不支持
获取超级群消息扩展	/ultragroup/message/expansion/query.json	100 次/每秒，可调频	暂不支持
查询用户是否为群成员	/ultragroup/member/exist.json	100 次/每秒	暂不支持
设置群/频道默认免打扰	/ultragroup/notdisturb/set.json	100 次/每秒，可调频	暂不支持
查询默认免打扰配置	/ultragroup/notdisturb/get.json	100 次/每秒	暂不支持

超级群私有频道

功能/文档页面	API URL	频率限制	北极星 API 调试地址
变更频道类型	/ultragroup/channel/type/change.json	100 次/每秒	暂不支持
添加私有频道成员	/ultragroup/channel/private/users/add.json	100 次/每秒	暂不支持
删除私有频道成员	/ultragroup/channel/private/users/del.json	100 次/每秒	暂不支持
查询私有频道成员列表	/ultragroup/channel/private/users/get.json	100 次/每秒	暂不支持
查询用户所属的私有频道	/ultragroup/user/channel/query.json	100 次/每秒	暂不支持

超级群用户组

功能/文档页面	API URL	频率限制	北极星 API 调试地址
创建用户组	/ultragroup/usergroup/add.json	100 次/每秒	暂不支持

功能/文档页面	API URL	频率限制	北极星 API 调试地址
删除用户组	/ultragroup/usergroup/del.json	100 次/每秒	暂不支持
查询用户组列表	/ultragroup/usergroup/query.json	100 次/每秒	暂不支持
添加用户	/ultragroup/usergroup/user/add.json	100 次/每秒	暂不支持
移出用户	/ultragroup/usergroup/user/del.json	100 次/每秒	暂不支持
查询用户所属用户组	/ultragroup/user/usergroup/query.json	100 次/每秒	暂不支持
绑定频道与用户组	/ultragroup/channel/usergroup/bind.json	100 次/每秒	暂不支持
解绑频道与用户组	/ultragroup/channel/usergroup/unbind.json	100 次/每秒	暂不支持
查询频道绑定的用户组	/ultragroup/channel/usergroup/query.json	100 次/每秒	暂不支持
查询用户组绑定的频道	/ultragroup/usergroup/channel/query.json	100 次/每秒	暂不支持

超级群禁言服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
禁言指定超级群成员	/ultragroup/userbanned/add.json	100 次/每秒	API 调试
取消指定超级群成员禁言	/ultragroup/userbanned/del.json	100 次/每秒	API 调试
查询超级群成员禁言列表	/ultragroup/userbanned/get.json	100 次/每秒	API 调试
设置超级群全体禁言	/ultragroup/globalbanned/set.json	100 次/每秒	API 调试
查询超级群全体禁言	/ultragroup/globalbanned/get.json	100 次/每秒	API 调试
加入超级群全体禁言白名单	/ultragroup/banned/whitelist/add.json	100 次/每秒	API 调试
移出超级群全体禁言白名单	/ultragroup/banned/whitelist/del.json	100 次/每秒	API 调试
查询超级群全体禁言白名单	/ultragroup/banned/whitelist/get.json	100 次/每秒	API 调试

聊天室管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
创建聊天室	/chatroom/create_new.json	100 次/每秒	暂不支持
设置聊天室销毁类型	/chatroom/destroy/set.json	100 次/每秒	暂不支持
销毁聊天室	/chatroom/destroy.json	100 次/每秒	API 调试
查询聊天室信息	/chatroom/get.json	100 次/每秒	暂不支持
绑定音视频房间	/chatroom/correlation/rtc.json	100 次/每秒	API 调试
创建聊天室（已废弃）	/chatroom/create.json	100 次/每秒	API 调试
查询聊天室信息（已废弃）	/chatroom/query.json	100 次/每秒	API 调试

聊天室保活服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
保活聊天室	/chatroom/keepalive/add.json	100 次/每秒	API 调试

功能/文档页面	API URL	频率限制	北极星 API 调试地址
取消保活聊天室	/chatroom/keepalive/remove.json	100 次/每秒	API 调试
查询保活聊天室	/chatroom/keepalive/query.json	100 次/每秒	API 调试

聊天室用户与禁言管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
获取聊天室成员	/chatroom/user/query.json	100 次/每秒	API 调试
查询是否在聊天室中	/chatroom/user/exist.json	100 次/每秒	API 调试
批量查询是否在聊天室中	/chatroom/users/exist.json	100 次/每秒	API 调试
禁言指定聊天室用户	/chatroom/user/gag/add.json	100 次/每秒	API 调试
取消禁言指定聊天室用户	/chatroom/user/gag/rollback.json	100 次/每秒	API 调试
查询聊天室禁言用户列表	/chatroom/user/gag/list.json	100 次/每秒	API 调试
设置聊天室全体禁言	/chatroom/ban/add.json	100 次/每秒	API 调试
取消聊天室全体禁言	/chatroom/ban/rollback.json	100 次/每秒	API 调试
查询聊天室全体禁言列表	/chatroom/ban/query.json	100 次/每秒	API 调试
查询聊天室全体禁言状态	/chatroom/ban/check.json	100 次/每秒	API 调试
加入聊天室全体禁言白名单	/chatroom/user/ban/whitelist/add.json	100 次/每秒	API 调试
移出聊天室全体禁言白名单	/chatroom/user/ban/whitelist/rollback.json	100 次/每秒	API 调试
查询聊天室全体禁言白名单	/chatroom/user/ban/whitelist/query.json	100 次/每秒	API 调试
全局禁言用户	/chatroom/user/ban/add.json	100 次/每秒	API 调试
取消全局禁言用户	/chatroom/user/ban/remove.json	100 次/每秒	API 调试
查询全局禁言用户列表	/chatroom/user/ban/query.json	100 次/每秒	API 调试
封禁聊天室用户	/chatroom/user/block/add.json	100 次/每秒	API 调试
解除封禁聊天室用户	/chatroom/user/block/rollback.json	100 次/每秒	API 调试
查询聊天室封禁用户	/chatroom/user/block/list.json	100 次/每秒	API 调试

聊天室属性（KV）

提示

下表中频率限制一栏标注「共享」的没有独立的接口频率限额。设置单个聊天室属性与批量设置聊天室属性接口共享 100 个属性每秒的限额。删除单个聊天室属性与批量删除聊天室属性接口共享 100 个属性每秒的限额。

功能/文档页面	API URL	频率限制	北极星 API 调试地址
设置聊天室属性（KV）	/chatroom/entry/set.json	100 个属性/每秒（共享），可调频	API 调试
批量设置聊天室属性（KV）	/chatroom/entry/batch/set.json	100 个属性/每秒（共享），可调频	暂不支持
删除聊天室属性（KV）	/chatroom/entry/remove.json	100 个属性/每秒（共享），可调频	API 调试
批量删除聊天室属性（KV）	/chatroom/entry/batch/remove.json	100 个属性/每秒（共享），可调频	暂不支持
查询聊天室属性（KV）	/chatroom/entry/query.json	100 次/每秒，可调频	API 调试

聊天室消息优先级服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
添加低级别消息类型	/chatroom/message/priority/add.json	100 次/每秒	API 调试
移除低级别消息类型	/chatroom/message/priority/remove.json	100 次/每秒	API 调试
查询低级别消息类型	/chatroom/message/priority/query.json	100 次/每秒	API 调试

聊天室白名单服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
加入聊天室用户白名单	/chatroom/user/whitelist/add.json	100 次/每秒	API 调试
移出聊天室用户白名单	/chatroom/user/whitelist/remove.json	100 次/每秒	API 调试
查询聊天室用户白名单	/chatroom/user/whitelist/query.json	100 次/每秒	API 调试
加入聊天室消息白名单	/chatroom/whitelist/add.json	100 次/每秒	API 调试
移出聊天室消息白名单	/chatroom/whitelist/delete.json	100 次/每秒	API 调试
查询聊天室消息白名单	/chatroom/whitelist/query.json	100 次/每秒	API 调试

内容审核

功能/文档页面	API URL	频率限制	北极星 API 调试地址
添加消息敏感词	/sensitiveword/add.json	100 次/每秒	API 调试
移除消息敏感词	/sensitiveword/delete.json	100 次/每秒	API 调试
批量移除消息敏感词	/sensitiveword/batch/delete.json	100 次/每秒	API 调试
查询消息敏感词	/sensitiveword/list.json	100 次/每秒	API 调试

推送与通知管理

功能/文档页面	API URL	频率限制	北极星 API 调试地址
设置指定会话免打扰	/conversation/notification/set.json	100 次/每秒，可调频	API 调试
查询指定会话免打扰	/conversation/notification/get.json	100 次/每秒	API 调试
设置指定会话类型免打扰	/conversation/type/notification/set.json	100 次/每秒	暂不支持
查询指定会话类型免打扰	/conversation/type/notification/get.json	100 次/每秒	暂不支持
设置用户免打扰时段	/user/blockPushPeriod/set.json	100 次/每秒，可调频	暂不支持
删除用户免打扰时段	/user/blockPushPeriod/delete.json	100 次/每秒，可调频	暂不支持
查询用户免打扰时段	/user/blockPushPeriod/get.json	100 次/每秒	暂不支持
推送 Plus	/push/custom.json	20 次/每小时，100 次/每自然日，可调频。	API 调试
推送聚合统计	/stat/getDayPushData	1 次/每秒	API 调试
单次推送统计	/stat/getPushIdData	1 次/每秒	API 调试
设置用户级推送备注名	/user/remarks/set.json	100 次/每秒	暂不支持
删除用户级推送备注名	/user/remarks/del.json	100 次/每秒	暂不支持
查询用户级推送备注名	/user/remarks/get.json	100 次/每秒	暂不支持
设置群成员推送备注名	/group/remarks/set.json	100 次/每秒	暂不支持
删除群成员推送备注名	/group/remarks/del.json	100 次/每秒	暂不支持
查询群成员推送备注名	/group/remarks/get.json	100 次/每秒	暂不支持

翻译服务

功能/文档页面	API URL	频率限制	北极星 API 调试地址
获取 JWT Token	/jwt/getToken.json	100 次/每秒	暂不支持

API 域名

更新时间:2024-08-30

您在控制台创建应用时，可根据业务所在的环境选择 **国内数据中心** 或 **海外数据中心**（参见数据中心 ）。不同数据中心使用独立的服务端 API 地址。应用服务器在调用 Server API 时必须使用与数据中心对应的 Server API 地址，否则 API 请求将无法正确返回结果。

> **重要**
>
> - **为避免 App Secret 泄漏等问题，所有 Server API 接口必须通过 App Server 进行调用。切勿通过客户端直接调用 Server API 接口。**
> - **在调用 Server API 时，建议不要使用 KeepAlive 方式。如有特殊情况需要使用 KeepAlive，建议每条长连接空闲超时小于 55 秒，并且复用次数小于 80 次。建议在一次连接空闲 55 秒或复用 80 次时切换新连接。长期使用同一条连接会导致 API 负载均衡失效，影响故障转移策略。**

Server API 域名

国内数据中心 API 地址

即时通讯服务为国内数据中心的应用提供了两个独立的服务端 API 地址：

- `api.rong-api.com`
- `api-b.rong-api.com`

> **提示**
>
> **建议您实现默认域名与备用域名自动切换的逻辑，尽量避免因单一 CDN 服务商问题，导致访问即时通讯服务端 API 受阻，进而影响您业务。**

以下 Server API 域名已过时，目前仍然可以使用，但已不推荐使用。

- `api.cn.ronghub.com`
- `api-cn.ronghub.com`
- `api2-cn.ronghub.com`
- `ai-sg.ronghub.com`（翻译服务）

海外数据中心 API 地址

海外数据中心的应用，请使用对应的服务端 API 地址：

- 新加坡：`api.sg-light-api.com`（主）、`api-b.sg-light-api.com`（备）
- 新加坡 B：`api.sg-b-light-api.com`（主）、`api-b.sg-b-light-api.com`（备）
- 北美：`api.us-light-api.com`（主）、`api-b.us-light-api.com`（备）
- 沙特：`api.sau-light-api.com`（主）、`api-b.sau-light-api.com`（备）

> **重要**
>
> 使用海外数据中心的应用还需要配置客户端 SDK 的数据中心地址。详见知识库文档海外数据中心使用指南。

域名切换最佳实践

> **提示**
>
> 域名切换仅针对使用国内数据中心的应用。

使用单一服务端 API 地址时，可能会因为 CDN 服务商问题，导致访问 Server API 延时，对自身业务造成影响。针对使用国内数据中心的应用，我们提供了两个域名，建议开发者实现默认域名与备用域名的切换逻辑。

1. 开发者调用即时通讯服务端 API 时，需要保存当前使用的域名。
2. 调用过程中，如访问出现超时或不可用时，立即切换至另一备用域名进行访问，同时更新当前使用域名，如此循环，避免对业务造成影响。

API 请求签名

更新时间:2024-08-30

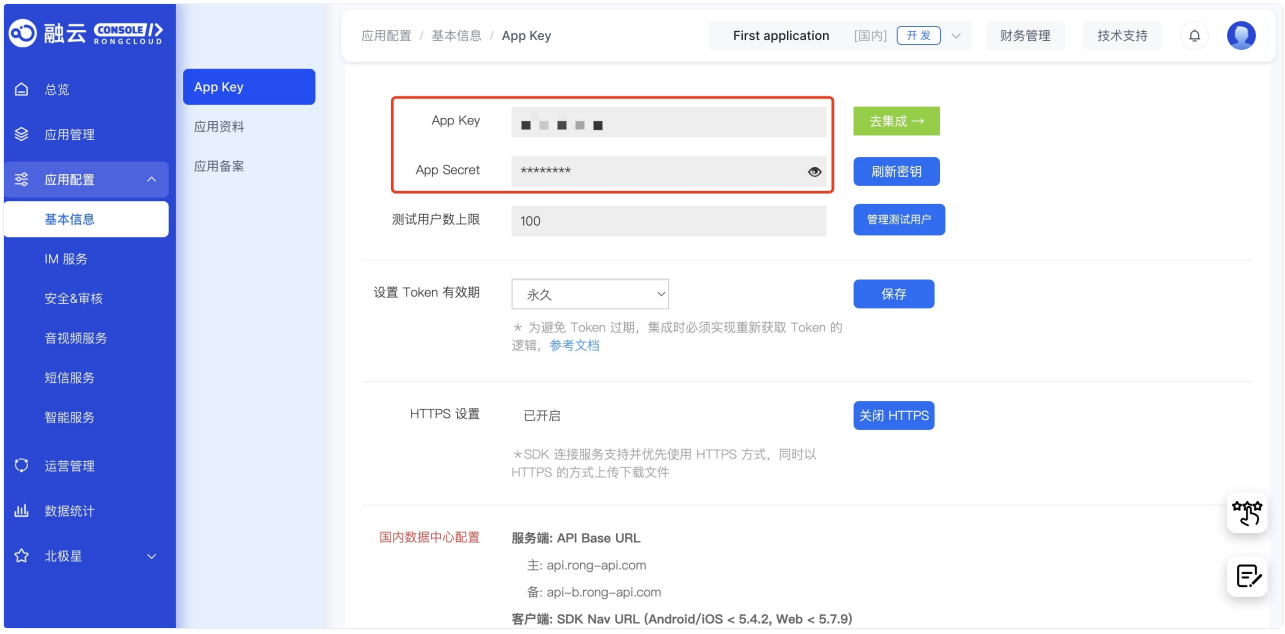
如需访问即时通讯服务端 API（Server API），您需要对 API 请求签名，让即时通讯服务端能够验证应用程序的身份。

基本过程如下：

1. **App 后端需要在认证服务中预先设置 App Key 和 App Secret。**
2. **在调用 Server API 时，需要满足 HTTP 标头 要求。** HTTP 标头中的数据签名 `signature` **需要使用 App Secret、随机数、时间戳进行计算。**
3. **即时通讯服务端收到请求后，会使用对应的 App Secret 进行同样的计算，并要求结果完全一致。**
4. **注意 App Secret 不可泄漏。请不要在网络中传输 App Secret，不要在不受信任的位置存放（浏览器等）。**

获取 App Key / App Secret

获取应用的 App Key / App Secret 是使用即时通讯服务端 API 的必要条件。您可以前往控制台 App Key 页面进行查询。



您需要记录上图所示的应用 App Key 和 App Secret，在本教程中使用。在请求即时通讯服务端 API 接口时，每个 HTTP 请求中需要携带应用的 App Key 与数据签名。App Secret 用于计算数据签名，请注意不要泄露。

HTTP 标头

App Server 调用 API 接口时，每个 HTTP 请求中均需要携带以下的 HTTP 标头字段（HTTP Request Header），用于向即时通讯服务端提供身份认证信息：

默认名称	带 RC-前缀	类型	说明
App-Key	RC-App-Key	String	控制台分配的 App Key。

默认名称	带 RC-前缀	类型	说明
Nonce	RC-Nonce	String	随机数，不超过 18 个字符。
Timestamp	RC-Timestamp	String	时间戳，从 1970 年 1 月 1 日 0 点 0 分 0 秒开始到现在的毫秒数。
Signature	RC-Signature	String	数据签名。您需要参考下文的签名计算方法生成该字段的值。

（可选）在标头中添加 X-Request-ID 字段

我们建议在向融云发送的 Server API 标头中携带 X-Request-ID 字段，这有助于融云更高效地排查问题。融云的 Server SDK 已自动带上 X-Request-ID 字段，因此您无需额外生成。

Server SDK 链接：

- server-sdk-java (GitHub)
- server-sdk-php (GitHub)
- server-sdk-go (GitHub)

如果您需要自行生成 X-Request-ID（长度最大 36 位），可以参考下方示例：

Java 示例：

```
import java.util.UUID;

HttpURLConnection conn = getHttpURLConnection(config, uri);
conn.setRequestProperty("X-Request-ID", UUID.randomUUID().toString().replaceAll("\\-", ""));
```

PHP 示例：

```
private function create_guid()
{
$charid = strtoupper(md5(uniqid(mt_rand(), true)));
$uuid = substr($charid, 0, 8)
. substr($charid, 8, 4)
. substr($charid, 12, 4)
. substr($charid, 16, 4)
. substr($charid, 20, 12);
return strtolower($uuid);
}
$header = [
'RC-App-Key:' . $appKey,
'RC-Nonce:' . $nonce,
'RC-Timestamp:' . $timeStamp,
'RC-Signature:' . $sign,
'X-Request-ID:' . $this->create_guid()
];
```

Go 示例：

```
import "github.com/google/uuid"
// getSignature 本地生成签名
// Signature（数据签名)计算方法：将系统分配的 App Secret、Nonce（随机数）、
// Timestamp（时间戳)三个字符串按先后顺序拼接成一个字符串并进行 SHA1 哈希计算。如果调用的数据签名验证失败，接口调用会返回 HTTP 状态码 401。
func (rc RongCloud) getSignature() (nonce, timestamp, signature string) {
nonceInt := rand.Int()
nonce = strconv.Itoa(nonceInt)
timeInt64 := time.Now().Unix()
timestamp = strconv.FormatInt(timeInt64, 10)
h := sha1.New()
_, _ = io.WriteString(h, rc.appSecret+nonce+timestamp)
signature = fmt.Sprintf("%x", h.Sum(nil))
return
}

// fillHeader 在 Http Header 增加API签名
func (rc RongCloud) fillHeader(req *httplib.BeegoHTTPRequest) string {
requestId := uuid.New().String()
nonce, timestamp, signature := rc.getSignature()
req.Header("RC-App-Key", rc.appKey)
req.Header("RC-Timestamp", timestamp)
req.Header("RC-Nonce", nonce)
req.Header("RC-Signature", signature)
req.Header("Content-Type", "application/json")
req.Header("User-Agent", USERAGENT)
req.Header("RC-Request-Id", requestId)
return requestId
}
```

> 提示
>
> 请在每次请求中携带新生成的 X-Request-ID，并做好记录。

签名计算方法

API 请求中需携带该数据签名（Signature）字段，该字段值需要由 App 后端计算生成。步骤如下：

1. 登录控制台，获取与应用 App Key 所对应的 App Secret。
2. 将以下三个字符串按顺序（App Secret + Nonce + Timestamp）拼接成一个字符串，进行 SHA1 哈希计算。
   - App Secret：应用 App Key 所对应的 App Secret。
   - Nonce：随机数
   - Timestamp：时间戳

即时通讯服务端在验证数据签名真实性后，会执行请求的动作。如果调用的数据签名验证失败，接口调用会返回 HTTP 状态码 401。其他状态码请参见状态码表。

以下是计算数据签名的 PHP 代码示例：

```php
// 重置随机数种子。
srand((double)microtime()*1000000);

$appSecret = 'your-own-app-secret'; // 请替换为您从开发者平台获取的 App Secret。
$nonce = rand(); // 获取随机数。
$timestamp = time()*1000; // 获取时间戳（毫秒）。

$signature = sha1($appSecret.$nonce.$timestamp);
```

HTTP 请求示例

以下 HTTP 请求示例展示了 API 请求中的 HTTP 标头字段。

```http
POST /user/getToken.json HTTP/1.1
Host: api.rong-api.com
App-Key: your-own-app-key
Nonce: 14314
Timestamp: 1408710653000
Signature: 30be0bbca9c9b2e27578701e9fda2358a814c88f
Content-Type: application/x-www-form-urlencoded
Content-Length: 78

userId=jlk456j5&name=Ironman&portraitUri=http%3A%2F%2Fabc.com%2Fmyportrait.jpg
```

接收服务端回调 服务端回调概述

更新时间:2024-08-30

即时通讯业务提供以下服务端回调，支持将业务的数据、状态通知同步到您设置的回调地址。所有服务端回调均需要在控制台启用。

您可以访问以下文档，了解如何在控制台配置并启用回调，以及如何解析回调数据：

- 用户在线状态：实时将用户状态同步到设置的 App 服务器地址。
- 用户注销与激活状态回调: 在注销用户、重新激活用户 ID 操作完成时将处理结果同步到设置的 App 服务器地址。
- 全量消息路由：实时将消息同步到设置的 App 服务器地址。
- 消息回调服务：将符合自定义条件的消息实时抄送至设置的 App 服务器地址，并根据 App 服务器响应结果决定是否发送给目标用户。
- 聊天室状态同步：将聊天室发生的状态变化，实时同步到设置的 App 服务器。
- 聊天室属性同步（KV）：将聊天室发生的自定义属性变化，实时同步到设置的 App 服务器。
- 审核结果回调：将 IM & 音视频审核 的审核结果实时同步到设置的 App 服务器地址。

设置 IP 白名单

如果您的网络有 IP 访问限制，请务必将以下 IP 地址加入到白名单中，否则无法正常接收服务端回调。

- 国内数据中心：39.107.252.146（开发）、39.105.128.42（开发）、39.105.147.30（开发）、39.106.14.117（开发）、39.106.61.148（开发）、59.110.153.153（开发）、47.95.215.244（开发）、182.92.215.38（生产）、182.92.84.148（生产）、39.106.150.151（生产）、39.107.75.101（生产）、101.201.34.95（生产）、47.93.57.144（控制台）
- 海外数据中心：52.221.93.74（新加坡）、8.219.168.45（新加坡）、8.219.93.148（新加坡）、8.219.215.35（新加坡）、8.219.43.97（新加坡）、47.245.124.194（新加坡）、52.41.206.152（北美）8.213.17.80（沙特）

> 提示
>
> 以下 IP 地址已不再使用。如果您的 IP 白名单中已添加以下地址，请移除。
> 120.92.12.217、120.92.12.60、120.92.12.253、120.92.12.113、120.92.12.29、120.92.12.214、120.92.12.164、120.92.12.153、120.92.12.204、120.92.12.138、120.92.13.82、120.92.13.83、120.92.13.84、120.92.13.85、120.131.13.147、117.50.18.131、106.75.117.2、47.94.106.191

验证回调签名

即时通讯服务向应用服务器推送数据时会添加 4 个 POST 请求参数。

- 路径参数（签名参数在 POST 请求 URL 中）：用户在线状态、全量消息路由、聊天室房间状态同步、消息回调服务、聊天室属性同步（KV）、用户注销与激活状态回调
- HTTP 标头（签名参数在 HTTP 请求头中）：审核结果回调

参数名	类型	说明
appKey	String	应用和环境的 App Key，可从控制台获取。 例外：消息回调服务 的 POST 请求路径中不含该参数，您可以使用回调请求正文中的 appKey 字段。
nonce	String	随机数，不超过 18 个字符。
timestamp	String	时间戳，从 1970 年 1 月 1 日 0 点 0 分 0 秒开始到现在的毫秒数。
signature (数据签名)	String	数据签名。将系统分配的 App Secret、Nonce (随机数)、Timestamp (时间戳)三个字符串按顺序拼接成一个字符串，进行 SHA1 哈希计算。

计算 Signature 代码示例

PHP 语言的代码示例：

```
$appSecret = 'your-own-app-secret'; // 请替换为您从开发者平台获取的 App Secret。
$nonce = $_GET['nonce']; // 获取随机数。
$timestamp = $_GET['timestamp']; // 获取时间戳。
$signature = $_GET['signature']; // 获取数据签名。
$local_signature = sha1($appSecret.$nonce.$timestamp); // 生成本地签名。
if(strcmp($signature, $local_signature)===0){
// TODO: 此处添加业务逻辑。
echo 'OK';
} else {
echo 'Error';
}
```

用户概述

更新时间:2024-08-30

App 用户需要接入即时通讯服务，才能使用即时通讯服务。对即时通讯服务来说，用户是指持有由即时通讯服务端分发的有效 Token，接入并使用即时通讯服务的 App 用户。

注册用户

应用服务端（App Server）应向即时通讯服务端提供 App 用户的用户 ID（`userId`），以向即时通讯服务端换取唯一用户 Token。对即时通讯服务来说，这个以 `userId` 获取 Token 的步骤即注册用户，且必须通过调用 Server API 来完成。

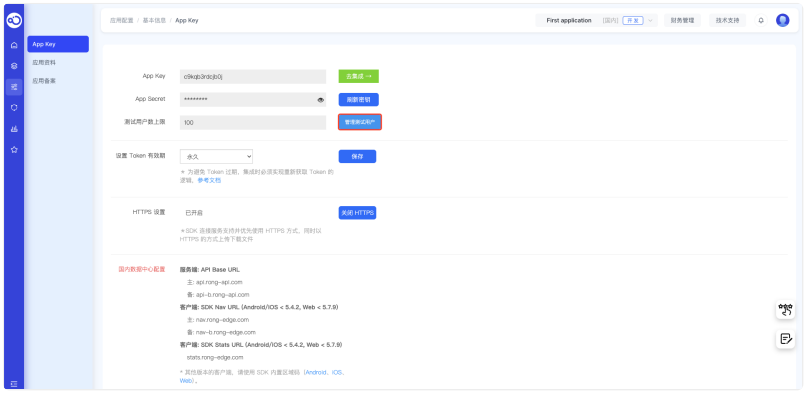
应用客户端必须持有有效 Token，才能成功连接到即时通讯服务端，使用即时通讯服务。当 App 客户端用户向服务器发送登录请求时，服务器会查询数据库以检查连接请求是否匹配。

注册用户数限制

- 开发环境中的注册用户数上限为 100 个。
- 在 生产环境中，升级为 **IM 旗舰版**或 **IM 尊享版**后不限制注册用户数。您可以前往控制台切换当前计费套餐的版本。

删除用户

删除用户是指在应用的开发环境中，通过控制台删除已注册的测试用户，以控制开发环境中的测试用户总数。生产环境不支持该操作。



注销用户

注销用户是指在即时通讯服务中删除用户数据。App 可使用该能力实现自身的用户销户功能，满足 App 上架或合规要求。

即时通讯服务返回注销成功结果后，与用户 ID 相关数据即被删除。您可以向即时通讯服务端查询所有已注销用户的 ID。如有需要，您可以重新激活已被注销的用户 ID（注意，用户个人数据无法被恢复）。

仅 IM Server API 提供上述能力。

用户信息

用户信息泛指用户的昵称、头像，以及群组的群昵称、群头像等数据。您需要自行维护用户信息数据。即时通讯服务端不提供用户信息托管维护服务。

IMKit SDK 中仅提供与展示用户信息相关的客户端接口。

好友关系

App 用户之间的好友关系需要由应用服务器（App Server）自行维护。即时通讯服务不会同步或保存 App 端的好友关系数据。

如果需要对客户端用户之间的消息收发行为进行限制（例如，App 的所有 `userId` 泄漏，导致某个恶意用户可越过好友关系向任意用户发送消息），可以考虑使用用户白名单服务。用户一旦开启并设置白名单，则仅可接收来自该白名单中用户的消息。

用户管理接口

功能分类	功能描述	服务端 API
注册用户	使用 App 用户的用户 ID 向即时通讯服务端换取 Token。	注册用户
删除用户	参见上文**删除用户**。	不提供该 API
废弃 Token	废弃在特定时间点之前获取的 Token。	作废 Token
注销用户	注销用户是指在即时通讯服务中停用用户 ID，并删除用户个人数据。	注销用户
查询已注销用户	获取已注销的用户 ID 列表。	查询已注销用户
重新激活用户 ID	在即时通讯服务中重新启用已注销用户的 ID。	重新激活用户 ID
设置即时通讯服务端的用户信息	设置在即时通讯的推送服务中使用的用户名称与头像。	未提供单独的设置接口。在注册用户时必须提供用户信息。
获取即时通讯服务端的用户信息	获取用户在即时通讯服务中注册的信息，包括用户创建时间和服务端的推送服务使用的用户名称、头像 URL。	获取信息
修改即时通讯服务端的用户信息	修改在即时通讯的推送服务中使用的用户名称与头像。	修改信息
封禁用户	禁止用户连接到即时通讯服务，并立即断开连接。可按时长解封或主动解封。查询被封禁用户的用户 ID、封禁结束时间。	添加封禁用户、解除封禁用户、查询封禁用户
查询用户在线状态	查询某用户的在线状态。	查询在线状态
加入黑名单	在用户的黑名单列表中添加用户。在 A 用户黑名单的用户无法向 A 发送消息。	加入黑名单
移出黑名单	在用户的黑名单中移除用户。	移出黑名单
获取黑名单列表	获取用户的黑名单列表。	查询黑名单
用户白名单	用户一旦开启并设置白名单，则仅可接收来自该白名单中用户的消息。	开启用户白名单、用户白名单状态查询、添加白名单、移出白名单、查询白名单

仅黑名单功能提供客户端 API。上表其他接口均只提供服务端 API。

用户服务配置　　新用户接收注册前广播消息

更新时间:2024-08-30

默认新注册的用户可以接收到注册前 7 天内的广播消息。您可以从控制台免费基础功能 页面关闭该服务。

新用户接收注册前广播消息开关影响以下接口发送的消息：

- 全量用户通知消息：`/message/broadcast.json`
- 按标签用户推送：`/push.json`
- 按应用包名推送：`/push.json`

订阅用户在线状态

在线状态订阅是即时通讯服务提供的回调服务，支持将用户状态变更（上线、离线、登出）实时同步到指定的服务器。开发环境下可免费使用。生产环境下，IM 旗舰版或IM 尊享版可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。

访问控制台 IM 服务管理 页面，切换到普通服务标签下，可打开订阅用户在线状态开关。开通服务时，请配置可正常访问的回调接收地址。如果您的网络有 IP 访问限制，请参考接收服务端回调配置 IP 白名单，否则无法正常接收服务端回调。

注册用户

更新时间:2024-08-30

客户端在与即时通讯服务端建立连接时必须传入用户在即时通讯服务的身份验证令牌（Token）。Token 是用户在即时通讯服务中的唯一身份标识，无论是使用即时通讯服务或是实时音视频服务，均需要使用该 Token。获取 Token 即表示一个用户在即时通讯服务中完成注册。

在开始之前，我们建议先阅读用户概述，以了解用户管理的相关接口、服务及配置。

获取 Token

客户端 SDK 不提供获取 Token 的 API。您需要在应用服务端（App Server）集成即时通讯服务端获取 Token API，实现获取 Token 的业务逻辑。

成功获取 Token 后，由您的 App Server 将 Token 分发给客户端。客户端在连接即时通讯服务端时使用。Token 在有效期内都可以正常使用。关于 Token 有效期的详细描述，详见作废 Token。

同一个用户 ID 可多次获取 Token，如果 Token 在有效期内，均可用于连接即时通讯服务。App 获取、使用 Token 进行连接、认证的参考方案与 UML 流程图可参考下文**使用 Token**。

> ⚠ 危险
>
> 一旦您在控制台刷新 App Secret ，App Key 下已获取的 Token 均会失效。请使用新的 App Secret 生成 API 签名，并重新获取 Token。

请求方法

通过 App 层定义的用户 ID（`userId`）换取即时通讯服务中使用的身份验证 Token。同一用户 ID 如需重新获取 Token 使用同一接口。

POST： https://数据中心域名/user/getToken.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 200 次

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

获取 Token 时传入的名称（name）和 头像（portraitUri），仅供移动端远程推送时使用。如需变更名称和头像信息，建议通过修改用户信息的方法进行变更。在重新获取 Token 时如果传入新的数据，则会覆盖旧的名称与头像数据。

参数	类型	必传	说明
userId	String	是	App 自行定义的用户 ID，用于换取 Token。支持大小写英文字母与数字的组合，最大长度 64 字节。
name	String	是	推送服务使用的用户名称。不区分符号、英文字符、中文字符，统一限制最多 64 个字符。注意：该 name 字段仅用于推送服务，作为在移动客户端推送通知中默认显示的用户名称。因为即时通讯服务端不提供用户信息托管服务，所以不支持客户端 SDK 主动获取该用户名称数据。
portraitUri	String	否	用户头像 URI，最大长度 1024 字节。注意：因为即时通讯服务端不提供用户信息托管服务，所以不支持客户端 SDK 主动获取该用户头像数据。

提示

- 如果发送单聊会话消息，只需要 userId 即可定位到接收方。考虑到安全问题，强烈建议开发者注册用户时不要使用连续的 userId，以防止被恶意利用。
- 部分旧版 Server SDK 中 `portraitUri` 字段为必填项；若不希望填写该字段，请升级 Server SDK 到最新版，或者传一个固定字符即可，比如："_"。

请求示例

```
POST /user/getToken.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&name=Ironman&portraitUri=http%3A%2F%2Fabc.com%2Fmyportrait.jpg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
token	String	用户身份验证 Token，长度在 256 字节以内，可以保存应用内。Token 中携带 IM 服务动态导航地址，开发者不需要进行处理。
userId	String	返回输入参数中提供的用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200, "userId":"jlk456j5", "token":"sfd9823ihufi"}
```

使用 Token

即时通讯服务不管理任何 App 用户信息。用户的注册仅需通过用户 ID 交换 Token 即可。

您的 App 客户端需要在建立 IM 连接时处理 Token 失效的情况，并且在 Token 失效时请求 App Server 重新获取 Token。

以下提供了基于 Token 认证与连接的方案，供您参考。

首次连接时获取 Token

应用获取 Token 后，根据情况可选择在应用本地保留当前用户的 Token。流程如下：

非首次连接时使用已有 Token

后续登录过程中，不必再向即时通讯服务端请求 Token，由 App Server 直接提供之前保存过的 Token。

> 提示
>
> - 请实现重新向服务器获取 Token 的代码逻辑，以处理 Token 失效的情况。
> - Token 具有有效期，默认为永久有效，可在控制台进行修改。
> - 刷新 App Secrect、作废 Token 均可使已有 Token 失效。

如果您的 App 是免登录设计，也可以将 Token 保存在 App 本地（注意保证本地数据存储安全），直接登录。

作废 Token

更新时间:2024-08-30

设置指定用户在某个时间点之前获取的 Token 失效。

同一个用户 ID 多次获取 Token（例如重新获取 Token），所有已获取 Token 在有效期内均可以正常用于连接。您可以根据业务需求作废指定用户的所有 Token。作废 Token 的操作不影响使用该 Token 建立的已有连接。

- 客户端用户一旦重新连接（包括弱网情况下断线自动重连），会返回 Token 失效错误。此时应重新获取 Token。
- 作废 Token 后，如果 App 用户在客户端使用已废弃的 Token 建立 IM 连接，会返回 Token 失效的错误。此时应重新获取 Token。

Token 有效期

Token 默认为永久有效。有效期可在控制台进行修改。

请求方法

POST： https://数据中心域名/user/token/expire.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	需要设置 Token 失效的用户 ID，支持设置多个最多不超过 20 个。

参数	类型	必传	说明
time	Long	是	Token 的过期时间，以毫秒为单位。过期时间不得超过融云服务器系统当前时间戳，超过将自动转换为系统当前时间。用户在此时间前获取的 Token 全部过期。您将无法通过过期的 Token 建立新的连接，但已经建立的连接不受影响。

请求示例

```
POST /user/token/expire.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=test1&userId=test2&time=1615362955344
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

获取用户信息

更新时间:2024-08-30

获取用户在即时通讯服务中注册的相关信息，包括：用户名称、用户头像 URL、用户创建时间。

请求方法

POST： https://数据中心域名/user/info.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID

请求示例

```
POST /user/info.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
userName	String	用户名称。
userPortrait	String	用户头像地址。
createTime	String	用户创建时间

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"userName":"123",
"userPortrait":"",
"createTime":"2016-05-24 10:38:19"
}
```

修改用户信息

更新时间:2024-08-30

修改在即时通讯服务端保存的用户名称（name）和 头像（portraitUri）数据。

注意事项

重要

- 即时通讯服务端保存的用户名称（`name`）和 头像（`portraitUri`）数据仅用于推送服务。`name` 字段在移动客户端推送通知中作为默认显示的用户名称。因为即时通讯服务端不提供用户信息托管服务，所以客户端 SDK 不提供主动获取该用户名称数据的方法。
- 刷新用户信息后 `name` 字段和 `portraitUri` 字段实时生效，Push 推送时将显示刷新后的用户名称。
- App 中展示的用户昵称、头像数据需要您自行实现。如果客户端使用 IMKit SDK，可使用「用户信息提供者」类将 App 层的数据提供给 SDK，用于展示。

请求方法

POST： https://数据中心域名/user/refresh.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，支持大小写英文字母与数字的组合，最大长度 64 字节。`userId` 是用户在 App 中的唯一标识，必须保证在同一个 App 内不重复，重复的用户 ID 将被当作是同一用户。
name	String	否	用户名称，最大长度 64 个字符（不区分符号、英文字符、中文字符，统一限制最多 64 个字符）。用来在 Push 推送时，显示用户的名称，不提供则不进行刷新。
portraitUri	String	否	用户头像 URI，最大长度 1024 字节。注意：因为即时通讯服务端不提供用户信息托管服务，所以不支持客户端 SDK 主动获取该用户头像数据。

请求示例

```
POST /user/refresh.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&name=newname&portraitUri=http%3A%2F%2Fabc.com%2Fmynewportrait.jpg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

设置用户信息

更新时间:2024-08-30

设置应用用户的信息，包括用户的基本信息（userProfile）和拓展信息（userExtProfile）。基本信息包括是所有用户都会有的共性信息，比如昵称、性别、邮箱、生日等信息，拓展信息是可以自定义的额外信息。

开通服务

调用此接口前，您须在控制台开通信息托管服务。

请求方法

POST： https://数据中心域名/user/profile/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID
userProfile	String	是	用户基本信息，为 JSON 数据。
userExtProfile	String	否	用户扩展信息，为 JSON 数据。key 的长度不超过 32 个字符，key的前缀必须加 ext_， value 的长度，value不超过 256 个字符，KV的长度有配置，默认最多 20个。

userProfile 中的 key 如下：

key 值	类型	长度/取值范围	描述
uniqueId	String	长度不超过 32 个字符	用户应用号
name	String	长度不超过 32 个字符	昵称
portraitUri	String	长度不超过 128 个字符	头像地址
email	String	长度不超过 128 个字符	Email 地址
birthday	String	长度不超过 32 个字符	生日
gender	Int	取值范为： • 0：未知 • 1：男 • 2： 女	性别
location	String	长度不超过 32 个字符	所在地
role	Int	0~100	角色
level	Int	0~100	级别

- 昵称：name
- 头像地址：portraitUri
- 用户应用号：
- Email：email
- 生日：birthday
- 性别：gender
- 所在地：location
- 角色：role
- 级别：level

请求示例

```
POST /user/profile/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=u1&userProfile={"AppName":"testAppName"}&userExtProfile={"ext_Profile1":"testpro1"}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
`code`	Number	返回码，200 为正常。
`profileKey`	String	如果返回的 code 不是 200，则设置失败，返回具体失败的 key。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
}
```

批量查询用户资料

更新时间:2024-08-30

批量获取指定用户的信息，返回的信息包括用户的基本信息和拓展信息。

开通服务

调用此接口前，您须在控制台开通信息托管服务。

请求方法

POST： https://数据中心域名/user/profile/batch/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	Array	是	用户 ID，最多一次100个。

请求示例

```
POST /user/profile/batch/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=u1&userId=u2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
userList	JSONArray	用户列表信息
userList 结构		

userList 结构的组成：

返回值	返回类型	说明
userList[i].userId	String	用户ID
userList[i].version	long	用户托管信息版本号
userList[i].userProfile	String	用户基本信息，为 JSON 数据
userList[i].userExtProfile	String	用户扩展信息，为 JSON 数据

`userProfile` 中的 key 如下：

key 值	类型	长度/取值范围	描述
uniqueId	String	长度不超过 32 个字符	用户应用号
name	String	长度不超过 32 个字符	昵称
portraitUri	String	长度不超过 128 个字符	头像地址
email	String	长度不超过 128 个字符	Email 地址
birthday	String	长度不超过 32 个字符	生日
gender	Int	取值范为： • `0`：未知 • `1`：男 • `2`：女	性别
location	String	长度不超过 32 个字符	所在地
role	Int	0~100	角色
level	Int	0~100	级别

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"userList": [
{
"userId": "u1",
"version": 12343433499942,
"userProfile": {"birthday":"20011221","level":2},
"userExtProfile": {"ext_1":"testext"}
},
{
"userId": "u2",
"userProfile": {"gender":1,"level":5}
}
]
}
```

分页获取应用全部用户列表

更新时间:2024-08-30

获取所有托管在融云服务器上的应用用户的列表，返回结果分页展示。

开通服务

调用此接口前，您须在控制台开通信息托管服务。

请求方法

POST： https://数据中心域名/user/profile/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
page	Number	否	默认 1
size	Number	否	默认20，最大 100

参数	类型	必传	说明
order	Number	否	根据注册时间的排序机制，默认正序，0为正序，1为倒序

请求示例

```
POST /user/profile/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

page=1&count=30&order=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
userList	JSONArray	用户列表信息
userList结构		

userList 结构的组成如下：

返回值	返回类型	说明
userList[i].userId	String	用户ID
userList[i].userProfile	String	用户基本信息, 为 JSON 数据
userList[i].userExtProfile	String	用户扩展信息, 为 JSON 数据

`userProfile` 中的 key 如下：

key 值	类型	长度/取值范围	描述
uniqueId	String	长度不超过 32 个字符	用户应用号
name	String	长度不超过 32 个字符	昵称
portraitUri	String	长度不超过 128 个字符	头像地址
email	String	长度不超过 128 个字符	Email 地址
birthday	String	长度不超过 32 个字符	生日
gender	Int	取值范为： • `0`：未知 • `1`：男 • `2`： 女	性别
location	String	长度不超过 32 个字符	所在地

key 值	类型	长度/取值范围	描述
role	Int	0~100	角色
level	Int	0~100	级别

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"userList": [
{
"userId": "u01",
"userProfile": {"appName":"testAppName","level":2},
"userExtProfile": {"ext_Profile":"testExt"}
}
]
}
```

清除用户托管信息

更新时间:2024-08-30

清除托管在融云服务器上的应用用户信息。

开通服务

调用此接口前，您须在控制台开通信息托管服务。

请求方法

POST： https://数据中心域名/user/profile/clean.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	Array	否	需要清除托管信息的用户ID数组，一次最多20个用户。

请求示例

```
POST /user/profile/clean.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=user1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200
}
```

同步托管资料变更状态

更新时间:2024-08-30

当应用用户资料发生变更时，您可以把实时用户资料同步到您的应用服务器上。

开通服务

调用此接口前，您须在控制台开通信息托管服务，并且设置好服务器的回调地址。

请求方法

POST： https://数据中心域名/user/profile/sync
签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID
type	int	是	变更类型： 0:清除 1:修改
time	Long	是	修改时间
userProfile	String	否	修改的用户基本信息，为 JSON 数据

参数	类型	必传	说明
userExtProfile	String	否	修改的用户扩展信息，为 JSON 数据

请求示例

```
POST /user/profile/sync HTTP/1.1
Host: example.com
Content-Type: application/json


{
"userId":"uid1",
"time":1574476797772,
"userProfile": {"uniqueId":"testAppName","level":2},
"userExtProfile": {"ext_Profile":"testExt"}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
}
```

订阅用户在线状态

更新时间:2024-08-30

用户在线状态订阅是即时通讯服务端提供的回调服务，采用 Webhook 机制。您需要提前在控制台注册您的回调地址。注册完成后，即时通讯服务端会在每一次用户状态变更（上线、离线、登出）时，将用户状态变化事件实时通知您的服务器。

在应用中需要实时展示用户在线、离线状态时，可利用该 Webhook 返回的事件修改用户状态。如需直接查询某用户的在线状态，可以主动调用 Server API 接口查询用户在线状态。

开通服务

使用**用户在线状态订阅**功能前，您需要为应用和当前环境的 App Key 开通服务。详见用户服务配置。

开通服务时，请配置可正常访问的回调接收地址。如果您的网络有 IP 访问限制，请务必配置 IP 白名单，否则无法正常接收服务端回调。

回调方法

请求方法： POST

数据格式： `application/json`

即时通讯服务端会在 POST 请求 URL 中添加签名参数，您可通过签名验证调用者身份和数据有效性，详细参见 服务端回调签名。

回调正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	说明
userid	String	用户 ID。
status	String	用户状态变更事件。`0` - 用户已上线。`1` - 用户已离线。`2`：用户已登出。
os	String	操作系统：iOS、Android、Websocket、PC、MiniProgram（小程序），用户上线时同步。
time	Int	发生时间。
clientIp	String	用户当前的 IP 地址及端口。
sessionId	String	一个连接的唯一 ID。如果同一用户在多端同时在线，则会有多个连接，可使用此 ID 进行区分。

用户状态变更事件（`status`）参数说明：

- `0` - 用户已上线，该事件表示客户端成功连接到即时通讯服务，即应用程序调用了 `connect` 方法，并连接成功。
- `1` - 用户已离线，该事件表示客户端已经断开与即时通讯服务的连接，即应用程序调用了 `disconnect` 方法，或因异常情况断开连接。部分极端情况下，离线状态会延迟 5 分钟同步，例如应用程序连接了代理网络，应用异常断网，但此时代理服

务与即时通讯服务的连接缓存仍然存在，这种情况下会导致用户离线事件延迟同步。

- `2` - 用户已登出，该事件表示客户端已经注销登录状态，即应用程序主动调用了注销方法（`logout`）注销登录，注销登录后即时通讯服务不会为该用户触发远程推送通知。

用户已离线事件（1）默认还覆盖以下场景：

- **多设备互踢场景**：App 用户在 A 设备登录后，又在 B 设备上登录，B 将 A 踢下线后，用户在 A 设备的离线状态也会触发服务端同步用户已离线事件（`1`）。
- **多平台登录场景**：App 用户在不同平台登录，在任意平台上断开连接后会触发服务端同步用户已离线事件（`1`）。

> 重要
>
> - 回调正文中的 `status` 参数代表用户状态变更事件，其取值与 查询用户在线状态 API 接口返回结果中代表用户当前状态的 `status` 取值（在线或不在线）不同，请注意区分。
> - 部分早期开通订阅用户在线状态服务的客户可能发现同步行为与上述描述不一致。如有需要，请提交工单 申请调整在线状态订阅-离线状态全通知开关状态。

回调请求示例

假设您在开通服务页面配置的接收地址：`http://example.com/online_status.php`

```
POST /online_status.php?
appKey=someappKey&timestamp=1408710653491&nonce=14314&signature=45beb7cc7307889a8e711219a47b7cf6a5b000e8
HTTP/1.1
Host: example.com
Content-Type: application/json

[
{
"userid": "apert60541",
"status": "1",
"os": "iOS",
"time": 1437982625625,
"clientIp":"172.20.10.94:41748",
"sessionId":"0O6Bvfvik_L9R_NxR6GqQb"
},
{
"userid": "apert60592",
"status": "1",
"os": "Android",
"time": 1437982625625,
"clientIp":"172.20.10.94:41748",
"sessionId":"0O9Fvfvik_L9R_NaR3GqYz"
},
{
"userid": "apert60533",
"status": "2",
"os": "Android",
"time": 1437982625625,
"clientIp":"172.20.10.94:41748",
"sessionId":"0O4Hvfvik_L3B_XaR7GxAf"
}
]
```

响应回调请求

提示

- 只要有 HTTP `200 OK` 成功响应，服务端会认为状态已经同步。
- 如果应答超时 5 秒，服务端会再尝试推送 2 次，如果仍然失败，将不再同步此条状态。
- 如短时间内有大面积超时，将暂时停止请求您的服务器，1 分钟后会继续发送回调请求。异常断网情况下的会延迟 5 分钟同步。

查询用户在线状态

更新时间:2024-08-30

主动查询某用户的在线状态。

通过 Server API 接口主动查询某用户的在线状态，接口调用有频率限制，不能频繁调用，可用于管理平台显示用户状态，或用于检查指定用户的客户端是否已成功连接即时通讯服务。

> 提示
>
> 如需即时通讯服务实时发送用户状态变更通知，可开通订阅用户在线状态服务。详见订阅用户在线状态。

请求方法

POST https://数据中心域名/user/checkOnline.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。

请求示例

```
POST /user/checkOnline.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
status	String	用户状态。1 - 用户当前在线。0 - 用户当前不在线。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"status":"1"}
```

设置用户订阅关系

更新时间:2024-08-30

融云允许您订阅用户在线状态，您不仅可以通过客户端来订阅在线状态，也可以通过服务端来设置订阅关系。

开通服务

您可以通过控制台开通服务。在融云控制台，选择 **IM 服务 -> IM 服务管理 -> 客户端用户在线状态订阅**，开启服务。

请求方法

POST： https://数据中心域名/user/subscribe/set.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
subType	int	否	默认为 1，表示状态订阅
userId	String	是	订阅者 ID
subUserIds	String	是	被订阅用户的 ID，最多可以一次性订阅 200 个用户，多个用户 ID 之间用逗号分割。

参数	类型	必传	说明
opType	int	是	操作类型，0 表示添加订阅，1 表示取消订阅。
expiry	long	否	订阅有效期，操作类型为 0 必填。单位为秒。

请求示例

```
POST /user/subscribe/set.json HTTP/1.1
Host: api-cn.ronghub.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=uid1&subUserIds=s_user1,s_user2&opType=0&expiry=60&subType=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为处理成功。
subUserIds	string[]	返回码 26021，仅返回被订阅用户达到限制的列表。

错误码

code	httpStatus	说明
200	200	成功
26001	400	错误请求参数，参数不符合要求
26003	500	内部错误
26022	403	订阅用户超限
26021	200	被订阅用户达到上限
26020	401	未开启订阅功能

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200}
```

查询用户订阅关系

更新时间:2024-08-30

您可以通过本接口，查询用户的订阅关系。

开通服务

您可以通过控制台开通服务。在融云控制台，选择 **IM 服务 -> IM 服务管理 -> 客户端用户在线状态订阅**，开启服务。

请求方法

POST： https://数据中心域名/user/subscribe/query.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
subType	int	否	默认 1 为状态订阅
userId	String	是	订阅者 ID
pageToken	String	否	分页标识，默认空即第一页，设置上一页返回的 `pageToken`。

参数	类型	必传	说明
pageSize	int	否	默认 200，最大为 200，最小 50。

请求示例

```
POST /user/subscribe/update.json HTTP/1.1
Host: api-cn.ronghub.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=uid1
```

返回结果

返回值	返回类型	说明
code	Int	返回码，200 为处理成功。
hasNext	boolean	是否有下一页
subscriptions	List	订阅列表
pageToken	String	下一页 PageToken。

- UserSubscriptionRecord 结构说明：

参数	返回类型	说明
subUserIds	String[]	订阅用户
subType	int	订阅类型
subTime	long	订阅时间
expiry	Long	订阅有效期

错误码

code	httpStatus	说明
200	200	成功
26001	400	错误请求参数，参数不符合要求
26003	500	内部错误
26022	403	订阅用户超限
26021	200	被订阅用户达到上限
26020	401	未开启订阅功能

返回结果示例

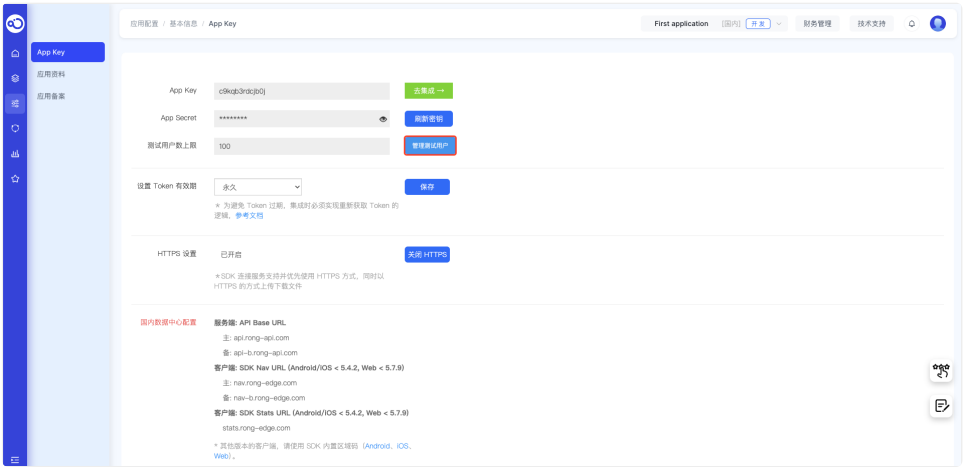
```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{
"code": 200,
"hasNext": false,
"pageToken": "1712909941322_2_3881222167",
"subscriptions": [
{
"subUserIds": [
"4",
"5"
],
"subType": 1,
"subTime": 1712909941322,
"expiry": 1000000
}
]
}
```

删除用户　　开发环境

更新时间:2024-08-30

注册用户上限为 100 个。在应用的开发环境中，您可以访问控制台的 App Key 页面，点击管理测试用户，删除当前已注册的测试用户。



生产环境

生产环境的注册用户数上限为 100，IM 旗舰版与 IM 尊享版不限制注册用户数。在应用的生产环境中，不提供删除功能。

注销用户

更新时间:2024-08-30

在即时通讯服务中注销用户。用户注销后，用户在即时通讯服务中存储的所有个人数据都会被删除，并且无法恢复，请谨慎操作。

适用场景

- **用户注销 App 账号**：用于实现 App 注销账号功能。用户发起注销操作，成功后用户立即退出 App，并且无法再重新登录，用户所有相关的数据都会被删除，无法恢复。
- **员工离职删除账号**：用于实现办公应用中员工离职后删除账号的需求。管理员发起注销操作，注销成功后，离职员工无法再使用该账号登录，并且账号相关的数据会被转移或删除。

数据清理范围

即时通讯服务仅会删除已注销用户在即时通讯服务端存储的个人数据：

- 个人信息：删除在即时通讯服务端存储的用户头像、昵称
- 消息数据：删除在即时通讯服务端存储的历史消息、离线消息、补偿消息、离线消息时长设置
- 会话信息：删除在即时通讯服务端存储的会话与会话列表、会话标签、会话设置（置顶、免打扰）
- 推送信息：关闭推送，删除用户推送的设备信息
- 用户设置：用户的单聊黑白名单，推送语言偏好设置，是否显示推送详情

即时通讯服务不会删除已注销用户在即时通讯服务端存储的非个人数据，例如群组关系、超级群关系。

> 提示
>
> 注销操作不删除该用户 ID 相关的群组关系、超级群关系。如 App 业务需要，可调用相关的 IM Server API 退出用户所在群组或超级群。

注销影响与结果

发起注销后，对客户端与服务端的影响如下：

- **客户端**：立即断开连接与即时通讯服务的 IM 连接。如果多端同时在线，则所有端立即断开连接。用户无法再获取 Token，无法继续通过通过 SDK 发送、接收消息。
- **客户端**：SDK 的连接状态监听返回连接断开的状态码。
- **服务端**：通过 Server API 发送消息时，即时通讯服务不会判断发件人的用户 ID 是否已注销。App 需要自行处理。
- **服务端回调**：如果 App Key 已开通**用户在线状态订阅**功能，服务端会在用户的客户端断开连接后向您同步在线状态（用户登出）。

如果 App Key 已开通用户注销或激活状态回调功能，服务端会将注销完成的状态同步到您指定的服务器。

请求方法

POST： https://数据中心域名/user/deactivate.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制：每秒钟限 100 个用户

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	被注销用户 ID，最多一次 100 个。

请求示例

```
POST /user/deactivate.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=uid1,uid2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。每个用户ID操作结果通过用户注销与激活状态回调传递.
operateId	String	操作 ID，为当前操作的唯一标识。开通用户注销与激活状态回调后，回调请求正文中会携带此参数。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200,"operateId":"C70B-B1D6-82E7-5SBO"}
```

查询已注销用户

更新时间:2024-08-30

获取已注销的用户 ID 列表，返回已被发起注销的用户 ID 的列表。

> 提示
>
> 返回的列表中会包含仍在注销中的用户。如果收取注销结果，请使用用户注销与激活状态回调。

请求方法

POST： https://数据中心域名/user/deactivate/query.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
pageNo	Number	否	分页获取注销用户列表时的当前页数，默认 1，最小 1。
pageSize	Number	否	分页获取注销用户列表时的每页行数，默认 50，最小 1，最大 50。

请求示例

```
POST /user/deactivate/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

pageSize=50&pageNo=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	Array of Strings	已注销的用户 ID 列表

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200,"users":["uid1","uid2"]}
```

重新激活用户 ID

更新时间:2024-08-30

在即时通讯服务中重新启用已注销用户的 ID，支持批量操作。

如果获取每个用户 ID 的操作结果，请提前在控制台注册用户注销与激活状态回调。

> 提示
>
> - 注销用户后，与 ID 相关的个人数据已被删除。重新激活后，已删除的个人数据不会被恢复。
> - 发起注销请求的一个小时内，用户 ID 无法被重新激活。

请求方法

POST： https://数据中心域名/user/reactivate.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制：每秒钟限 100 个用户

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	激活用户 ID，单次请求最多传入 100 个用户 ID。

请求示例

```
POST /user/reactivate.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=uid1,uid2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。每个用户 ID 操作结果通过用户注销与激活状态回调传递。
operateId	String	操作 ID，为当前操作的唯一标识。开通用户注销与激活状态回调后，回调请求正文中会携带此参数。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200,"operateId":"C70B-B1D6-82E8-5SBO"}
```

用户注销与激活状态回调

更新时间:2024-08-30

即时通讯服务支持在用户注销或重新激活完成时，将处理结果同步到您的应用服务器，便于您知晓单个或批量注销与激活用户的处理进度与结果。

- 发起注销后，服务端会在 15 分钟内通过回调通知注销结果。
- 重新激活用户 ID、重复注销、重复激活的情况下，即时通讯服务端会实时发起回调通知。

开通服务

使用**用户注销与激活状态回调**功能前，请确认已为当前 App Key 开通相关服务。您可以在控制台**免费基础功能**页面启用并配置该回调服务。

开通服务时，请配置可正常访问的回调接收地址。如果您的网络有 IP 访问限制，请务必配置 IP 白名单，否则无法正常接收服务端回调。

回调方法

请求方法： POST

数据格式： `application/x-www-form-urlencoded`

即时通讯服务端会在 POST 请求 URL 中添加签名参数，您可通过签名验证调用者身份和数据有效性，详细参见 服务端回调签名。

回调正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，包含以下 HTTP 表单参数：

参数	类型	说明
userId	String	用户 ID
operateId	String	操作 ID，为当前操作的唯一标识。与**注销用户**与**重新激活用户 ID** API 返回结果中的操作 ID 一致。
type	Int	操作类型。`0`：注销用户；`1`：重新激活用户 ID。
code	String	激活/注销用户的服务处理结果码。`0` 表示成功。其他错误码见下表。
time	Long	操作时间

- **code** 参数说明（激活/注销用户的服务处理结果码）

Code	描述	备注
24353	重复注销	调用注销接口，传入已完成注销的用户 ID 时，返回该错误码。

Code	描述	备注
24354	重复激活	调用重新激活用户 ID 的接口，传入已激活的用户 ID 时，返回该错误码。
24356	用户注销中	调用注销接口，传入正在注销流程中的用户 ID 时，返回该错误码。
其他	未知错误	

回调请求示例

以下示例假设您在开通服务页面配置的回调接收地址为 `http://example.com/user_actiavtation_status_sync.php`。

```
POST /user_actiavtation_status_sync.php?
appKey=uwd1c0sxdlx2&signTimestamp=1681202504348&nonce=14314&signature=45beb7cc7307889a8e711219a47b7cf6a5
8&appKey=uwd1c0sxdlx2 HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
User-Agent: RongCloud/1.0

userId=uid1&operateId=C70B-B1D6-82E7-5SBO&type=0&code=0&time=1681202504348
```

响应回调请求

> 提示
>
> - 只要有 HTTP `200 OK` 成功响应，服务端会认为状态已经同步。
> - 如果应答超时 5 秒，服务端会再尝试推送 2 次，如果仍然失败，将不再同步此条状态。
> - 如短时间内有大面积超时，将暂时停止请求您的服务器，1 分钟后会继续发送回调请求。异常断网情况下的会延迟 5 分钟同步。

封禁用户

更新时间:2024-08-30

对 App 用户进行封禁处理，一般用于 App 用户违规情形。

- 封禁时必须设置封禁时间范围，以分钟为最小单位，最长封禁 30 天（43200 分钟）。
- 不支持永久封禁。
- 封禁期满后，自动解除封禁。也可以主动调用解除封禁方法进行解封。

封禁指定的单个或多个用户。封禁操作即刻生效，影响如下：

- 如果被封禁用户此时已经与即时通讯服务保持连接，封禁后会即时断开该用户的 IM 连接。
- 被封禁用户无法与即时通讯服务建立 IM 连接，无法主动使用 IM 服务。
- 被封禁期间，其他用户可向被封禁用户发送消息，但被封禁用户无法接收消息，无法收到推送通知。解封后用户可正常连接并使用即时通讯服务，再次上线可以收到被封禁期间的离线消息[?]。请注意，离线消息最长存储 7 天，如果 7 天内客户端都没有上线，服务端将抛弃掉过期的消息。

请求方法

POST： https://数据中心域名/user/block.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，支持一次封禁多个用户，最多不超过 20 个。
minute	Number	是	封禁时长，单位为分钟，最大值为 43200 分钟。

请求示例

```
POST /user/block.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=BWSZNmXBD&userId=jlk456j5&minute=10
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

解除封禁

更新时间:2024-08-30

解除对 App 用户的封禁，单次可解封多个用户。

解除封禁后，用户可重新建立 IM 连接，使用 IM 服务正常收发消息。上线时可收到被封禁期间的离线消息[?]。请注意，离线消息最长存储 7 天，如果 7 天内客户端都没有上线，服务端将抛弃掉过期的消息。

请求方法

POST： https://数据中心域名/user/unblock.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，支持一次解除多个用户，最多不超过 20 个。

请求示例

```
POST /user/unblock.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&userId=jlk45211
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

获取封禁用户列表

更新时间:2024-08-30

查询被封禁用户信息，包括被封禁用户的用户 ID、封禁结束时间。

封禁期满后会自动解除封禁。也可以主动调用解除封禁方法进行解封。

请求方法

POST： https://数据中心域名/user/block/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
page	Number	否	分页获取封禁用户列表时当前页数，不传或传入 0 时不做分页处理，默认获取前 1000 个被封禁的用户列表，按封禁结束时间倒序排序。
size	Number	否	分页获取封禁用户列表时每页行数，不传时默认为 50 条。

请求示例

```
POST /user/block/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

page=1&size=50
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	Array	被封禁用户数组。

返回值	返回类型	说明
users[i].userId	String	被封禁用户 ID。
users[i].blockEndTime	String	封禁结束时间。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"users":[{"userId":"jlk456j5","blockEndTime":"2015-01-11 01:28:20"}]}
```

设置用户单聊禁言

更新时间:2024-08-30

设置指定用户单聊会话场景下禁言，或解除禁言。如果用户被设置为单聊禁言状态，则该用户无法发送单聊消息。

> 提示
>
> 服务端（Server API）发送单聊消息接口不受禁言状态的限制，被禁言用户可通过 Server API 往单聊会话中发送消息。

请求方法

POST： https://数据中心域名/user/chat/fb/set.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	被禁言用户 ID，支持批量设置，最多不超过 1000 个。
state	Int	是	禁言状态，0 解除禁言、1 添加禁言
type	String	是	会话类型，目前支持单聊会话 PERSON

请求示例

```
POST /user/chat/fb/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=123&userId=456&state=1&type=PERSON
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询单聊禁言用户列表

更新时间:2024-08-30

查询在单聊会话场景下被禁言的用户。

请求方法

POST： https://数据中心域名/user/chat/fb/querylist.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
num	Int	否	获取行数，默认为 100，最大支持 200 个。
offset	Int	否	查询开始位置，默认为 0。
type	String	是	会话类型，目前支持单聊会话 PERSON。

请求示例

```
POST /user/chat/fb/querylist.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

num=10&offset=0&type=PERSON
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
total	Int	被禁言用户总数。
users	String[]	被禁言用户数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"total": 3,
"users": [
"all_00001",
"all_00002",
"all_00003"
]
}
```

设置用户标签

更新时间:2024-08-30

为单个用户设置标签。注意，用户标签仅限于在以下功能中使用：

- 发送标签用户通知
- 发送全量用户通知
- 推送 Plus

> 提示
> - 每次设置时需要传入用户的全量标签数据。每个用户最多可设置 20 个标签。
> - `tags` 字段传入空数组表示清除该用户的所有标签。

请求方法

POST： https://数据中心域名/user/tag/set.json

调用频率：每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
userId	String	是	用户 ID。
tags	String[]	是	用户标签，一个用户最多添加 20 个标签，每个 tag 最大不能超过 40 个字节，标签中不能包含特殊字符。每次设置时需要传入用户的全量标签数据。传入空数组表示清除该用户的所有标签。

请求示例

```
POST /user/tag/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408706337
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{"userId":"31232","tags":["bj","男"]}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

批量设置用户标签

更新时间:2024-08-30

为指定的单个用户或多个用户设置标签。注意，用户标签仅限于在以下功能中使用：

- 发送标签用户通知
- 发送全量用户通知
- 推送 Plus

> 提示
>
> - 每次设置时需要传入全量标签数据。传入的所有用户（`userIds`）的标签数据都被会覆盖更新。每个用户最多可设置 20 个标签。
> - `tags` 字段传入空数组表示清除指定用户的所有标签。

请求方法

POST： https://数据中心域名/user/tag/batch/set.json

调用频率：每秒钟限 10 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
userIds	String[]	是	用户 ID，一次最多支持 1000 个用户。传入的所有用户的标签都被会覆盖更新为 `tags` 中的标签。
tags	String[]	是	用户标签，一个用户最多添加 20 个标签，每个 tag 最大不能超过 40 个字节，标签中不能包含特殊字符。每次设置时需要传入全量标签数据。传入空数组表示清除用户的所有标签。

请求示例

```
POST /user/tag/batch/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408706337
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{"userIds":["id1","id2"],"tags":["bj","男"]}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

获取用户标签

更新时间:2024-08-30

获取用户标签。

请求方法

POST： https://数据中心域名/user/tags/get.json

调用频率：每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userIds	String[]	是	用户 ID，一次最多支持 50 个用户。

请求示例

```
POST /user/tags/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408706337
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/x-www-form-urlencoded

userIds=111&userIds=222
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
result	Object	Map 结构的 JSON 对象，包含了用户 ID 与用户所有的标签数组的映射关系。用户 ID 为键，标签数组为值，其中包含字符串类型的标签。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"result": {
"u01": [
"tag4",
"tag3"
],
"u02": [
"tag4",
"tag3"
]
}
}
```

用户黑名单概述

更新时间:2024-08-30

用户黑名单服务可以限制用户之间发送单聊消息，可用于实现单聊会话场景下的拉黑功能。

默认情况下，App Key 下所有用户均已开启用户黑名单服务，可以使用以下接口：

- 添加用户到黑名单
- 从黑名单中移除用户
- 获取某用户的黑名单列表

用户黑名单服务的要点如下：

- 用户黑名单人数存在上限。详见 IM 尊享版、IM 旗舰版功能对照表 。
- 在即时通讯服务端建立用户黑名单关系的同时，需要在自己的应用服务器维护一份用户黑名单关系，以方便自己对业务数据的处理。
- 用户黑名单服务不能与用户白名单服务同时使用。开启白名单服务会导致用户黑名单服务失效。原黑名单中设置的信息不会丢失，切换回黑名单服务时，原黑名单设置仍然生效。

> 用户白名单服务也可以可以限制用户之间发送单聊消息。该服务可针对消息发件人的进行过滤，只有在发件人处于当前用户设置的白名单中时，该发件人的消息才允许发送给当前用户。详见用户白名单服务概述。

添加用户到黑名单

更新时间:2024-08-30

为指定用户的黑名单中添加一个或多个用户。被加入黑名单中的用户无法向当前用户发送单聊消息。

- 黑名单服务可限制用户之间发送消息，但加入黑名单为单向操作，例如：用户 A 拉黑用户 B，代表 B 无法给 A 发消息。但 A 向 B 发消息，B 仍然能正常接收。
- 单个用户的黑名单总人数存在上限，具体与计费套餐有关。IM 旗舰版与 IM 尊享版上限为 3000 人，其他套餐详见功能对照表 中的服务限制。
- 调用服务端 API 发送单聊消息默认不受黑名单限制。如需启用限制，请在调用相关 API 时设置 `verifyBlacklist` 为 `1`。

> 提示
>
> 在即时通讯服务端建立用户黑名单关系的同时，需要在自己的应用服务器维护一份用户黑名单关系，以方便自己对业务数据的处理。

请求方法

POST： https://数据中心域名/user/blacklist/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
blackUserId	String	是	被加入黑名单的用户 ID。单次可添加最多 20 个 `blackUserId`。单个用户的黑名单总人数存在上限，具体与计费套餐有关。IM 旗舰版与 IM 尊享版上限为 3000 人，其他套餐详见功能对照表 中的服务限制。

请求示例

```
POST /user/blacklist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&blackUserId=jlk454&blackUserId=jlk457
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移除黑名单中用户

更新时间:2024-08-30

从指定用户的黑名单中移除一个或多个用户。

请求方法

POST： https://数据中心域名/user/blacklist/remove.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
blackUserId	String	是	被移除黑名单的用户 ID，每次最多移除 20 个用户。

请求示例

```
POST /user/blacklist/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&blackUserId=jlk454
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

获取黑名单用户列表

更新时间:2024-08-30

查询指定用户的黑名单用户列表。黑名单中的用户无法向当前用户发送消息。

请求方法

POST： https://数据中心域名/user/blacklist/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。

请求示例

```
POST /user/blacklist/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	String[]	黑名单用户数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"users":["jlk454","jlk457"]}
```

用户白名单概述

更新时间:2024-08-30

用户白名单服务可以限制用户之间发送单聊消息。白名单服务一般用于满足强好友关系的业务需求，即只允许白名单中的用户向当前用户发送消息。两个用户要在单聊会话中互发消息，必须在各自的用户白名单中。

用户白名单服务开启成功后，可使用以下接口：

- 添加用户到白名单
- 移除白名单中用户
- 获取用户白名单列表

用户白名单服务要点：

- 用户白名单人数存在上限。详见 IM 尊享版、IM 旗舰版功能对照表 。
- 用户白名单服务与用户黑名单服务不能同时使用。默认情况下，App Key 下所有用户仅开启用户黑名单服务。如果需要使用开启用户白名单服务，方式如下：
  - 为指定单个或多个用户开启用户白名单服务。您可以直接调用 API 进行设置，详见开启用户白名单。**如果为指定用户开启白名单服务，该用户的黑名单服务失效。**
  - 为 App Key 下所有用户均启用单聊用户白名单，需要提交工单 。**如果为 App Key 全部用户开启白名单服务，所有用户的黑名单服务失效。**
- 在即时通讯服务端建立用户白名单关系的同时，需要应用的后端服务中维护一份用户白名单关系，以方便自己对业务数据的处理。

设置白名单服务开关

更新时间:2024-08-30

为指定单个或多个用户开启用户白名单服务。开启用户白名单服务后，黑名单功能不再生效。

开启服务后，可通过「添加白名单」接口功能为启用服务的用户设置白名单用户列表。

> 提示
>
> - 当前接口仅为指定用户启用白名单服务。如果您需要 App Key 下所有用户均启用白名单服务，请提交工单，申请开通单聊用户白名单。
> - 如果 App Key 启用了单聊用户白名单服务，无法通过当前接口将指定单个或多个用户切换为用户黑名单服务。
> - 用户白名单人数存在上限。详见 IM 尊享版、IM 旗舰版功能对照表。

请求方法

POST： https://数据中心域名/user/whitesetting/set.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String[]	是	用户 ID，最多设置 20 个。
whiteSetting	Int	是	状态，1 为开启白名单、0 为开启黑名单，默认为黑名单。

请求示例

```
POST /user/whitesetting/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=98&userId=99&whiteSetting=0
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询白名单服务状态

更新时间:2024-08-30

查询指定用户是否开启白名单过滤功能。

请求方法

POST： https://数据中心域名/user/whitesetting/query.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，最多查询 20 个。

请求示例

```
POST /user/whitesetting/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=98&userId=99
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	String[]	用户状态列表。
userId	String	用户 ID。
whiteSetting	String	状态，1 为开启白名单、0 为开启黑名单，默认为黑名单。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"users": [
{
"whiteSetting": "0",
"userId": "00"
}
]}
```

添加用户到白名单

更新时间:2024-08-30

为指定用户设置白名单用户列表。设置后该用户只能接收到自己白名单列表中用户发送的消息。使用该接口前请确认已开启白名单服务。详见白名单服务概述。

- 单个用户的白名单总人数存在上限，具体与计费套餐有关。IM 旗舰版与 IM 尊享版上限为 3000 人，其他套餐详见功能对照表中的服务限制。
- 调用服务端 API 发送单聊消息默认不受白名单限制。如需启用限制，请在调用 API 时设置 `verifyBlacklist` 为 `1`。

请求方法

POST： https://数据中心域名/user/whitelist/add.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
whiteUserId	String	是	被加入白名单的用户 ID。单次可添加最多 20 个 `whiteUserId`。单个用户的白名单总人数存在上限，具体与计费套餐有关。IM 旗舰版与 IM 尊享版上限为 3000 人，其他套餐详见功能对照表中的服务限制。

请求示例

```
POST /user/whitelist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=7&whiteUserId=123&whiteUserId=1456
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移除白名单中用户

更新时间:2024-08-30

从指定用户的白名单中移除一个或多个用户。

使用该接口前请确认已开启白名单服务。详见白名单服务概述。

请求方法

POST： https://数据中心域名/user/whitelist/remove.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
whiteUserId	String	是	被移除的用户 ID，每次最多移除 20 个用户。

请求示例

```
POST /user/whitelist/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=7&whiteUserId=123&whiteUserId=1456
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询白名单中用户列表

更新时间:2024-08-30

查询指定用户的白名单用户列表。仅白名单中的用户可向当前用户发送消息。

使用该接口前请确认已开启白名单服务。详见白名单服务概述。

请求方法

POST： https://数据中心域名/user/whitelist/query.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。

请求示例

```
POST /user/whitelist/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
users	String[]	白名单用户数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"users":["jlk454","jlk457"]}
```

消息类型概述

更新时间:2024-08-30

即时通讯服务提供丰富的预定义消息类型（或“内置消息类型），以简化即时通讯应用的开发，提高开发效率。

一个预定义消息类型（或“内置消息类型）具有唯一的类型标识（ObjectName），且消息内容必须符合预定义的内容体结构。不同消息类型在服务端和客户端具有不同的处理逻辑。

消息属性

不同消息类型在服务端和客户端具有不同的处理逻辑，具体体现为是否存储、是否计入未读数、是否支持离线消息机制、是否支持推送等几个属性。

- **客户端计数与存储策略**：计数策略是指是否影响未读消息数。如接收的消息影响会话未读数，则会话列表中会话的未读消息数 +1。存储策略是指发送、接收该消息后，客户端本地数据库是否存储该消息。
- **离线消息缓存**：即时通讯服务端在接收者不在线时，是否对单聊、群聊、系统会话中的消息进行缓存，默认最长缓存 7 天。接收人在 7 天内上线，均可接收到该消息。超过 7 天后，消息被离线缓存淘汰。如果消息类型不支持离线消息缓存，则只有接收人在线时，才可收到该消息。
- **远程推送通知**：在接收者不在线时，是否默认为该条消息触发远程推送通知。支持远程推送通知的一个必要条件是该条消息设置了推送通知标题和推送通知内容。即时通讯服务为部分用户常用的消息类型预置了推送通知标题和通知内容。除非已明确声明不支持推送，其他消息类型如果在发送消息时主动设置了推送通知标题和通知内容，则可在接收者不在线时触发远程推送通知。

> 重要
>
> - 客户端的计数行为只影响会话未读数，不影响 App 应用角标显示的未读数。
> - 通过服务端 API 发送单聊、群聊、聊天室、超级群、系统消息，默认仅会存入收件人在服务端的历史消息记录，且该行为依赖对应的云端存储服务。如果云端存储服务不可用，则无法存入历史消息记录。如果需要同时在发件人在服务端的历史消息记录中存储该消息，请参考客户端如何同步已发消息。
> - 超级群业务和聊天室业务不支持离线消息缓存机制。
> - Web 端 和 小程序端因本地存储不可靠，不支持客户端消息存储。
> - 由于 Web、小程序、PC 端没有推送服务平台，无法收到推送提醒。

消息分类

为了便于理解，我们将预定义的消息类型大致分为以下几类：

- **用户内容类消息**：用户内容类消息包含了用户之间可能发送的消息内容，包含文本、图片、GIF、语音、文件、小视频、位置、引用、合并转发等类型。
- **通知类消息**：表示一个通知信息，可能需要展现在聊天界面上，如提示条通知。

- 状态类消息：表示一个状态，用来实现输入状态（提示“对方正在输入”）、单聊会话已读通知等功能。
- 信令类消息：即时通讯服务在实现自身业务功能时使用的，应用程序一般不需要对其做任何处理。

每个分类下均提供了多种预定义的消息类型。如果客户端 SDK 发送预定义的消息类型，可直接构建消息对象，设置相应类型的消息内容体并发送。如果使用服务端 API，需要指定需要发送的消息类型的类型标识（Object Name），并严格按照预定义的消息类型内容体结构要求传入消息内容。

用户内容类消息

用户内容类消息包含了用户之间可能发送的消息内容，包含文本、图片、GIF、语音、文件、小视频、位置、引用、合并转发等类型。

即时通讯服务端为用户内容类的消息类型预置了推送通知标题和通知内容。如果发送消息时未提供自定义的推送通知标题和通知内容，则默认使用预置的推送通知标题和通知内容。默认的推送通知标题与内容可参见下表中各消息类型文档。

消息类型	ObjectName	客户端计数与存储策略	离线消息缓存	远程推送通知
文本消息	RC:TxtMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
高清语音消息	RC:HQVCMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
图片消息	RC:ImgMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
GIF 图片消息	RC:GIFMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
图文消息	RC:ImgTextMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
文件消息	RC:FileMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
位置消息	RC:LBSMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
小视频消息	RC:SightMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
引用消息	RC:ReferenceMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送
合并转发消息	RC:CombineMsg	ISCOUNTED：在客户端存储，且计入未读数	支持	默认已支持推送

通知类消息

通知信息一般需要展示在聊天界面上，如“张三加入了群组”。此类消息不增加未读消息计数。

即时通讯服务端没有为通知类的消息类型预置推送通知标题和通知内容。如果发送消息时未提供自定义的推送通知标题和通知内容，即使收件人不在线，默认也不会触发远程推送。如需支持推送，请传入自定义的推送通知内容。

消息类型	ObjectName	客户端计数与存储策略	离线消息缓存	远程推送通知
撤回通知消息	RC:RCNtf	ISPERSISTED：在客户端存储，不计入未读数	支持	默认未支持推送
联系人(好友)通知消息	RC:ContactNtf	ISPERSISTED：在客户端存储，不计入未读数	支持	默认未支持推送
资料通知消息	RC:ProfileNtf	ISPERSISTED：在客户端存储，不计入未读数	支持	默认未支持推送
提示条通知消息	RC:InfoNtf	ISPERSISTED：在客户端存储，不计入未读数	支持	默认未支持推送

消息类型	ObjectName	客户端计数与存储策略	离线消息缓存	远程推送通知
群组通知消息	RC:GrpNtf	ISPERSISTED：在客户端存储，不计入未读数	支持	默认未支持推送
命令提醒消息	RC:CmdNtf	ISPERSISTED：在客户端存储，不计入未读数	支持	默认未支持推送

状态类消息及属性

表示的是即时的状态，例如输入状态，仅当用户在线时可以接收。因为状态消息在客户端与服务端均不会存储，如果接收方不在线，则无法再收到该状态消息。如果使用即时通讯服务端 API 发送，请使用发送状态消息接口：发送单聊状态消息、发送群聊状态消息。

消息类型	ObjectName	客户端计数与存储策略	支持离线消息缓存	远程推送通知
正在输入状态消息	RC:TypSts	STATUS：在客户端不存储，不计入未读数	不支持	不支持推送

信令类消息

> 提示
>
> 即时通讯服务在实现 SDK 自身业务功能时使用。一般由 SDK 内部或即时通讯服务端发送，开发者不需要对其做任何处理。

一般为需要确保收到，但不需要展示的消息，例如运营平台向终端发送的指令信息。如果消息接收方不在线，再次上线时可通过离线消息收到。全量消息路由数据中会包含这些类型的消息。消息回调服务支持配置这些消息类型。

信令类消息

即时通讯服务端为部分信令类的消息类型预置了推送通知标题和通知内容。如果发送消息时未提供自定义的推送通知标题和通知内容，则默认使用预置的推送通知标题和通知内容。默认的推送通知标题与内容可参见下表中各消息类型文档。

消息类型	ObjectName	客户端计数与存储策略	支持离线消息缓存	远程推送通知
命令消息	RC:CmdMsg	NONE：在客户端不存储，不计入未读数	支持	默认未支持推送
撤回命令消息	RC:RcCmd	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送
单聊已读回执消息	RC:ReadNtf	NONE：在客户端不存储，不计入未读数	支持	默认未支持推送
群聊已读回执请求消息	RC:RRReqMsg	NONE：在客户端不存储，不计入未读数	支持	默认未支持推送
群聊已读回执响应消息	RC:RRRspMsg	NONE：在客户端不存储，不计入未读数	支持	默认未支持推送
多端已读状态同步消息	RC:SRSMsg	NONE：在客户端不存储，不计入未读数	支持	默认未支持推送
聊天室属性通知消息	RC:chrmKVNotiMsg	NONE：在客户端不存储，不计入未读数	支持	默认未支持推送

音视频信令类消息

即时通讯服务端为音视频信令类的消息类型预置了推送通知标题和通知内容。如果发送消息时未提供自定义的推送通知标题和通知内容，则默认使用预置的推送通知标题和通知内容。

消息类型	ObjectName	客户端计数与存储策略	支持离线消息缓存	远程推送通知
实时音视频接受信令	RC:VCAccept	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送
实时音视频挂断信令	RC:VCHangup	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送
实时音视频邀请信令	RC:VCInvite	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送
实时音视频切换信令	RC:VCModifyMedia	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送
实时音视频成员变化信令	RC:VCModifyMem	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送
实时音视频响铃信令	RC:VCRinging	NONE：在客户端不存储，不计入未读数	支持	默认已支持推送

消息扩展功能消息

消息扩展功能消息 ObjectName 为 RC:MsgExMsg。在超级群消息业务中，设置、删除消息扩展信息时，可通过消息回调或全量消息路由接收到消息扩展功能类消息。客户端无法收到该类消息。

用户内容类消息格式

更新时间:2024-08-30

即时通讯服务将用户使用即时通讯业务时可能发送的消息归为一类，称为用户内容类消息。即时通讯服务根据具体消息内容，为文本、图片、图文（已不推荐使用）、GIF、语音、文件、小视频、位置、引用、合并转发等提供了预定义的消息内容结构。

文本消息

即时通讯服务定义了消息类型标识（ObjectName）为 RC:TxtMsg 的文本消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:TxtMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`）中传入当前消息类型的消息内容结构体。

文本消息内容结构体

文本消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 content 字段的值）：

```
{
"content":"@张三 Hello world!",
"mentionedInfo":{
"type":2,
"userIdList":["zhangsan"],
"mentionedContent":"有人@你"
},
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

文本消息内容结构参数如下表所示：

名称	类型	必传	说明
content	String	是	文字消息的文字内容，包括表情。

名称	类型	必传	说明
mentionedInfo	Object	否	在群组中发送 @ 消息时，需要传入被 @ 用户信息。单聊场景下无需设置此属性。使用 IM Server API 发送群聊 @ 消息时，必须设置 `isMentioned` 字段为 `1`，指定当前消息为 @消息。否则默认发送普通消息，`mentionedInfo` 数据无效。
mentionedInfo.type	int	是	@ 功能类型，1 表示 @ 所有人、2 表示 @ 指定用户。
mentionedInfo.userIdList	String[]	否	指定用户列表，type 为 2 时有效，为 1 时 userIdList 可以为空。
mentionedInfo.mentionedContent	String	否	@ 消息的自定义 Push 内容。@消息携带的 `mentionedContent` 优先级最高，会覆盖所有默认或自定义的 `pushContent` 数据。
user	String	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为消息内容。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：消息内容。可在发送消息时自定义推送内容。

图片消息

即时通讯服务定义了消息类型标识（`ObjectName`） 为 `RC:ImgMsg` 的图片消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:ImgMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

图片消息内容结构体

图片消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"content":"/9j/4AAQSkZJRgABAgAAZABkAAD",
"localPath":"",
"imageUri":"http://p1.cdn.com/fds78ruhi.jpg",
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

图片消息的消息内容 JSON 对象中包含的图片资源分为两部分：

- 缩略图：JSON 中的 `content` key 用于表示缩略图 Base64 编码后的值。
  - 从服务端发送图片消息时，您需要自行生成缩略图（建议最长边不超过 240 像素），进行 Base64 编码后放入 `content` 中。注意，部分工具图片转 Base64 输出结果会携带 Data URI 前缀。例如：``。请丢弃其中 Data URI 前缀，仅保留数据部分，例如：`/9j/4AAQSkZJRgABAgAAZABkAAD`。
  - 从客户端发送图片消息时，SDK 自动以宽度和高度中较长的边不超过 240 像素进行等比压缩生成缩略图。接收方的会话页面仅展示缩略图（最长边 240 像素）。
- 大图（或原图）：JSON 中的 `imageUri` key 用于表示图片的远程地址。
  - 从服务端发送图片消息时，您需要自行提供图片远程地址。例如，您自行上传图片文件到 App 的文件服务器，生成地址后进行发送。接收方在会话页面中点击缩略图后展示原始尺寸与比例的图片。
  - 从客户端发送图片消息时，SDK 自动上传到文件服务器（默认上传到七牛云存储），并将云存储服务返回的大图或原图的远程地址放入消息体对应字段后进行发送。如果未设置为发送原图，SDK 会以宽度和高度中较长的边不超过 960 像素等比压缩。接收方在会话页面中点击缩略图后展示大图（最长边 960 像素）。如果未设置发送原图，接收方在会话页面中点击缩略图后展示原图。

图片消息内容结构参数如下表所示：

名称	类型	必传	说明
content	String	是	图片缩略图进行 Base64 编码的结果值。Base64 字符串长度建议为 5k，最大不超过 10k。注意在 Base64 进行 Encode 后需要将所有 \r\n 和 \r 和 \n 替换成空。。
name	String	否	文件名称。如不传，调用客户端 SDK 中的 downloadMediaMessage 方法下载后会默认生成一个名称。
localPath	String	是	图片的本地地址。仅客户端使用此字段，服务端不传入。
imageUri	String	是	图片上传到图片存储服务器后的地址。通过 IM Server API 发送图片消息时，需要自行上传文件到应用的文件服务器，生成图片地址后进行发送。
user	String	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。

名称	类型	必传	说明
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[图片]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[图片]。可在发送消息时自定义推送内容。

GIF 图片消息

即时通讯服务定义了消息类型标识（ObjectName）为 RC:GIFMsg 的图片消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:GIFMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

GIF 消息内容结构体

GIF 图片消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 content 字段的值）：

```
{
"gifDataSize":34563,
"height":246,
"localPath":"/var/mobile/.../GIF_53",
"remoteUrl":"https://rongcloud-image.cn.ronghub.com/image_jpe64562665566.gif",
"width":263,
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

GIF 图片消息内容结构参数如下表所示：

名称	类型	必传	说明
gifDataSize	Int	是	GIF 图片文件大小，单位为字节 Byte。
name	String	否	文件名称。如不传，调用客户端 SDK 中的 `downloadMediaMessage` 方法下载后会默认生成一个名称。
localPath	String	是	GIF 图片的本地地址。仅客户端使用此参数，服务端不传入。
remoteUrl	String	是	GIF 图片的服务器地址。通过 IM Server API 发送 GIF 图片消息时，需要自行上传文件到应用的文件服务器，生成 GIF 图片地址后进行发送。
width	Int	是	GIF 图片宽度。
height	Int	是	GIF 图片高度。
user	String	是	消息中携带的用户信息，IMKit SDK 会话界面中优先显示消息中携带的用户信息，可去掉此属性。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[图片]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[图片]。可在发送消息时自定义推送内容。

高清语音消息

即时通讯服务定义了消息类型标识（`ObjectName`）为 `RC:HQVCMsg` 的高清语音消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:HQVCMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

高清语音消息内容结构体

高清语音消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON** 对象序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"localPath":"/9j/4AAQSkZ/2wBaSiimB//9k=",
"remoteUrl":"http://p1.cdn.com/fds78ruhi.aac",
"duration":7,
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

高清语音消息内容结构参数如下表所示：

参数	类型	必传	说明
name	String	否	文件名称。如不传，调用客户端 SDK 中的 downloadMediaMessage 方法下载后会默认生成一个名称。
localPath	String	是	采用 AAC 格式进行编码录制的媒体内容本地路径。仅客户端使用此参数，服务端不传入。
remoteUrl	String	是	媒体内容上传服务器后的网络地址。通过 IM 服务端 API 发送高质量语音消息时，需要自行生成 AAC 格式文件并上传文件到应用的文件服务器，生成地址后进行发送。
duration	Int	是	语音消息的时长，最长为 60 秒（单位：秒）。
user	String	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[语音]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[语音]。可在发送消息时自定义推送内容。

> 提示
>
> 关于旧版本语音消息 RC:VcMsg 的说明：
>
> - ObjectName `RC:VcMsg` 的消息为旧版语音消息。旧版语音消息中直接发送 BASE64 编码后的语音数据。

- 推荐使用高清语音消息 `RC:HQVCMsg`。高清语音消息可将录制的音频数据存储到服务端，而消息体内只保存 URL。摆脱了消息体 128K 的大小限制，所以拥有更高音质。语音时长上限为 60 秒。

文件消息

即时通讯服务定义了消息类型标识（ObjectName） 为 RC:FileMsg 的文件消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:HQVCMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

文件消息内容结构体

文件消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"name":"file.txt",
"size":190184,
"type":"txt",
"localPath":"",
"fileUrl":"http://www.demo.com/am.ind",
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

文件消息内容结构参数如下表所示：

名称	类型	必传	说明
name	String	否	文件名称。如不传，调用客户端 SDK 中的 `downloadMediaMessage` 方法下载后会默认生成一个名称。
size	String	是	文件大小，单位：Byte。
type	String	是	文件类型。
localPath	String	是	文件的本地地址。仅客户端使用此参数，服务端不传入。
fileUrl	String	是	文件的服务器地址。通过 IM Server API 发送文件消息时，需要自行上传文件到应用的文件服务器，生成文件地址后进行发送。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。

名称	类型	必传	说明
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为：[文件] + 文件名，如："[文件] 123.txt"。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[文件] + 文件名。可在发送消息时自定义推送内容。

小视频消息

即时通讯服务定义了消息类型标识（ObjectName）为 RC:SightMsg 的小视频消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:SightMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

小视频消息内容结构体

小视频消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 content 字段的值）：

```
{
"sightUrl":"http://rongcloud...com/video...",
"content":"/9j/4AAQSkZ/2wB...hDSaSiimB//9k=",
"duration":2,
"size":734320,
"name":"video_xx.mp4",
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":"extra"
}
```

小视频消息**消息内容 JSON 对象**中的 content key 用于表示缩略图 Base64 编码后的值。

- 从服务端发送小视频消息时，您需要自行生成缩略图（建议最长边不超过 240 像素），进行 Base64 编码后放入 `content`

中。注意，部分工具图片转 Base64 输出结果会携带 Data URI 前缀。例如：``。请丢弃其中 Data URI 前缀，仅保留数据部分，例如：`/9j/4AAQSkZJRgABAgAAZABkAAD`。

- 从客户端发送小视频消息时，SDK 自动以宽度和高度中较长的边不超过 240 像素进行等比压缩生成缩略图。接收方的会话页面仅展示缩略图（最长边 240 像素）。

小视频消息内容结构参数如下表所示：

名称	类型	必传	说明
sightUrl	String	是	上传到文件服务器的小视频地址。通过 Server API 发送视频消息时，需要自行上传视频文件到应用的文件服务器，生成文件地址后进行发送。
content	String	是	小视频首帧的图片缩略图进行 Base64 编码的结果值。Base64 字符串长度建议为 5k，最大不超过 10k。注意在 Base64 进行 Encode 后需要将所有 `\r\n` 和 `\r` 和 `\n` 替换成空。
duration	Int	是	视频时长，单位：秒。 • 如使用 IMKit 小视频插件进行录制，默认最长可录制 10 秒。 • 如选择本地视频文件，请注意服务端的默认视频时长上限为 2 分钟。如需调整上限，请联系商务。
size	String	是	视频大小单位 Byte。
name	String	是	发送端视频的文件名，小视频文件格式为 MP4。IMKit SDK 中目前支持播放的视频文件格式为 mp4（H.264+AAC），IMLib SDK 中播放功能需要开发者自行实现。
user	String	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- 在客户端本地存储
- 计入会话消息未读数
- 默认支持离线消息推送
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[小视频]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[小视频]。可在发送消息时自定义推送内容。

位置消息

即时通讯服务定义了消息类型标识（ObjectName）为 RC:LBSMsg 的位置消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:LBSMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

位置消息内容结构体

位置消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"content":"bhZPzJXimRwrtvc=",
"latitude":39.9139,
"longitude":116.3917,
"poi":"北京云中融信网络科技有限公司",
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

位置消息内容结构参数如下表所示：

名称	类型	必传	说明
content	String	是	表示位置图片缩略图，格式为 JPG，以 Base64 进行 Encode 后需要将所有 \r\n 和 \r 和 \n 替换成空。
latitude	Number	是	位置的纬度值。
longitude	Number	是	位置的经度值。
poi	String	是	表示位置的 poi 信息。
user	String	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

属性	描述
是否在本地存储	发送、接收该消息后，会在本地数据库存储。Web 端 和 小程序端因本地存储不可靠，不支持客户端消息存储 。
是否计入消息计数属	会话未读消息数 +1。该属性只影响会话列表未读数计数。App 应用角标可根据每个会话列表未读数累加后获得。

属性	描述
是否可接收推送通知	支持推送到客户端（Web、小程序、PC 端除外）。当有离线缓存消息时，进行远程推送提醒。
推送通知标题	默认为发送者昵称（群昵称）。可在发送消息时指定自定义推送标题。
推送通知内容	默认中文推送内容为 “[位置]“。可在发送消息时自定义推送内容。

引用消息

即时通讯服务定义了消息类型标识（ObjectName） 为 RC:ReferenceMsg 的引用消息。

> 提示
>
> 一般应用于单聊、群聊场景。对指定消息内容进行回复时可发送引用消息。目前支持被引用的消息，包括：文本消息、文件消息、图文消息、图片消息，被引用的消息可以再次被引用回复。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:ReferenceMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

引用消息内容结构体

引用消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"content":"引用消息自身内容结构体内包含的具体内容!",
"referMsgUserId":"432432",
"objName":"RC:TxtMsg",
"referMsg":{
"content":"Hello world!",
"extra":""
},
"mentionedInfo":{
"type":2,
"userIdList":["123","456"],
"mentionedContent":"有人@你"
},
"user":
{
"id":"4242",
"name":"Robin",
"portrait":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

引用消息内容结构参数如下表所示：

名称	类型	必传	说明
content	String	是	引用消息时发送的文字内容，包括表情。
referMsgUserId	String	是	被引用消息的发送用户 Id。
referMsg	String	是	被引用消息的消息结构 JSON 格式。
objName	String	是	被引用消息的消息类型，目前支持的消息类型：文本 `RC:TxtMsg`、图片 `RC:ImgMsg`、文件 `RC:FileMsg`
mentionedInfo	Object	否	在群组中发送 @ 消息时，需要传入被 @ 用户信息。单聊场景下无需设置此属性。使用 IM Server API 发送群聊 @ 消息时，必须设置 `isMentioned` 字段为 `1`，指定当前消息为 @消息。否则默认发送普通消息，`mentionedInfo` 数据无效。
mentionedInfo.type	int	是	@ 功能类型，1 表示 @ 所有人、2 表示 @ 指定用户。
mentionedInfo.userIdList	String[]	否	指定用户列表，type 为 2 时有效，为 1 时 userIdList 可以为空。
mentionedInfo.mentionedContent	String	否	@ 消息的自定义 Push 内容。@消息携带的 `mentionedContent` 优先级最高，会覆盖所有默认或自定义的 `pushContent` 数据。
user	Object	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为消息内容。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：消息内容。可在发送消息时自定义推送内容。

合并转发消息

即时通讯服务定义了消息类型标识（`ObjectName`） 为 `RC:CombineMsg` 的合并转发消息。

> 提示
>
> 客户端仅 IMKit SDK 支持发送合并转发消息。从客户端发送合并转发消息后，消息以 HTML 文件的方式存储到即时通讯服务端，对端收到消息后计入未读消息数、进行存储。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:CombineMsg`。

- 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。

合并转发消息内容结构体

合并转发消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON** 对象序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"localPath":"file:///storage/emulated/0/Android/data/cn.rongcloud.im/cache/combine/157829828.html",
"remoteUrl":"https://rongcloud-html-cn.ronghub.com/text_plain__RC-2019-12-17_754_157130.html?e=1592Q=",
"conversationType":1,
"nameList":["lisx","ddddd"],
"summaryList":["lisx : nzj","ddddd : 记得记得晋级赛","ddddd : 就是就是睡觉觉","lisx : nznznn"]
}
```

合并转发消息内容结构参数如下表所示：

名称	类型	必传	说明
localPath	String	否	合并转发消息详细信息的本地 HTML 文件路径。仅客户端使用此字段，服务端不传入。
remoteUrl	String	是	远端 HTML 文件路径。
conversationType	Int	是	会话类型，目前支持二人会话及群聊会话，二人会话是 1 、群组会话是 3 。
nameList	String	是	在会话界面显示的合并转发消息中，前 4 条消息的用户名称。
summaryList	String	是	在会话界面显示的合并转发消息中，前 4 条消息的简略信息，与 nameList 属性相对应。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[聊天记录]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[聊天记录]。可在发送消息时自定义推送内容。

图文消息

即时通讯服务定义了消息类型标识（`ObjectName`） 为 `RC:ImgTextMsg` 的图文消息。

- 客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。
- 如果使用服务端 API 发送该类型消息，注意传入正确的正确的消息类型标识以及消息内容结构体：
  - 必须在 API 接口的消息类型字段（字段名一般为 `ObjectName`）传入消息类型标识 `RC:ImgTextMsg`。
  - 必须在 API 接口的消息内容字段（字段名一般为 `content`） 中传入当前消息类型的消息内容结构体。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

图文消息内容结构体

图文消息的消息内容 JSON 对象结构如下（将下方的**消息内容 JSON 对象**序列化为 JSON String，作为 API 接口的 `content` 字段的值）：

```
{
"title":"标题",
"content":"消息描述",
"imageUri":"http://p1.cdn.com/fds78ruhi.jpg",
"url":"http://www.rongcloud.cn",
"user":
{
"id":"4242",
"name":"Robin",
"icon":"http://example.com/p1.png",
"extra":"extra"
},
"extra":""
}
```

图文消息内容结构参数如下表所示：

名称	类型	必传	说明
title	String	是	消息的标题。
content	String	是	消息的文字内容。
imageUri	String	是	消息中图片地址，图片尺寸为：120 x 120 像素。通过 IM Server API 发送图片消息时，需要自行上传文件到应用的文件服务器，生成图片地址后进行发送。
url	String	是	点击图片消息后跳转的 URL 地址。
user	String	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
user.id	String	否	消息发送者的用户 ID。
user.name	String	否	消息发送者的用户昵称。
user.portrait	String	否	消息发送者的头象。
user.extra	String	否	扩展信息，可以放置任意的数据内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **计入会话消息未读数**
- **默认支持离线消息推送**
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[图文]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[图文]。可在发送消息时自定义推送内容。

如何发送 @ 消息

@消息并非预定义的消息类型，不具有指定的 ObjectName。

即时通讯服务支持在向群组、超级群中发送以下类型的消息时，在消息内容中添加 mentionedInfo 对象，实现提及他人（@）的效果。

- 文本消息（ObjectName 为 `RC:TxtMsg`）
- 引用消息（ObjectName 为 `RC:ReferenceMsg`）

@ 消息的消息内容内部需要携带 mentionedInfo 对象，结构如下：

名称	类型	必传	说明
mentionedInfo	Object	否	在群组中发送 @ 消息时，需要传入被 @ 用户信息。单聊场景下无需设置此属性。使用 IM Server API 发送群聊 @ 消息时，必须设置 isMentioned 字段为 1，指定当前消息为 @消息。否则默认发送普通消息，mentionedInfo 数据无效。
mentionedInfo.type	int	是	@ 范围。1 表示 @ 所有人。2 表示需要 @ 指定用户，用户列表由 mentionedInfo.userIdList 指定。
mentionedInfo.userIdList	String[]	否	指定用户列表，type 为 2 时有效，为 1 时 userIdList 可以为空。
mentionedInfo.mentionedContent	String	否	@ 消息的自定义 Push 内容。@消息携带的 mentionedContent 优先级最高，会覆盖所有默认或自定义的 pushContent 数据。

> 提示
>
> 如果通过 IM Server API 发送 @消息，必须同时设置 isMentioned 为 1，指定当前消息为 @ 消息。

携带 @ 信息的文本消息内容结构

往群组、超级群发送文本消息时，在消息内容中携带 mentionedInfo 属性可发送 @ 消息。以下示例构造了一条文本内容，提及了用户 ID 为「123」和「456」的用户。@消息携带的 mentionedContent 优先级最高，会覆盖所有默认或自定义的 pushContent 数据。

```
{
"content":"@张三 @李四 Hello World!",
"mentionedInfo":{
"type":2,
"userIdList":["123","456"],
"mentionedContent":"有人@你"
},
"extra":""
}
```

注意，从服务端发送 @消息时，需要手动在消息内容 **JSON** 对象内部 content 字段中填入对应 ID 的用户昵称，在本例中为用户 ID 为「123」和「456」对应的昵称为「张三」「李四」。

以下为 JSON 结构图。具体字段说明参见文本消息内容结构。

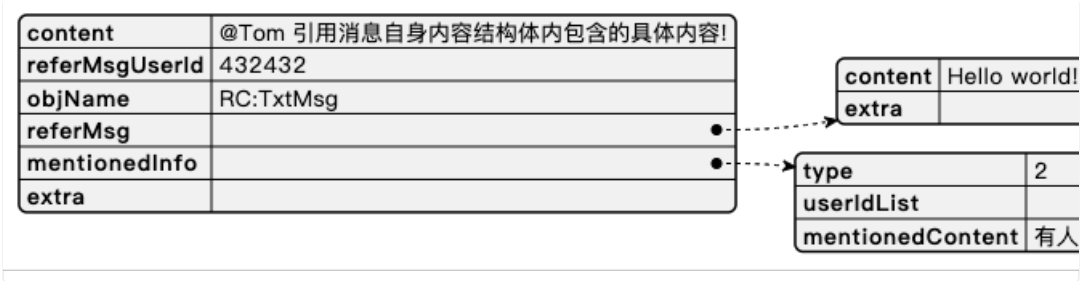
携带 @ 信息的引用消息内容结构

往群组、超级群发送引用消息时，在消息内容中携带 mentionedInfo 属性可发送 @ 消息。@消息携带的 mentionedContent 优先级最高，会覆盖所有默认或自定义的 pushContent 数据。

```
{
"content":"@Tom 引用消息自身内容结构体内包含的具体内容!",
"referMsgUserId":"432432",
"objName":"RC:TxtMsg",
"referMsg":{
"content":"Hello world!",
"extra":""
},
"mentionedInfo":{
"type":2,
"userIdList":["tom1999"],
"mentionedContent":"有人@你"
},
"extra":""
}
```

注意，从服务端发送 @消息时，需要手动在消息内容 JSON 对象内部 content 字段中填入对应 ID 的用户昵称，在本例中为用户 ID 为「tom1999」，对应的昵称为「Tom」。

以下为 JSON 结构图。具体字段说明参见文本消息内容结构。



通知类消息格式　　提示（小灰条）通知消息

更新时间:2024-08-30

即时通讯服务定义了 ObjectName 为 RC:InfoNtf 的提示（小灰条）通知消息。您可以需要通知、提示用户时，向会话中发送或本地插入该类型的消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 content 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

提示小灰条消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"message":"请在聊天中注意人身财产安全",
"extra":""
}
```

内容结构参数

提示小灰条消息内容结构参数如下表所示：

名称	类型	必传	说明
message	String	是	提示条消息内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **不计入会话消息未读数**
- **默认未支持远程推送**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

撤回通知消息

即时通讯服务定义了 ObjectName 为 RC:RcNtf 的撤回通知消息，用于提示消息已撤回。

客户端 SDK 中已内置该消息类型。SDK 内部用于在执行撤回消息操作后展示撤回通知。不建议客户直接发送该消息。

内容结构参数

撤回通知消息内容结构参数如下表所示：

名称	类型	必传	说明
operatorId	String	否	发起撤回消息的用户 ID。
recallTime	Long	是	被撤回的原始消息的发送时间（毫秒）。
originalObjectName	String	是	原消息的消息类型名。
originalMessageContent	Object	否	原消息的内容。
recallContent	String	否	撤回的文本消息的内容。
recallActionTime	Long	否	发送撤回命令消息（`RC:RcCMD`）的时间（毫秒）。
admin	Boolean	是	是否由管理员操作。
delete	Boolean	是	指定移动端接收方是否需要从本地删除原始消息记录。为 `false` 时，移动端不会删除原始消息记录，会将消息内容替换为撤回提示（小灰条通知）。为 `true` 时，移动端会删除原始消息记录，不显示撤回提示（小灰条通知）。Web SDK 从 5.3.1 版本开始支持该参数。
user	Object	否	消息中携带的消息发送者的用户信息。一般情况下不建议在消息中携带用户信息。建议仅在直播场景下使用。
extra	String	否	附加信息。

客户端默认属性

- **在客户端本地存储**
- **不计入会话消息未读数**
- **默认未支持远程推送**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

资料变更通知消息

即时通讯服务定义了 ObjectName 为 `RC:ProfileNtf` 的资料变更通知消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

资料变更通知消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"operation":"Update",
"data":"{\"nickname\":\"韩梅梅\", \"hometown\":\"beijing\"}",
"extra":""
}
```

内容结构参数

资料变更通知消息内容结构参数如下表所示：

名称	类型	必传	说明
operation	String	是	资料通知操作，可以自行定义。
data	String	是	操作的数据。

名称	类型	必传	说明
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- 在客户端本地存储
- 不计入会话消息未读数
- 默认未支持远程推送：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

联系人（好友）通知消息

即时通讯服务定义了 ObjectName 为 `RC:ContactNtf` 的联系人（好友）通知消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

联系人（好友）通知消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"operation":"Request",
"sourceUserId":"123",
"targetUserId":"456",
"message":"我是小艾，能加一下好友吗？",
"extra":""
}
```

内容结构参数

联系人（好友）通知消息内容结构参数如下表所示：

名称	类型	必传	说明
operation	String	是	联系人操作的指令，官方针对 `operation` 属性定义了 "Request", "AcceptResponse", "RejectResponse" 几个常量，也可以由开发者自行扩展。
sourceUserId	String	是	发出通知的用户 Id。
targetUserId	String	是	单聊会话为接收通知的用户 Id，群聊、聊天室会话为会话 Id。
message	String	是	表示请求或者响应消息，如添加理由或拒绝理由。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- 在客户端本地存储
- 不计入会话消息未读数
- 默认未支持远程推送：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

群组通知消息

即时通讯服务定义了 ObjectName 为 RC:GrpNtf 的群组通知消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 content 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

群组通知消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"operatorUserId":"4324",
"operation":"Rename",
"data":"群组中各种通知的操作数据",
"message":"修改本群名为本地生活",
"extra":""
}
```

内容结构参数

群组通知消息内容结构参数如下表所示：

名称	类型	必传	说明
operatorUserId	String	是	操作人用户 Id。
operation	String	是	群组中各种通知的操作名称。IMKit 客户端默认仅支持在会话列表与会话页面中展示部分内置群组通知消息。详见下方**群组通知消息结构数据说明**。
data	String	是	操作数据，IMKit 客户端在会话列表与会话页面中展示消息时会使用该字段。详见下方**群组通知消息结构数据说明**。
message	String	是	消息内容。
extra	String	否	扩展信息，可以放置任意的数据内容，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **不计入会话消息未读数**
- **默认未支持远程推送**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

群组通知消息结构数据说明

以下列出了内置的群组操作通知消息结构 JSON 格式：

> 提示
>
> Android/iOS IMKit SDK 对内置群组通知消息结构的支持与展示可能存在差异，具体请参考源码。

- 创建群组（Operation: Create）

```
{
"operatorUserId":"4324",
"operation":"Create",
"data":"{\"operatorNickname\":\"李天\",\"targetGroupName\":\"群名\"}",
"message":"创建群组",
"extra":""
}
```


属性说明：

- `operatorUserId`：为操作人用户 Id
- `operation`：操作名，默认为 Create
- `data`：数据内容，`operatorNickname` 为操作者，`targetGroupName` 为群名称

- 修改群名称（Operation: Rename）


```
{
"operatorUserId":"4324",
"operation":"Rename",
"data":"{\"operatorNickname\":\"李天\",\"targetGroupName\":\"群名\"}",
"message":"修改群名称",
"extra":""
}
```


属性说明：

- `operatorUserId`：操作人用户 Id
- `operation`：操作名，默认为 Rename
- `data`：数据内容，operatorNickname 为操作者，targetGroupName 为群名称

- 添加群成员（Operation: Add）


```
{
"operatorUserId":"4324",
"operation":"Add",
"data":"{\"operatorNickname\":\"李天\",\"targetUserIds\":[\"wGPkc0\"],\"targetUserDisplayNames\":[\"腾飞\"]}",
"message":"添加群成员",
"extra":""
}
```


属性说明：

- `operatorUserId`：操作人用户 Id

- `operation`：操作名，默认为 Add
- `data`：数据内容，operatorNickname 为操作者
- `targetUserIds`：被加入群的用户 ID
- `targetUserDisplayNames`：被加入群的用户名，与 `targetUserIds` 相对应

- 移出群成员


```
{
"operatorUserId":"4324",
"operation":"Kicked",
"data":"{\"operatorNickname\":\"李天\",\"targetUserIds\":[\"wGPkc0\"],\"targetUserDisplayNames\":[\"腾飞\"]}",
"message":"移出群成员",
"extra":""
}
```


属性说明：

- `operatorUserId`：操作人用户 Id
- `operation`：操作名，默认为 Kicked
- `data`：数据内容，operatorNickname 为操作者
- `targetUserIds`： 被移出群的用户 ID
- `targetUserDisplayNames`： 被移出群的用户名，与 `targetUserIds` 相对应

- 退出群组（Operation: Quit）


```
{
"operatorUserId":"4324",
"operation":"Quit",
"data":"{\"operatorNickname\":\"李天\",\"targetUserIds\":[\"wGPkc0VpO\"],\"targetUserDisplayNames\":[\"腾飞\"],\"newCreatorId\":\"newCreatorId\"}",
"message":"退出群组",
"extra":""
}
```


属性说明：

- `operatorUserId`：操作人用户 Id
- `operation`：操作名，默认为 Quit
- `data`：数据内容，`operatorNickname` 为操作者，`targetUserIds` 为操作者 ID ，`targetUserDisplayNames` 为操作者
- `newCreatorId`：如退出群的用户为群创建者，则 `newCreatorId` 为新的群创建者 ID ，否则为 null

- 解散群组（Operation: Dismiss）

```
{
"operatorUserId":"4324",
"operation":"Dismiss",
"data":"{\"operatorNickname\":\"李天\"}",
"message":"解散群组",
"extra":""
}
```

属性说明：

- `operatorUserId`：操作人用户 Id
- `operation`：操作名，默认为 Dismiss
- `data`：数据内容，`operatorNickname` 为操作者

命令提醒消息

即时通讯服务定义了 ObjectName 为 RC:CmdNtf 的命令提醒消息。与命令消息（RC:CmdMsg）的区别是，此消息会进行存储并在界面上显示。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 content 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

命令提醒消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"name":"AtPerson",
"data":"{\"sourceId\":\"9527\"}"
}
```

内容结构参数

命令提醒消息内容结构参数如下表所示：

名称	类型	必传	说明
name	String	是	命令名。
data	String	否	设置命令数据，可以放置任意格式的数据内容，如 JSON，也可以去掉此属性。

客户端默认属性

- **在客户端本地存储**
- **不计入会话消息未读数**
- **默认未支持远程推送**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

状态类型消息格式

更新时间:2024-08-30

即时通讯服务将客户端默认不存储、不计入会话消息未读数的消息类型归为一类，称为状态类消息。

> 提示
>
> 请注意区分状态类消息与 IM 服务端发送状态消息 API 的作用。
>
> - 状态类消息特点：客户端默认不存储、不计入会话未读消息数。可由客户端发送（一般由 SDK 内部发送），或通过 IM Server API 发送。
> - 发送状态消息的 IM Server API：可接受任意内置消息类型，如果接收者不在线，则无法收到消息。

正在输入状态消息

即时通讯服务定义了 ObjectName 为 `RC:TypSts` 的正在输入状态消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置 **正在输入状态消息** 消息类型。客户端收到消息后不计入未读消息数、不存储。

- 如果在客户端 SDK 配置中启用了单聊输入状态功能，SDK 内部会自动往单聊会话中发送该类型消息。群聊场景下不支持由 SDK 自动发送正在输入状态消息。
- 如果使用即时通讯服务端 API 发送，请使用发送状态消息接口：发送单聊状态消息、发送群聊状态消息。

正在输入状态消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"typingContentType":"RC:TxtMsg"
}
```

内容结构参数

正在输入状态消息内容结构参数如下表所示：

名称	类型	必传	说明
typingContentType	String	是	正在输入消息类型。

客户端默认属性

- **不会在客户端本地存储**
- **不计入会话消息未读数**

- **不支持离线消息推送**

信令类消息格式

更新时间:2024-08-30

即时通讯服务预定义了信令消息，仅用于 SDK 在实现自身业务功能时使用。开发者不需要对其做任何处理。

命令消息

即时通讯服务定义了 ObjectName 为 RC:CmdMsg 的命令消息。与命令提醒消息（RC:CmdNtf）的区别是命令消息不存储，也不会在界面上显示。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 content 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

运营平台向终端发送指令信息时可使用此命令消息。

命令消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"name":"AtPerson",
"data":"{\"sourceId\":\"9527\"}"
}
```

内容结构参数

命令消息内容结构参数如下表所示：

名称	类型	必传	说明
name	String	是	命令名称，可以自行定义。
data	String	是	命令的内容。

客户端默认属性

- **不会在客户端本地存储**
- **不计入会话消息未读数**
- **默认未支持远程推送通知**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

撤回命令消息

即时通讯服务定义了 ObjectName 为 RC:RcCmd 的撤回命令消息。

客户端 SDK 中已内置该消息类型。调用客户端撤回消息的 API 时，SDK 内部发送该消息。不建议客户直接发送该消息。

内容结构参数

命令消息内容结构参数如下表所示：

名称	类型	必传	说明
MessageUId	String	是	消息 UID，为服务端生成的全局唯一 ID。
TargetId	String	是	会话 ID。
ChannelId	String	是	超级群频道 ID。
SentTime	String	是	消息发送时间。
ConversationType	String	是	会话类型。
isAdmin	boolean	是	是否是管理员操作。
isDelete	boolean	是	指定移动端接收方是否需要从本地删除原始消息记录。为 `false` 时，移动端不会删除原始消息记录，会将消息内容替换为撤回提示（小灰条通知）。为 `true` 时，移动端会删除原始消息记录，不显示撤回提示（小灰条通知）。Web 端撤回消息接口从 SDK 5.3.1 版本开始支持该参数。

客户端默认属性

- 不会在客户端本地存储
- 不计入会话消息未读数
- 默认已支持远程消息推送：
  - 推送通知标题：单聊与系统会话消息的推送标题默认为发送者昵称。群组与超级群的推送标题默认为群名称。默认使用注册用户或创建群组时传入即时通讯服务端的数据。您可以在发送消息时自行指定推送标题。发消息时指定的推送标题具有最高优先级。
  - 推送通知内容：默认中文推送内容为固定字符串：[撤回了一条消息]。如果为群聊或超级群消息，会带上发送者昵称前缀，例如：发送者昵称：[撤回了一条消息]。可在撤回消息时自定义推送内容。

已读通知消息

即时通讯服务定义了 ObjectName 为 `RC:ReadNtf` 的已读通知消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

用来发送消息已经被接收到的状态消息，客户端收到消息后不计入未读消息数、不存储，此类型消息没有 Push 通知。

已读通知消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"lastMessageSendTime":1408706337,
"messageUId":"XXXXXX",
"type":1
}
```

内容结构参数

已读通知消息内容结构参数如下表所示：

名称	类型	必传	说明
lastMessageSendTime	Int	是	已读的最后一条消息的发送时间。
messageUId	String	是	已读的最后一条消息的 UId，消息唯一标识。详见表格下方的 `messageUId` 说明。
type	Int	是	会话类型，目前该消息只支持单聊会话，类型为 1。

`messageUId` 说明

- 从 Web 端发出的已读通知消息默认包含 `messageUId` 字段。
- 从 Android / iOS 等移动端发出的已读通知消息默认不含 `messageUId` 字段。

客户端默认属性

- 不会在客户端本地存储
- 不计入会话消息未读数
- 默认未支持远程推送通知：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

群聊已读回执请求消息

即时通讯服务定义了 ObjectName 为 `RC:RRReqMsg` 的群聊已读回执请求消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

群组中发送一条消息，需要获取此条消息的已读回执时，可向群组中发送已读请求消息。用户收到此条消息后，当指定消息已读后，会向消息发送用户发送已读状态消息。

群聊已读回执请求消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"messageUId":"596E-P5PG-4FS2-7OJK"
}
```

内容结构参数

群聊已读回执请求消息内容结构参数如下表所示：

名称	类型	必传	说明
messageUId	String	是	需要请求已读回执消息的消息 ID

客户端默认属性

- 不会在客户端本地存储
- 不计入会话消息未读数

- **默认未支持远程推送通知**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

群聊已读回执响应消息

即时通讯服务定义了 ObjectName 为 `RC:RRRspMsg` 的群聊已读回执响应消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

用来发送群组中已读的消息状态，客户端收到消息后不计入未读消息数、不存储。

群聊已读回执响应消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"receiptMessageDic":{"wX7zFv8dR":["BJN3-LSG0-7MUC-OR7A"]}
}
```

内容结构参数

群聊已读回执响应消息内容结构参数如下表所示：

名称	类型	必传	说明
receiptMessageDic	String	是	需要回执的消息的字典，Key 为发送者 userId ，Value 是该用户需要回执的消息的 MessageUId 的数组。

客户端默认属性

- **不会在客户端本地存储**
- **不计入会话消息未读数**
- **默认未支持远程推送通知**：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

多端已读状态同步消息

即时通讯服务定义了 ObjectName 为 `RC:SRSMsg` 的多端已读状态同步消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

用户多端时，一端消息已读后，同步到另一端的已读通知消息。

多端已读状态同步消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"lastMessageSendTime":1610941689734
}
```

内容结构参数

多端已读状态同步消息内容结构参数如下表所示：

名称	类型	必传	说明
lastMessageSendTime	Int	是	已读的最后一条消息的发送时间

客户端默认属性

- 不会在客户端本地存储
- 不计入会话消息未读数
- 默认未支持远程推送通知：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

聊天室属性通知消息

即时通讯服务定义了 ObjectName 为 `RC:chrmKVNotiMsg` 的聊天室属性通知消息。

调用服务端 API 发送消息时，必须指定 ObjectName，并将对应类型的消息内容 **JSON** 对象序列化为 JSON String，放入 API 接口的 `content` 字段中。客户端 SDK 中已内置该消息类型，可直接调用相关方法发送。

聊天室属性通知消息包含的「消息内容」为 JSON 对象，结构如下：

```
{
"type":1,
"key":"name",
"value":"主播",
"extra":""
}
```

内容结构参数

聊天室属性通知消息内容结构参数如下表所示：

名称	类型	必传	说明
type	Int	是	聊天室中对属性操作后发送通知的类型，1 为设置属性内容、2 为删除属性内容。
key	String	是	聊天室中属性名称，大小不超过 128 个字符。
value	String	是	属性对应的内容，大小不超过 4096 个字符。
extra	String	否	通过消息中携带的附加信息，对应到设置属性接口中的 notificationExtra 值。

客户端默认属性

- 在客户端本地存储
- 不计入会话消息未读数
- 默认未支持远程推送通知：即时通讯服务端没有为该消息类型预置推送通知标题和通知内容。如果需要在收件人不在线触发远程推送通知，请在发送消息时传入自定义的推送通知内容。

消息扩展功能消息

> 提示
>
> **消息扩展功能消息仅在超级群业务中使用。客户端无法收到该类消息。**

消息扩展功能消息 ObjectName 为 `RC:MsgExMsg`。

在超级群消息业务中，设置、删除消息扩展信息时，可通过消息回调或全量消息路由接收到消息扩展功能类消息。客户端无法收到该类消息。

设置消息扩展，消息结构如下：

```
{
"clear":0,"put":{"123":"示例文本","2":"222","3":"示例文本"}
}
```

删除消息扩展，消息结构如下：

```
{
"clear":0,"del":{"123":"示例文本","2":"222","3":"示例文本"}
}
```

结构说明：

字段	描述
clear	是否删除全部属性值，1 为删除全部、0 为不删除全部
del	删除指定属性 Key/Value 数组
put	设置消息属性 Key/Value 数组

实时音视频信令

使用音视频通话 SDK（CallLib / CallKit）由客户端 SDK 内部处理的消息。全量消息路由数据中会包含这些类型的消息。消息回调服务支持配置这些消息类型。

音视频接受信令消息

音视频接受信令消息 ObjectName 为 `RC:VCAccept`。

「消息内容」为 JSON 对象，结构如下：

```
{
"callId":"DBAA6F6A-21DC-4A2E-B353-2EEED958B0FC",
"mediaType":2,
"deviceId":"NxM4MmRlMDk3ZWU0ZWM3Yw",
"platform":"iOS"
}
```

音视频挂断信令消息

音视频挂断信令消息 ObjectName 为 RC:VCHangup。

「消息内容」为 JSON 对象，结构如下：

```
{
"callId":"D80A1E4B-F3B2-4B71-86F9-C6B9FA0EEEB1",
"deviceId":"YtliMzY1MDc1ZDg4MWFlYQ",
"reason":1,
"platform":"iOS"
}
```

音视频邀请信令消息

音视频邀请信令消息 ObjectName 为 RC:VCInvite。

「消息内容」为 JSON 对象，结构如下：

```
{
"callId":"53287BC6-5805-481D-A8A2-CDEC221EEB5B",
"mediaType":1,
"deviceId":"YTliMzY2MDc1ZDg4MWFlYQ",
"observerUserIds":[

],
"platform":"iOS",
"inviteUserIds":[
"O6fKSQwOt"
],
"channelInfo":{
"Key":"RongRTCChannelKey",
"Id":"53287BC6-5805-481D-A8A2-CDEC221EEB5B"
},
"engineType":4
}
```

音视频切换信令消息

音视频切换信令消息 ObjectName 为 RC:VCModifyMedia。

```
{
"callId":"b462616b-1247-4986-b77a-7f438a50bfe6",
"mediaType":1
}
```

音视频成员变化信令消息

音视频接受信令消息 ObjectName **为** `RC:VCModifyMem`**。**

```
{
"platform":"iOS",
"callId":"DBAA6F6A-21DC-4A2E-B353-2EEED958B0FC"
}
```

消息管理服务配置　　免费基础功能

更新时间:2024-08-30

以下是控制台**免费基础功能**页面与 Server API 的消息管理功能相关的配置：

- **Server API 发送消息过滤敏感词**：控制是否对 Server API 发送的消息启用敏感词机制。
- **Server API 发送消息实时路由**：启用后，Server API 发送的消息也会进行全量消息路由。仅在已开通**全量消息路由服务**的情况下生效。
- **Server API 历史消息日志下载**：即时通讯服务端可以保存 APP 内所有会话的历史消息记录，历史消息记录以日志文件方式提供，并已经过压缩。您可以使用服务端 API 获取指定 App 的历史消息日志。注意，服务端 API 接口仅返回消息记录文件下载地址，获取地址后请您自行下载。

启用服务

访问控制台免费基础功能 页面，可启用以上功能：

IM 旗舰版/尊享版

以下是控制台 **IM 服务管理**页面与 Server API 的消息管理功能相关的配置：

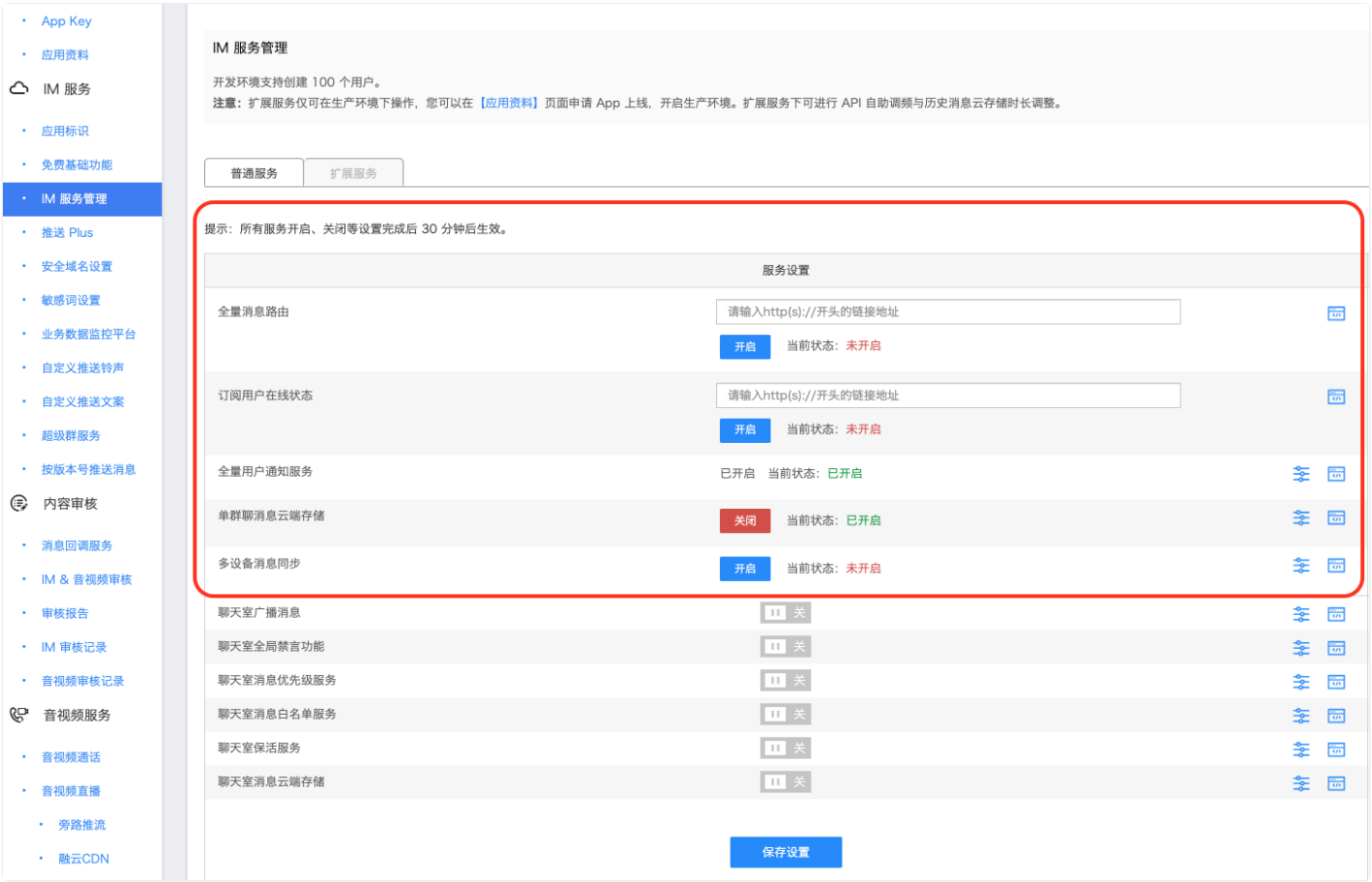
- **全量消息路由**：全量消息路由服务支持将单聊、群组、超级群、聊天室等的消息数据同步到指定的应用服务器。开通服务时，需要配置可正常访问的回调接收地址。**注意**：
  - 对超级群消息扩展启用全量消息路由，请提交工单 。
  - 对含有屏蔽敏感词的消息启用全量消息路由，请提交工单 申请开通。

- 对客户端发送的状态类消息启用全量消息路由，请提交工单申请开通单聊状态消息消息路由功能。如需需要将通过发送单聊状态消息、发送群聊状态消息接口发送的任何消息启用全量消息路由，需要开通 Server API 发送消息实时路由服务。
- 对超级群消息扩展变更启用全量消息路由，请提交工单申请调整超级群消息扩展是否进行路由。开通服务后，如果变更超级群消息扩展，服务端会生成一条类型为 `RC:MsgExMsg` 的消息。新消息仅用于消息路由同步，不会同步给客户端 SDK。新消息 ID 由服务端生成，`originalMsgUID` 字段中会携带原消息的 `MsgUID`。注意，在发消息时即携带的扩展数据无法通过全量消息路由获取。

- **单群聊消息云存储**：在即时通讯服务端存储单聊、群聊、系统会话的历史消息记录。
- **多设备消息同步**：可用于同一用户账号的多个设备之间同步收发消息。如需将 Server API 发送的消息同步给向发件人离线的客户端，建议启用。

修改服务配置

访问控制台 IM 服务管理页面，在普通服务标签下可启用以上服务。在开发环境下，可免费使用。IM 旗舰版或 IM 尊享版可开通该服务。具体功能与费用以官方价格说明页面及计费说明文档为准。



客户端如何同步已发消息

更新时间:2024-08-30

> 提示
>
> 默认行为是不向发件人的客户端同步通过 Server API 发出的消息，也不会将该已发消息存入发件人的服务端历史消息记录。

App 业务端调用 IM Server API 发送消息时，通常期待发件人用户的客户端也可同步该已发消息，例如：

- 发件人的客户端在线时，通过 Server API 发送的消息可自动同步至发件人的客户端设备。
- 发件人的离线的客户端再次上线时，可自动收取通过 Server API 发送的消息。
- 发件人的客户端拉取历史消息时，总可以获取从通过 Server API 发送的消息。

即时通讯服务端默认不向发件人的客户端同步通过 Server API 发出的消息。您可以通过以下方式实现上述需求。

设置 isIncludeSender

调用 IM Server API 的发送消息接口时，设置 `isIncludeSender` 参数为 `1`，这将保证在发件人客户端在线的情况下同步接收已发送的消息。部分接口支持 `isSyncSender` 参数，作用一致。

如果不传该参数，服务端使用默认值 `0`，即不同步，发送者的客户端无法获取该条已发消息。

以下接口支持 `isIncludeSender`：

- 发送单聊普通消息
- 发送单聊模板消息
- 发送单聊状态消息
- 发送群聊消息
- 发送群聊状态消息
- 发送聊天室消息
- 发送全体聊天室广播消息

以下接口支持 `isSyncSender`：

- 设置单群聊消息扩展
- 删除单群聊消息扩展

开通多设备消息同步

从 Server API 发送消息时，发件人的客户端可能不在线，此时即使 `isIncludeSender` 参数设为 `1`，服务端也无法向客户端

实时同步从 Server API 发出的消息。

如果需要发件人的客户端上线时自动从服务端同步已发消息，需要开通**多设备消息同步**功能。开启该功能后，Server API 发出的消息将通过离线消息补偿机制同步给客户端。客户端上线时，默认自动同步 1 个自然日（含当天）的消息。

您可以从控制台自助开通**多设备消息同步**，详见消息管理服务配置。

因聊天室、超级群业务不支持离线补偿，该方式仅支持以下单聊、群聊业务接口：

- 发送单聊普通消息
- 发送单聊模板消息
- 发送群聊消息

开通历史消息云存储服务

App 可能还需要在拉取历史消息时获取从 Server API 发出的消息。例如，从 Server API 发送消息后，发件人可能长期不登录客户端，导致离线补偿中存储的已发消息过期。或者由于发件人在客户端误操作，在本地删除了 从 Server API 发出的消息。

为满足上述需求，建议 App 同时开启历史消息云存储服务。

- **单群聊历史消息云端存储**：开通后，从 Server API 发出的单聊、群聊、系统会话消息可存入发件人的服务端历史消息记录，默认保存 6 个月。注意，必须在发送消息时将 `isIncludeSender` 参数设为 `1` 才会存储。
- **聊天室历史消息云端存储**：开通后，从 Server API 发出的聊天室消息可存入发件人的服务端历史消息记录，默认保存 2 个月。注意，必须在发送消息时将 `isIncludeSender` 参数设为 `1` 才会存储。

您可以从控制台自助开通以上服务，详见消息管理服务配置。

- 发送单聊普通消息
- 发送单聊模板消息
- 发送群聊消息
- 发送聊天室消息

例外情况

即时通讯服务不支持将聊天室广播消息存入服务端聊天室历史消息记录。因此仅支持发件人已登陆客户端（在线）的情况下同步聊天室广播消息。

超级群业务开通后，默认开启历史消息存储，无需额外开通，默认存储 7 天内的历史消息。发送超级群消息接口不支持设置 `isIncludeSender`。发件人拉取历史消息时会包含本人通过 Server API 发出的消息。

发送单聊普通消息

更新时间:2024-08-30

应用下的用户可向其他用户发送单聊消息。

- 单次支持向单个或多个用户发送单聊消息，单次最多向 1000 个用户发送消息。
- 通过该接口发送的消息，默认不会向消息发件人客户端同步，也不会存入发件用户的历史消息记录。如需同步，请参见 `isIncludeSender` 参数用法。

请求方法

POST： https://数据中心域名/message/private/publish.json

频率限制： 每分钟限发送 6000 条信息。请注意，如果一次发送给 1000 人，视为 1000 条消息。另见已知问题 1。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
toUserId	String	是	接收用户 ID，可以实现向多人发送消息，每次上限为 1000 人。
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 `{"content":"Hello world!"}` 序列化后的结果作为此处 `content` 字段的值。 • 自定义消息类型（`objectName` 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。

参数	类型	必传	说明
pushContent	String	否	指定收件人离线时触发的远程推送通知中的通知内容。**注意**：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。 • 如果消息类型（`objectName` 字段）为即时通讯服务预定义消息类型中的用户内容类消息格式，可不填写该字段，远程推送通知默认使用服务端预置的推送通知内容。 • 如果消息类型（`objectName` 字段）为即时通讯服务预定义消息类型中通知类、信令类（"撤回命令消息" 除外），且需要支持远程推送通知，则必须填写 `pushContent`，否则收件人不在线时无法收到远程推送通知。如无需触发远程推送，可不填该字段。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 `pushContent` 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 `pushContent` 字段留空禁用远程推送通知。
pushData	String	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 `appData`（**提示**：`rc` 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 `appData`。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。`1` 表示同步，默认值为 `0`，即不同步。**注意**，仅设置该参数无法确保发件人客户端一定能获取到该条已发消息，您可能还需要启用其他服务。详见发件人客户端如何同步已发消息。
count	Int	否	仅目标用户为 iOS 设备有效，Push 时用来控制桌面角标未读消息数，只有在 toUserId 为一个用户 ID 时有效，客户端获取远程推送内容时为 `badge`。具体参见 iOS 客户端文档「APNs 推送开发指南」目录下的集成 **APNs 远程推送**。为 -1 时不改变角标数，传入相应数字表示把角标数改为指定的数字，最大不超过 9999。
verifyBlacklist	Int	否	是否过滤接收用户黑名单列表，0 表示为不过滤、 1 表示为过滤，默认为 0。
isPersisted	Int	否	是否需要为收件人在历史消息云端存储服务中存储此条消息。`0` 表示不存储；`1` 表示存储。默认值为 `1`，存储（依赖**单群聊消息云端存储服务**）。 此属性不影响离线消息功能，用户未在线时都会转为离线消息存储。 **提示**：一般情况下（第 1、2 种情况），客户端是否存储消息不依赖此参数。以下第 3 种情况属于例外： 1. 如果消息属于内置消息类型，客户端 SDK 会根据消息类型本身的存储属性标识判断是否存入本地数据库。详见消息类型概述。 2. 如果消息属于自定义消息类型，则客户端 SDK 会根据该类型在客户端上注册时的存储属性标识判断是否需要存入本地数据库。 3. 如果消息属于客户端 App 上未注册自定义消息类型（例如客户端使用的 App 版本过旧），则客户端 SDK 会根据当前参数值确定是否将消息存储在本地。但因消息类型未注册，客户端无法解析显示该消息。
contentAvailable	Int	否	仅目标用户为 iOS 设备时有效，应用处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0。
expansion	Boolean	否	是否为可扩展消息，默认为 `false`，设为 `true` 时终端在收到该条消息后，可对该条消息设置扩展信息。 移动端 SDK 4.0.3 版本、Web 端 3.0.7 版本支持此功能。

参数	类型	必传	说明
extraContent	Object	否	仅在 `expansion` 为 `true` 时有效。 自定义的消息扩展信息，该字段接受 JSON 字符串格式的键值对（key-value pairs）。请注意区别于消息体内的 `extra` 字段，`extraContent` 的值在消息发送后可修改，修改方式请参见服务端 API 接口文档消息扩展，或参考各客户端「消息扩展」接口文档。 **KV 详细要求**：以 Key、Value 的方式进行设置，如：`{"type":"3"}`。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符`+ = _` 的组合方式，不支持汉字。Value 最大 4096 个字符。单次可设置最多 100 对 KV 扩展信息，单条消息最多可设置 300 对 KV 扩展信息。
disablePush	Boolean	否	是否为静默消息，默认为 false，设为 true 时终端用户离线情况下不会收到通知提醒。
pushExt	String	否	配置消息的推送通知，如推送通知的标题等。`disablePush` 属性为 `true` 时此属性无效。具体请查看下方 `pushExt` 参数说明。

- `pushExt` 参数说明

  `pushExt` 参数支持设置消息推送通知的标题、推送内容模板、是否强制通知及推送 `ChannelID` 等。`pushExt` 为 JSON 结构请求时需要做转义处理。

pushExt 参数	类型	必传	说明
title	String	否	通知栏显示标题，最长不超过 50 个字符，默认情况下通知标题单聊会话显示用户名称，群聊会话显示群名称。
templateId	String	否	推送模板 ID，设置后根据目标用户通过 SDK 设置的语言环境，匹配模板中设置的语言内容进行推送。未匹配成功时使用默认内容进行推送。模板内容在“控制台-自定义推送文案”中进行设置。详情请查看自定义多语言推送模板。
forceShowPushContent	Number	否	是否越过客户端配置，强制在推送通知内显示通知内容（`pushContent`）。默认值 `0` 表示不强制，`1` 表示强制。 **说明**：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置`forceShowPushContent` 为 `1` 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
pushConfigs	Array	否	按厂商设置不同推送属性。支持的推送通道值为 `MI`（小米）、`HONOR`（荣耀）、`HW`（华为）、`OPPO`、`VIVO`、`APNs`、`FCM`。
pushConfigs.HONOR.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。

pushExt 参数	类型	必传	说明
pushConfigs.HONOR.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.channelId	String	否	华为推送通知渠道的 ID，详细请参见 华为自定义通知渠道。更多通知渠道信息，请参见 Android 官方文档。
pushConfigs.HW.importance	String	否	华为通知栏消息优先级，取值： • NORMAL（默认值，重要信息） • LOW
pushConfigs.HW.image	String	否	华为推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
pushConfigs.MI.channelId	String	否	小米推送通知渠道的 ID，详细请参见 小米推送消息分类新规。
pushConfigs.MI.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息的右侧图标 URL，若不设置，则不展示通知栏右侧图标。 • 国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。 • 图片要求：大小 120 * 120px，格式为 png 或者 jpg 格式。
pushConfigs.OPPO.channelId	String	否	OPPO 推送通知渠道的 ID，详细请参见 OPPO PUSH 通道升级说明。

pushExt 参数	类型	必传	说明
pushConfigs.VIVO.classification	Number	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
pushConfigs.VIVO.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category`，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
pushConfigs.APNs.thread-id	String	否	iOS 平台通知栏分组 ID。相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示。
pushConfigs.APNs.apns-collapse-id	String	否	适用于 iOS 平台。设置后设备收到有相同 ID 的消息，会合并成一条。 仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.richMediaUri	String	否	适用于 iOS 平台。iOS 推送自定义的通知栏消息右侧图标 URL，需要 App 自行解析 `richMediaUri` 并实现展示。仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
pushConfigs.FCM.channelId	String	否	FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见安卓官方文档。
pushConfigs.FCM.collapse_key	String	否	适用于 Android 平台。可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
pushConfigs.FCM.imageUrl	String	否	适用于 Android 平台。FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。

- `pushExt` 结构示例

```
{
"title":"you have a new message.",
"templateId":"123456",
"forceShowPushContent":0,
"pushConfigs": [
{
"HW": {
"channelId": "hw-123",
"importance": "NORMAL",
"image":"https://example.com/image.png"
}
},
{
"MI": {
"channelId": "mi-123",
"large_icon_uri":"https://example.com/image.png"
}
},
{
"OPPO": {"channelId": "oppo-123"}
},
{
"VIVO" : {"classification": "0"}
},
{
"APNs": {
"thread-id": "123",
"apns-collapse-id":"123456",
"richMediaUri":"https://example.com/image.png"
}
},
{
"FCM": {
"channelId":"rongcloud_channelid",
"collapse_key":"1234",
"imageUrl":"https://example.com/image.png"
}
}
]
}
```

请求示例

```
POST /message/private/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&toUserId=2193
erId=2192&objectName=RC:TxtMsg&pushContent=thisisapush&pushData=%7B%22pushData%22%3A%22hello%22%7D&count
rifyBlacklist=0&isPersisted=1&isIncludeSender=0&disablePush=false&expansion=false
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送单聊模板消息

更新时间:2024-08-30

应用下的用户可向其他用户发送单聊模板消息。通过您自定义的模板，可实现一次向多个用户发送不同的消息内容。例如，App 可以向应用中不同用户发送消息告知已获得的积分。

- 单次最多向 1000 个用户发送单聊模板消息。
- 支持通过模板字段区分不同用户收到的离线推送通知内容。如有需要，请在 `pushContent` 中按照 `toUserId` 中的用户 ID 列表逐个定义推送内容。
- 通过该接口发送的消息，默认不会向消息发件人客户端同步，也不会存入发件用户的历史消息记录。如需同步，请参见 `isIncludeSender` 参数用法。

如何配置与使用消息模板

消息模板通过字段内容模板（带标识位），和按接收者提供的标识位定义，实现单次向多个接收者发送不同内容的效果。

定义内容模板

发送消息时，可在 content、pushContent、pushData 字段中传入带标识位的字段内容，例如：

```
"toUserId":["21","22","23"],
"content":"{\"content\":\"{c}{d}\",\"extra\":\"bb\"}",
"pushContent":["hello {c}","hello {c}","hello {c}"],
```

上例中的 {c}、{d} 为自定义的标识位。

- 当前支持定义模板的字段包括：`content`、`pushContent`、`pushData`。
- 消息内容（`content`）字段仅支持插入一个模板，所有接收者共用该模板。
- 消息推送通知内容（`pushContent`）、推送附加数据（`pushData`）字段类型为数组，必须按照 `toUserId` 中的用户 ID 列表逐个定义模板。即使不在 `pushContent`、`pushData` 中使用模板标识位，或所有接收者接收相同内容，都必须按照 `toUserId` 中的用户 ID 列表逐个定义各自的 `pushContent`、`pushData`。

指定模板内容标识位的值

按照收件人列表（toUserId）在 values 字段提供标识位的定义，即为各个收件人指定需要接收的个性化内容。

```
"values":[
{
"{c}":"Tom",
"{d}":"：2"
},
{
"{c}":"Jerry",
"{d}":"：5"},
{
"{c}":"Rose",
"{d}":"：10"}
],
```

上面示例中，各接收者在线时收到的消息内容如下：

- 用户 ID 为 21 收到：Tom：2
- 用户 ID 为 22 收到：Jerry：5
- 用户 ID 为 23 收到：Rose：10

如果接收者离线，各自收到的推送通知的内容（pushContent）也不同，分别为 Hello Tom，Hello Jerry，Hello Rose。

> 提示
>
> 如 content 中定义了标识 {d} ，则在 values 中必须设置 {d} 的值，否则 {d} 会以文本方式随消息发送给用户。

请求方法

POST： https://数据中心域名/message/private/publish_template.json

频率限制： 每分钟限发送 6000 条信息。请注意，如果一次发送给 1000 人，视为 1000 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/json，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
toUserId	String[]	是	接收用户 ID，提供多个本参数可以实现向多人发送消息，上限为 1000 人。
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。

参数	类型	必传	说明
values	Array of objects	是	为消息内容（content）、推送通知内容（pushContent）、推送数据（pushData）中的标识位（标识位示例：{d}）提供对应的值。
content	String	是	所发送消息的内容，单条消息最大 128k。 • **内置消息类型**：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • **自定义消息类型**（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
pushContent	String[]	是	指定收件人离线时触发的远程推送通知中的通知内容。**注意**：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。支持定义模板标识位，使用 values 中的值进行替换。 • 如果消息类型（objectName 字段）为即时通讯服务预定义的消息类型，填写该字段后，离线推送通知中显示模板定义的推送内容，而非消息类型的默认推送内容。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 pushContent 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 pushContent 字段对应数组传空值禁用离线推送。
pushData	String[]	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 appData（**提示**：rc 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 appData。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。1 表示同步，默认值为 0，即不同步。**注意**，仅设置该参数无法确保发件人客户端一定能获取到该条已发消息，您可能还需要启用其他服务。详见发件人客户端如何同步已发消息。
verifyBlacklist	Int	否	是否过滤发送人黑名单列表，0 为不过滤、 1 为过滤，默认为 0 不过滤。
isPersisted	Int	否	是否需要为收件人在历史消息云端存储服务中存储此条消息。0 表示不存储；1 表示存储。默认值为 1，存储（依赖**单群聊消息云端存储服务**）。 此属性不影响离线消息功能，用户未在线时都会转为离线消息[?]存储。 **提示**：一般情况下（第 1、2 种情况），客户端是否存储消息不依赖此参数。以下第 3 种情况属于例外： 1. 如果消息属于内置消息类型，客户端 SDK 会根据消息类型本身的存储属性标识判断是否存入本地数据库。详见消息类型概述。 2. 如果消息属于自定义消息类型，则客户端 SDK 会根据该类型在客户端上注册时的存储属性标识判断是否需要存入本地数据库。 3. 如果消息属于客户端 App 上未注册自定义消息类型（例如客户端使用的 App 版本过旧），则客户端 SDK 会根据当前参数值确定是否将消息存储在本地。但因消息类型未注册，客户端无法解析显示该消息。

参数	类型	必传	说明
contentAvailable	Int	否	仅目标用户为 iOS 设备时有效，对 SDK 处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
expansion	Boolean	否	是否为可扩展消息，默认为 false，设为 true 时终端在收到该条消息后，可对该条消息设置扩展信息。
disablePush	Boolean	否	是否为静默消息，默认为 false，设为 true 时终端用户离线情况下不会收到通知提醒。
pushExt	Object	否	配置消息的推送通知，如推送通知的标题等。`disablePush` 属性为 `true` 时此属性无效。具体请查看下方 `pushExt` 参数说明。

- `pushExt` 参数说明

  `pushExt` 参数支持设置消息推送通知的标题、推送内容模板、是否强制通知及推送 `ChannelID` 等。`pushExt` 为 JSON 结构请求时需要做转义处理。

pushExt 参数	类型	必传	说明
title	String	否	通知栏显示标题，最长不超过 50 个字符，默认情况下通知标题单聊会话显示用户名称，群聊会话显示群名称。
templateId	String	否	推送模板 ID，设置后根据目标用户通过 SDK 设置的语言环境，匹配模板中设置的语言内容进行推送。未匹配成功时使用默认内容进行推送。模板内容在“控制台-自定义推送文案”中进行设置。详见自定义多语言推送模板。
forceShowPushContent	Number	否	是否越过客户端配置，强制在推送通知内显示通知内容（`pushContent`）。默认值 `0` 表示不强制，`1` 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置`forceShowPushContent` 为 `1` 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
pushConfigs	Array	否	按厂商设置不同推送属性。支持的推送通道值为 `MI`（小米）、`HONOR`（荣耀）、`HW`（华为）、`OPPO`、`VIVO`、`APNs`、`FCM`。
pushConfigs.HONOR.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。

pushExt 参数	类型	必传	说明
pushConfigs.HONOR.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.channelId	String	否	华为推送通知渠道的 ID，详细请参见 华为自定义通知渠道。更多通知渠道信息，请参见 Android 官方文档。
pushConfigs.HW.importance	String	否	华为通知栏消息优先级，取值： • NORMAL（默认值，重要信息） • LOW
pushConfigs.HW.image	String	否	华为推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.MI.channelId	String	否	小米推送通知渠道的 ID，详细请参见 小米推送消息分类新规。
pushConfigs.MI.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息的右侧图标 URL，若不设置，则不展示通知栏右侧图标。 • 国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。 • 图片要求：大小 120 * 120px，格式为 png 或者 jpg 格式。
pushConfigs.OPPO.channelId	String	否	OPPO 推送通知渠道的 ID，详细请参见 OPPO PUSH 通道升级说明。
pushConfigs.VIVO.classification	Number	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。

pushExt 参数	类型	必传	说明
pushConfigs.VIVO.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category`，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
pushConfigs.APNs.thread-id	String	否	iOS 平台通知栏分组 ID。相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示。
pushConfigs.APNs.apns-collapse-id	String	否	适用于 iOS 平台。设置后设备收到有相同 ID 的消息，会合并成一条。 仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.richMediaUri	String	否	适用于 iOS 平台。iOS 推送自定义的通知栏消息右侧图标 URL，需要 App 自行解析 `richMediaUri` 并实现展示。仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
pushConfigs.FCM.channelId	String	否	FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见安卓官方文档。
pushConfigs.FCM.collapse_key	String	否	适用于 Android 平台。可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
pushConfigs.FCM.imageUrl	String	否	适用于 Android 平台。FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。

- `pushExt` 结构示例

```
{
"title":"you have a new message.",
"templateId":"123456",
"forceShowPushContent":0,
"pushConfigs": [
{
"HW": {
"channelId": "hw-123",
"importance": "NORMAL",
"image":"https://example.com/image.png"
}
},
{
"MI": {
"channelId": "mi-123",
"large_icon_uri":"https://example.com/image.png"
}
},
{
"OPPO": {"channelId": "oppo-123"}
},
{
"VIVO" : {"classification": "0"}
},
{
"APNs": {
"thread-id": "123",
"apns-collapse-id":"123456",
"richMediaUri":"https://example.com/image.png"
}
},
{
"FCM": {
"channelId":"rongcloud_channelid",
"collapse_key":"1234",
"imageUrl":"https://example.com/image.png"
}
}
]
}
```

请求示例

```
POST /message/private/publish_template.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/json

{
"fromUserId":"fromuser",
"objectName":"RC:TxtMsg",
"content":"{\"content\":\"{c}{d}{e}\",\"extra\":\"bb\"}",
"toUserId":["21","22"],
"values":[{"{c}":"1","{d}":"2","{e}":"3"},{"{c}":"4","{d}":"5","{e}":"6"}],
"pushContent":["push{c}","push{c}"],
"pushData":["pushdata","{c}{e}"],
"verifyBlacklist":0,
"disablePush":false,
"expansion":false
}
```

上面示例中用户 ID 为 21 的用户，收到信息为 123，上面示例中用户 ID 为 22 的用户，收到信息为 456。

> 提示
>
> 如 content 中定义了标识 {d} ，则在 values 中需要对 {d} 进行设置，否则 {d} 会以文本方式随消息一起发送给用户。toUserId、values、pushContent、pushData 的数量必须相等。

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送单聊状态消息

更新时间:2024-08-30

应用下的用户可向其他用户发送单聊状态消息。通过该接口发送的消息，仅收件人在线时可收到。如收件人当前不在线，则无法再收到此条消息。

- 通过该接口发送的消息，默认不会向消息发件人客户端同步。如需同步，请参见 `isIncludeSender` 参数用法。
- 单次最多向 1000 个用户发送消息。

关于单聊状态消息

服务端提供单聊状态消息接口 `/statusmessage/private/publish.json`。任何类型的消息，只要通过该接口发送，均具有以下特点：

- 只有收件人用户在线时会收到此条消息。
- 在服务端不计数、不存储。因此，如果接收者当前未在线，则不会再收到此条消息，也无法从服务端的历史消息中获取该消息。
- 默认不支持全量消息路由。

移动端在接收该接口发送的消息时，与处理其他群聊会话消息的方式一致，会根据消息类型本身的存储、计数属性决定是否计入未读消息数、是否进行本地存储。如需了解即时通讯服务预定义的消息类型的存储、计数属性，可参见消息类型概述。如果发送的是您自定义的消息类型，需要关注该自定义消息类型在客户端的具体计数属性与存储属性定义。

请求方法

POST： https://数据中心域名/statusmessage/private/publish.json

频率限制： 每分钟限发送 6000 条信息，每次可发送目标用户上限为 1000 人。请注意，如果一次发送给 1000 人，视为 1000 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。
toUserId	String	是	接收用户 ID，支持向多人发送消息，每次上限为 1000 人。

参数	类型	必传	说明
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • 自定义消息类型（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
verifyBlacklist	Int	否	是否过滤发送人黑名单列表，0 表示为不过滤、 1 表示为过滤，默认为 0 不过滤。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。1 表示同步，默认值为 0，即不同步。注意，该接口用于发送状态消息，因此仅支持在发件人已登陆客户端（在线）的情况下同步已发消息。

请求示例

```
POST /statusmessage/private/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&toUserId=2191
erId=2192&objectName=RC:TxtMsg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送群聊消息

更新时间:2024-08-30

向应用下的单个群组或多个群组发送消息。请注意，通过 Server API 向群组发送消息时，不要求发送者为群组成员。App 需要自行控制发消息权限。该接口可以实现以下功能：

- **发送普通群聊消息**：单次支持向最多 3 个群组发送消息。
- **发送定向群聊消息**：可向群组中指定的一个或多个用户发送消息，但单次仅支持指定一个目标群组。

通过该接口发送的消息，默认不会向消息发件人客户端同步，也不会存入发件用户的历史消息记录。如需同步，请参见 `isIncludeSender` 参数用法。

请求方法

POST： https://数据中心域名/message/group/publish.json

频率限制： 每秒钟限发送 20 条消息。请注意，如果一次向 3 个群组发送消息，视为 3 条消息。另见已知问题 1。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

> 提示
>
> 发送群聊定向消息时，仅支持一个 `toGroupId`，且不支持 `expansion`、`extraContent`、`disablePush`、`pushExt` 字段。

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID，通过 Server API 非群成员也可以向群组中发送消息。 **注意**：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
toGroupId	String	是	接收消息的群 ID。支持最多 3 个群组 ID。发送群聊定向消息时，仅支持传入一个群组 ID。
toUserId	String	否	发送群聊定向消息时，接收消息的群成员用户 ID 列表，群中其他用户无法收到该定向消息。仅当 `toGroupId` 传入单个群组 ID 时有效。
objectName	String	是	接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 **注意**：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。

参数	类型	必传	说明
content	String	是	所发送消息的内容，单条消息最大 128k。 • **内置消息类型**：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 `{"content":"Hello world!"}` 序列化后的结果作为此处 `content` 字段的值。 • **自定义消息类型（`objectName` 字段必须指定为自定义消息类型）**：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
pushContent	String	否	指定收件人离线时触发的远程推送通知中的通知内容。**注意**：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。 • 如果消息类型（`objectName` 字段）为即时通讯服务预定义消息类型中的用户内容类消息格式，可不填写该字段，远程推送通知默认使用服务端预置的推送通知内容。 • 如果消息类型（`objectName` 字段）为即时通讯服务预定义消息类型中通知类、信令类（"撤回命令消息" 除外），且需要支持远程推送通知，则必须填写 `pushContent`，否则收件人不在线时无法收到远程推送通知。如无需触发远程推送，可不填该字段。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 `pushContent` 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 `pushContent` 字段留空禁用远程推送通知。
pushData	String	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 `appData`（**提示**：`rc` 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 `appData`。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。`1` 表示同步，默认值为 `0`，即不同步。**注意**，仅设置该参数无法确保发件人客户端一定能获取到该条已发消息，您可能还需要启用其他服务。详见发件人客户端如何同步已发消息。
isPersisted	Int	否	是否需要为收件人在历史消息云端存储服务中存储此条消息。`0` 表示不存储；`1` 表示存储。默认值为 `1`，存储（依赖**单群聊消息云端存储服务**）。 **注意**：即使已开通**单群聊消息云存储**功能，群组定向消息也不会存入服务端历史消息记录。如有需要，请提交工单 申请开通**群定向消息云存储**服务。 此属性不影响离线消息功能，用户未在线时都会转为离线消息 存储。 **提示**：一般情况下（第 1、2 种情况），客户端是否存储消息不依赖此参数。以下第 3 种情况属于例外： 1. 如果消息属于内置消息类型，客户端 SDK 会根据消息类型本身的存储属性标识判断是否存入本地数据库。详见消息类型概述。 2. 如果消息属于自定义消息类型，则客户端 SDK 会根据该类型在客户端上注册时的存储属性标识判断是否需要存入本地数据库。 3. 如果消息属于客户端 App 上未注册自定义消息类型（例如客户端使用的 App 版本过旧），则客户端 SDK 会根据当前参数值确定是否将消息存储在本地。但因消息类型未注册，客户端无法解析显示该消息。

参数	类型	必传	说明
isMentioned	Int	否	是否为 @ 消息，不传时默认为非 @ 消息（效果等于传 `0`）。如果需要发送 @ 消息，必须指定为 `1`，且必须在消息内容字段（`content`）内部携带 @ 相关信息（`mentionedInfo`，可参考下方请求示例）。关于 `mentionedInfo` 结构的详细说明，参见如何发送 @ 消息。
contentAvailable	Int	否	针对 iOS 平台，对 SDK 处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
expansion	Boolean	否	是否为可扩展消息，默认为 `false`，设为 `true` 时终端在收到该条消息后，可对该条消息设置扩展信息。移动端 SDK 4.0.3 版本、Web 端 3.0.7 版本支持此功能。仅当 `toGroupId` 传入单个群组 ID 时有效。
extraContent	Object	否	仅当 `expansion` 为 `true` 时有效，仅当 `toGroupId` 传入单个群组 ID 时有效。 自定义的消息扩展信息，该字段接受 JSON 字符串格式的键值对（key-value pairs）。请注意区别于消息体内的 `extra` 字段，`extraContent` 的值在消息发送后可修改，修改方式请参见服务端 API 接口文档消息扩展，或参考各客户端「消息扩展」接口文档。 **KV 详细要求**：以 Key、Value 的方式进行设置，如：`{"type":"3"}`。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符`+ = _` 的组合方式，不支持汉字。Value 最大 4096 个字符。单次可设置最多 100 对 KV 扩展信息，单条消息最多可设置 300 对 KV 扩展信息。
disablePush	Boolean	否	是否为静默消息，默认为 false，设为 true 时终端用户离线情况下不会收到通知提醒。仅当 `toGroupId` 传入单个群组 ID 时有效。
pushExt	String	否	配置消息的推送通知，如推送通知的标题等。`disablePush` 属性为 `true` 时此属性无效。具体请查看下方 `pushExt` **参数说明**。仅当 `toGroupId` 传入单个群组 ID 时有效。

- `pushExt` **参数说明**

`pushExt` 参数支持设置消息推送通知的标题、推送内容模板、是否强制通知及推送 `ChannelID` 等。`pushExt` 为 JSON 结构请求时需要做转义处理。

pushExt 参数	类型	必传	说明
title	String	否	通知栏显示标题，最长不超过 50 个字符，默认情况下通知标题单聊会话显示用户名称，群聊会话显示群名称。
templateId	String	否	推送模板 ID，设置后根据目标用户通过 SDK 设置的语言环境，匹配模板中设置的语言内容进行推送，未匹配成功时使用默认内容进行推送，模板内容在“控制台-自定义推送文案”中进行设置。详情请查看自定义多语言推送模板。
forceShowPushContent	Number	否	是否越过客户端配置，强制在推送通知内显示通知内容（`pushContent`）。默认值 `0` 表示不强制，`1` 表示强制。 **说明**：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置`forceShowPushContent` 为 `1` 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
pushConfigs	Array	否	按厂商设置不同推送属性。支持的推送通道值为 `MI`（小米）、`HONOR`（荣耀）、`HW`（华为）、`OPPO`、`VIVO`、`APNs`、`FCM`。

pushExt 参数	类型	必传	说明
pushConfigs.HONOR.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
pushConfigs.HONOR.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.channelId	String	否	华为推送通知渠道的 ID，详细请参见 华为自定义通知渠道。更多通知渠道信息，请参见 Android 官方文档。
pushConfigs.HW.importance	String	否	华为通知栏消息优先级，取值： • NORMAL（默认值，重要信息） • LOW
pushConfigs.HW.image	String	否	华为推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
pushConfigs.MI.channelId	String	否	小米推送通知渠道的 ID，详细请参见 小米推送消息分类新规。

pushExt 参数	类型	必传	说明
pushConfigs.MI.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息的右侧图标 URL，若不设置，则不展示通知栏右侧图标。 • 国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。 • 图片要求：大小 120 * 120px，格式为 png 或者 jpg 格式。
pushConfigs.OPPO.channelId	String	否	OPPO 推送通知渠道的 ID，详细请参见 OPPO PUSH 通道升级说明 。
pushConfigs.VIVO.classification	Number	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明 。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
pushConfigs.VIVO.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明 。如果指定 `category` ，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
pushConfigs.APNs.thread-id	String	否	iOS 平台通知栏分组 ID。相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示。
pushConfigs.APNs.apns-collapse-id	String	否	适用于 iOS 平台。设置后设备收到有相同 ID 的消息，会合并成一条。 仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.richMediaUri	String	否	适用于 iOS 平台。iOS 推送自定义的通知栏消息右侧图标 URL，需要 App 自行解析 `richMediaUri` 并实现展示。仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
pushConfigs.FCM.channelId	String	否	FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见安卓官方文档 。
pushConfigs.FCM.collapse_key	String	否	适用于 Android 平台。可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。

pushExt 参数	类型	必传	说明
pushConfigs.FCM.imageUrl	String	否	适用于 Android 平台。FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。

- pushExt 结构示例

```
{
"title":"you have a new message.",
"templateId":"123456",
"forceShowPushContent":0,
"pushConfigs": [
{
"HW": {
"channelId": "hw-123",
"importance": "NORMAL",
"image":"https://example.com/image.png"
}
},
{
"MI": {
"channelId": "mi-123",
"large_icon_uri":"https://example.com/image.png"
}
},
{
"OPPO": {"channelId": "oppo-123"}
},
{
"VIVO" : {"classification": "0"}
},
{
"APNs": {
"thread-id": "123",
"apns-collapse-id":"123456",
"richMediaUri":"https://example.com/image.png"
}
},
{
"FCM": {
"channelId":"rongcloud_channelid",
"collapse_key":"1234",
"imageUrl":"https://example.com/image.png"
}
}
]
}
```

请求示例

普通群聊消息示例

```
POST /message/group/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

fromUserId=0MglYiqxW&toGroupId=d9Uia1h8C&content=%7B%22content%22%3A%22%40%E6%B5%8B%E8%AF%9511%20c%23hel
%2C%22mentionedInfo%22%3A%7B%22type%22%3A2%2C%20%22userIdList%22%3A%5B%22wX7zFv8dR%22%5D%2C%22mentionedC
t%22%3A%22%22%7D%7D&objectName=RC%3ATxtMsg&pushContent=thisisapush&pushData%22%3A%22hello%22%7D&isPersis
&isIncludeSender=0&isMentioned=1
```

定向群聊消息示例

```
POST /message/group/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&toGroupId=219
serId=123&toUserId=456&objectName=RC%3ATxtMsg&pushContent=thisisapush&pushData%22%3A%22hello%22%7D&isPer
d=1&isIncludeSender=0&isMentioned=1&mentionedInfo=%7B%22type%22%3A2%2C%22userIdList%22%3A%5B%22123%22%2C
6%22%5D%2C%22mentionedContent%22%3A%22%E6%9C%89%E4%BA%BA%40%E4%BD%A0%22%7D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送群聊状态消息

更新时间:2024-08-30

应用下的用户可以向群组中发送群聊状态消息。通过该接口发送的消息，仅当前在线群成员用户可收到。如群成员当前未在线，则无法再收到此条状态消息。

- 通过该接口发送的消息，默认不会向消息发件人客户端同步。如需同步，请参见 `isIncludeSender` 参数用法。
- 单次最多向 3 个群组发送消息。

关于群聊状态消息

服务端提供群聊状态消息接口 `/statusmessage/group/publish.json`。任何类型的消息，只要通过该接口发送，均具有以下特点：

- 仅在线群成员可以收到此条消息。
- 在服务端不计数、不存储。因此，如果接收者当前未在线，则不会再收到此条消息，也无法从服务端的历史消息中获取该消息。
- 默认不支持全量消息路由。

移动端在接收该接口发送的消息时，与处理其他单聊会话消息的方式一致，会根据消息类型本身的存储、计数属性决定是否计入未读消息数、是否进行本地存储。如需了解即时通讯服务预定义的消息类型的存储、计数属性，可参见消息类型概述。如果发送的是您自定义的消息类型，需要关注该自定义消息类型在客户端的具体计数属性与存储属性定义。

请求方法

POST： https://数据中心域名/statusmessage/group/publish.json

频率限制： 每秒钟限发送 20 条消息。请注意，如果一次向 3 个群组发送消息，视为 3 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID，通过 Server API 非群成员也可以向群组中发送消息。
toGroupId	String	是	接收群ID，提供多个本参数可以实现向多群发送消息，最多不超过 3 个群组。
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。

参数	类型	必传	说明
content	String	是	所发送消息的内容，单条消息最大 128k。 • **内置消息类型**：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • **自定义消息类型（objectName 字段必须指定为自定义消息类型）**：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。1 表示同步，默认值为 0，即不同步。注意，该接口用于发送状态消息，因此仅支持在发件人已登陆客户端（在线）的情况下同步已发消息。

请求示例

```
POST /statusmessage/group/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&toGroupId=219
roupId=2192&objectName=RC:TxtMsg&isIncludeSender=0
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送超级群消息

更新时间:2024-08-30

向应用下的单个或多个超级群发送消息。

该接口可以实现以下功能：

- **发送普通超级群消息**：单次支持向最多 3 个超级群发送消息，但单次仅支持指定一个频道 ID。如果超级群下无此频道 ID，则不会下发此条消息。
- **发送定向群聊消息**：可向超级群频道中指定一个或多个用户发送消息，但单次仅支持指定一个超级群 ID 和一个频道 ID。如果向私有频道发送定向消息，默认情况下只有已加入该私有频道的目标用户才能收到消息。

通过 Server API 向超级群发送消息时：

- 不要求发送者为超级群成员。App 需要自行控制发消息权限。
- 终端用户在线状态下，发送者会自动同步接收该条已发出消息，该条已发消息会被保存到该用户的服务端历史消息中。

> 提示
>
> - 如果您的应用/环境在 2022.10.13 日及以后开通超级群服务，超级群业务中会包含一个 ID 为 `RCDefault` 的默认频道。如果发消息时不指定频道 ID，则该消息会发送到 `RCDefault` 频道中。客户端 SDK 在获取 `RCDefault` 频道的历史消息时，需要传入该频道 ID。
> - 如果您的应用/环境在 2022.10.13 日前已开通超级群服务，在发送消息时如果不指定频道 ID，则该消息不属于任何频道。客户端 SDK 获取历史消息时，如果不传入频道 ID，可获取不属于任何频道的消息。即时通讯服务支持客户调整服务至最新行为。该行为调整将影响客户端、服务端收发消息、获取会话、清除历史消息、禁言等多个功能。如有需要，请提交工单咨询详细方案。

应用下的用户可以向应用下的单个超级群或多个超级群发送消息。通过 Server API 发送超级群消息时，终端用户在线状态下，发送者也会接收该消息，同时保存到该用户的历史消息中。

超级群消息发送成功后支持修改消息内容，详见修改超级群消息。

请求方法

POST： https://数据中心域名/message/ultragroup/publish.json

频率限制： 每秒钟限发送 100 条超级群消息。单个频道每秒钟最多发送 20 条；不传频道参数时（`busChannel`），往指定群中发消息限每秒钟 20 条。请注意，如果一次向 3 个超级群发送消息，视为发送 3 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID，通过 Server API 非群成员也可以向群组中发送消息。 **注意**：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
toGroupIds	String[]	是	超级群 ID，提供多个本参数可以实现向多个群发送消息，最多不超过 3 个超级群。默认情况下只有已加入超级群的用户才能收到消息。
busChannel	String	否	超级群频道 ID。如果使用默认频道，请传入 `RCDefault`，否则请传入自建的频道 ID。如果超级群 ID 下无此频道 ID，则不会下发此条消息。
toUserIds	String[]	否	接收定向消息的用户 ID，最多不超过 300 个用户 ID。如果传入的 `busChannel` 为私有频道 ID，默认情况下只有已加入私有频道的用户才能收到消息。仅当 `toGroupIds` 数组中只包含单个超级群 ID 时有效。
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 **注意**：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
content	String	是	所发送消息的内容，单条消息最大 128k。 • **内置消息类型**：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 `{"content":"Hello world!"}` 序列化后的结果作为此处 `content` 字段的值。 • **自定义消息类型（`objectName` 字段必须指定为自定义消息类型）**：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
pushContent	String	否	指定收件人离线时触发的远程推送通知中的通知内容。**注意**：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。 • 如果消息类型（`objectName` 字段）为即时通讯服务预定义消息类型中的用户内容类消息格式，可不填写该字段，远程推送通知默认使用服务端预置的推送通知内容。 • 如果消息类型（`objectName` 字段）为即时通讯服务预定义消息类型中通知类、信令类（"撤回命令消息" 除外），且需要支持远程推送通知，则必须填写 `pushContent`，否则收件人不在线时无法收到远程推送通知。如无需触发远程推送，可不填该字段。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 `pushContent` 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 `pushContent` 字段留空禁用远程推送通知。
pushData	String	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 `appData`（提示：`rc` 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 `appData`。

参数	类型	必传	说明
isPersisted	Int	否	是否需要为收件人在历史消息云端存储服务中存储此条消息。0 表示不存储；1 表示存储。默认值为 1，存储。 一般情况下（第 1、2 种情况），客户端是否存储消息不依赖此参数。以下第 3 种情况属于例外： 1. 如果消息属于内置消息类型，客户端 SDK 会根据消息类型本身的存储属性标识判断是否存入本地数据库。详见消息类型概述。 2. 如果消息属于自定义消息类型，则客户端 SDK 会根据该类型在客户端上注册时的存储属性标识判断是否需要存入本地数据库。 3. 如果消息属于客户端 App 上未注册自定义消息类型（例如客户端使用的 App 版本过旧），则客户端 SDK 会根据当前参数值确定是否将消息存储在本地。但因消息类型未注册，客户端无法解析显示该消息。
isCounted	Int	否	用户未在线时是否计入未读消息数。0 表示为不计数、1 表示为计数，默认为 1。
isMentioned	Int	否	是否为 @ 消息，不传时默认为非 @ 消息（效果等于传 0）。如果需要发送 @ 消息，必须指定为 1，且必须在消息内容字段（content）内部携带 @ 相关信息（mentionedInfo，可参考下方请求示例）。关于 mentionedInfo 结构的详细说明，参见如何发送 @ 消息。
contentAvailable	Int	否	针对 iOS 平台，对 SDK 处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
pushExt	Object	否	配置消息的推送通知，如推送通知的标题等。具体请查看下方 pushExt 参数说明。
expansion	Boolean	否	是否为可扩展消息，默认为 false，设为 true 时终端在收到该条消息后，可对该条消息设置扩展信息
extraContent	Object	否	仅在 expansion 为 true 时有效。 自定义的消息扩展信息，该字段接受 JSON 字符串格式的键值对（key-value pairs）。请注意区别于消息体内的 extra 字段，extraContent 的值在消息发送后可修改，修改方式请参见服务端 API 接口文档消息扩展，或参考各客户端「消息扩展」接口文档。 **KV 详细要求**：以 Key、Value 的方式进行设置，如：{"type":"3"}。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符+ = _ 的组合方式，不支持汉字。Value 最大 4096 个字符。单次可设置最多 100 对 KV 扩展信息，单条消息最多可设置 300 对 KV 扩展信息。

- pushExt 参数说明

pushExt 参数支持设置消息推送通知的标题、推送内容模板、是否强制通知及推送 ChannelID 等。pushExt 为 JSON 结构请求时需要做转义处理。

pushExt 参数	类型	必传	说明
title	String	否	通知栏显示标题，最长不超过 50 个字符，默认情况下通知标题单聊会话显示用户名称，群聊会话显示群名称。
templateId	String	否	推送模板 ID，设置后根据目标用户通过 SDK 设置的语言环境，匹配模板中设置的语言内容进行推送，未匹配成功时使用默认内容进行推送，模板内容在“控制台-自定义推送文案”中进行设置。详情请查看自定义多语言推送模板。

pushExt 参数	类型	必传	说明
forceShowPushContent	Number	否	是否越过客户端配置，强制在推送通知内显示通知内容（pushContent）。默认值 0 表示不强制，1 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置forceShowPushContent 为 1 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
pushConfigs	Array	否	按厂商设置不同推送属性。支持的推送通道值为 MI（小米）、HONOR（荣耀）、HW（华为）、OPPO、VIVO、APNs、FCM。
pushConfigs.HONOR.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
pushConfigs.HONOR.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.channelId	String	否	华为推送通知渠道的 ID，详细请参见 华为自定义通知渠道。更多通知渠道信息，请参见 Android 官方文档。
pushConfigs.HW.importance	String	否	华为通知栏消息优先级，取值： • NORMAL（默认值，重要信息） • LOW
pushConfigs.HW.image	String	否	华为推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。

pushExt 参数	类型	必传	说明
pushConfigs.HW.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准 。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
pushConfigs.MI.channelId	String	否	小米推送通知渠道的 ID，详细请参见 小米推送消息分类新规 。
pushConfigs.MI.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息的右侧图标 URL，若不设置，则不展示通知栏右侧图标。 • 国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。 • 图片要求：大小 120 * 120px，格式为 png 或者 jpg 格式。
pushConfigs.OPPO.channelId	String	否	OPPO 推送通知渠道的 ID，详细请参见 OPPO PUSH 通道升级说明 。
pushConfigs.VIVO.classification	Number	否	vivo 推送通道类型。`0`：运营消息。`1`：系统消息。默认使用 App Key 在控制台 vivo 推送中设置的推送通道类型。详见 vivo 官方文档消息分类功能说明 。
pushConfigs.APNs.thread-id	String	否	iOS 平台通知栏分组 ID。相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示。
pushConfigs.APNs.apns-collapse-id	String	否	适用于 iOS 平台。设置后设备收到有相同 ID 的消息，会合并成一条。 仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.richMediaUri	String	否	适用于 iOS 平台。iOS 推送自定义的通知栏消息右侧图标 URL，需要 App 自行解析 `richMediaUri` 并实现展示。仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式”都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
pushConfigs.FCM.channelId	String	否	FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见安卓官方文档 。
pushConfigs.FCM.collapse_key	String	否	适用于 Android 平台。可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。

pushExt 参数	类型	必传	说明
pushConfigs.FCM.imageUrl	String	否	适用于 Android 平台。FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。

- pushExt 结构示例

```
{
"title":"you have a new message.",
"templateId":"123456",
"forceShowPushContent":0,
"pushConfigs": [
{
"HW": {
"channelId": "hw-123",
"importance": "NORMAL",
"image":"https://example.com/image.png"
}
},
{
"MI": {
"channelId": "mi-123",
"large_icon_uri":"https://example.com/image.png"
}
},
{
"OPPO": {"channelId": "oppo-123"}
},
{
"VIVO" : {"classification": "0"}
},
{
"APNs": {
"thread-id": "123",
"apns-collapse-id":"123456",
"richMediaUri":"https://example.com/image.png"
}
},
{
"FCM": {
"channelId":"rongcloud_channelid",
"collapse_key":"1234",
"imageUrl":"https://example.com/image.png"
}
}
]
}
```

请求示例

```
POST /message/ultragroup/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/json

{
"fromUserId":"why456",
"objectName":"RC:TxtMsg",
"content":"{\"content\":\"hh0217890\",\"mentionedInfo\":{\"type\":2,\"userIdList\":
[\"123\",\"456\"],\"mentionedContent\":\"有人@你\"}}",
"toGroupIds":["why66-ultra"],
"isPersisted":1,
"isMentioned":1}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送聊天室消息

更新时间:2024-08-30

应用下的用户可向指定的单个或多个聊天室发送消息。

- 通过该接口发送的消息，默认不会向消息发件人客户端同步。如需同步，请参见 `isIncludeSender` 参数用法。
- 支持单次向多个聊天室发送消息，建议不要超过 10 个。

> 提示
>
> 如需向应用下所有聊天室发送消息，可使用发送全体聊天室广播消息（/message/chatroom/broadcast.json）接口。详见发送全体聊天室广播消息。

请求方法

POST： https://数据中心域名/message/chatroom/publish.json

频率限制： 每秒钟限 100 条，同时向 10 个聊天室发送消息视为 10 条。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。
toChatroomId	String	是	接收聊天室 ID，提供多个本参数可以实现向多个聊天室发送消息，建议最多不超过 10 个聊天室。
objectName	String	是	接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。

参数	类型	必传	说明
content	String	是	所发送消息的内容，单条消息最大 128k。 • **内置消息类型**：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 `{"content":"Hello world!"}` 序列化后的结果作为此处 `content` 字段的值。 • **自定义消息类型（`objectName` 字段必须指定为自定义消息类型）**：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
isPersisted	Int	否	是否需要为收件人在历史消息云端存储服务中存储此条消息。`0` 表示不存储；`1` 表示存储。默认值为 `1`，存储（依赖**聊天室消息云端存储服务**）。 一般情况下（第 1、2 种情况），客户端是否存储消息不依赖此参数。以下第 3 种情况属于例外： 1. 如果消息属于内置消息类型，客户端 SDK 会根据消息类型本身的存储属性标识判断是否存入本地数据库。详见消息类型概述。 2. 如果消息属于自定义消息类型，则客户端 SDK 会根据该类型在客户端上注册时的存储属性标识判断是否需要存入本地数据库。 3. 如果消息属于客户端 App 上未注册自定义消息类型（例如客户端使用的 App 版本过旧），则客户端 SDK 会根据当前参数值确定是否将消息存储在本地。但因消息类型未注册，客户端无法解析显示该消息。 **注意**：客户端会在用户退出聊天室时自动清除本地的聊天室历史消息记录。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。`1` 表示同步，默认值为 `0`，即不同步。**注意**，仅设置该参数无法确保发件人客户端一定能获取到该条已发消息，您可能还需要启用其他服务。详见发件人客户端如何同步已发消息。
priority	Int	否	此条消息的优先级级别，默认不传的话为 0，此消息的优先级按照原有逻辑设定。 • 1：白名单消息（消息高保障，需要开通聊天室白名单功能）。 • 2：高优先级消息。 • 3：低优先级消息（消息优先抛弃，需要开通聊天室消息优先级服务）。

请求示例

```
POST /message/chatroom/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&toChatroomId=
toChatroomId=2193&objectName=RC:TxtMsg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送全体聊天室广播消息

更新时间:2024-08-30

应用下的用户可向应用下所有聊天室发送一条广播消息。仅当时在聊天室中的用户可以收到此条消息，后加入聊天室的用户无法收到此消息。

- 不支持将聊天室广播消息存入服务端聊天室历史消息记录（聊天室消息云端存储服务）。
- 不支持通过向聊天室发送广播消息来维护聊天室存活。如需控制聊天室存活时长，可前往控制台免费基础功能 页面，调整聊天室销毁等待时间。您也可以使用保活聊天室 API 动态设置需要保活的聊天室。
- 通过该接口发送的消息，默认不会向消息发件人客户端同步。如需同步，请参见 `isIncludeSender` 参数用法。

开通服务

使用聊天室广播消息功能前，请确认已为当前 App Key 开通相关服务。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/message/chatroom/broadcast.json

频率限制： 每秒钟限 1 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。
objectName	String	是	接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。

参数	类型	必传	说明
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • 自定义消息类型（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
isIncludeSender	Int	否	是否向发件人客户端同步已发消息。1 表示同步，详见发件人客户端如何同步已发消息。请注意，该 API 仅支持在发件人已登陆客户端（在线）的情况下同步已发消息。默认值为 0，即不同步。

请求示例

```
POST /message/chatroom/broadcast.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&objectName=RC
sg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

撤回消息

更新时间:2024-08-30

撤回指定消息本质是发送一条撤回命令消息。移动端收到撤回命令消息后，原目标消息将被删除，同时生成一条 `objectName` 是 `RC:RcNtf` 的通知消息，会话界面中可根据 `RC:RcNtf` 展示。

撤回命令消息也会存储到历史消息中，SDK 在获取历史消息时，会获取到撤回命令消息和被撤回的原始消息。移动端 SDK 已内部实现删除逻辑，开发者无需额外处理。

开通服务

使用撤回消息功能无需开通服务。注意，通过服务端撤回消息时，如果撤回操作者的客户端在线，客户端会同步执行该撤回操作。如果操作者客户端长期离线，则下次上线时通过 Server API 撤回的消息可能仍会在发件人客户端展示（收件方不受影响）。建议开通多设备消息同步功能，在发件人客户端再次上线时由 SDK 自动处理。开通方式详见消息管理服务配置。

请求方法

POST： https://数据中心域名/message/recall.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

参数说明

参数	类型	必传	说明
fromUserId	String	是	消息发送人用户 ID。
conversationType	Int	是	会话类型。支持的会话类型包括：`1`（二人会话）、`3`（群组会话）、`4`（聊天室会话）、`6`（系统会话）、`10`（超级群会话）。
targetId	String	是	目标 ID，根据不同的会话类型（`ConversationType`），可能是用户 ID、群组 ID、聊天室 ID、超级群 ID，系统目标 ID。
busChannel	String	否	超级群频道 ID，仅适用于撤回超级群消息。使用要求如下： • 如果发送消息时指定了频道 ID，则撤回时必须指定频道 ID，否则无法撤回。 • 如果发送消息时未指定频道 ID，则撤回时不可指定频道 ID，否则无法撤回。 客户端发送超级群消息时，频道 ID 对应字段名称为 `channelId`。
messageUID	String	是	消息唯一标识。 • 可通过全量消息路由服务获取消息唯一标识，对应名称为 `msgUID`。 • 全量消息路由服务不支持的消息，目前只能通过历史消息日志获取，对应字段名称为 `msgUID`。

参数	类型	必传	说明
sentTime	Long	是	消息发送时间。 • 可通过全量消息路由服务获取消息发送时间，对应名称为 `msgTimestamp`。 • 全量消息路由服务不支持的消息，目前只能通过历史消息日志获取，对应字段名称为 `dateTime`。
isAdmin	Int	否	是否为管理员，默认为 0，设为 1 时，IMKit 收到此条消息后，小灰条默认显示为“管理员 撤回了一条消息”。
isDelete	Int	否	指定移动端接收方是否需要在本地删除原始消息记录及显示撤回消息提示，默认为 `0`。 • 为 `0` 时，移动端接收方仅将原始消息内容替换为撤回提示（小灰条通知），不删除该原始消息记录。 • 为 `1` 时，移动端接收方会删除原始消息记录，不显示撤回提示（小灰条通知）。 注意：即时通讯服务端历史消息不保存已撤回的超级群消息记录。如果 `isDelete` 设置为 `0`，撤回超级群消息后，移动端本地会存有记录（显示为撤回提示），而 Web 端无记录，可能会造成用户体验差异。
disablePush	Boolean	否	是否为静默撤回，默认为 `false`，设为 `true` 时终端用户离线情况下不会收到撤回通知提醒。该字段不支持聊天室、超级群会话类型。
extra	String	否	扩展信息，可以放置任意的数据内容。不支持超级群会话（`conversationType` 为 `10`）。

请求示例

```
POST /message/recall.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

fromUserId=fDR2cVpxxR5zSMUNh3yAwh&targetId=MersNRhaKwJkRV9mJR5JXY&conversationType=1&messageUID=5FGT-7VA9-G4DD-4V5P&sentTime=1507778882124
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

清除消息

更新时间:2024-08-30

如果 App（App Key/环境）开通了单群聊消息云存储服务或聊天室消息云端存储服务，可通过此接口从即时通讯服务端删除历史消息。

- 针对单聊、群聊、系统会话，该接口会为指定用户（`fromUserId`）清除服务端存储的指定时间之前的会话历史消息。清除成功之后，该用户无法再通过历史消息服务获取该会话中早于指定时间的历史消息。
- 针对聊天室会话，该接口会为所有用户清除服务端存储的指定时间之前的会话历史消息。清除成功之后，所有用户均无法再通过历史消息服务获取该会话中早于指定时间的历史消息。请谨慎操作。

> 提示
>
> - 清除历史消息不会影响离线消息收取。如果通过该 API 清除的历史消息中包含客户端离线期间的消息（即客户端未收取过的消息），则客户端再次上线时会自动收取离线消息。
> - 清除历史消息不会影响离线消息补偿机制（该机制仅会在打开多设备消息同步开关后生效）。如果 App 用户卸载重装、换端登录，会触发离线消息补偿机制，默认收取 1 个自然日内（含当天）的消息。如需彻底删除消息补偿，请提交工单，申请开通删除服务端历史消息时同时删除多端补偿的离线消息。

请求方法

POST： https://数据中心域名/conversation/message/history/clean.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
conversationType	String	是	会话类型。支持的会话类型包括：1（单聊会话）、3（群组会话）、4（聊天室会话）、6（系统通知）

参数	类型	必传	说明
fromUserId	String	是	• **单聊、群聊、系统会话**：表示删除指定用户的服务端历史消息。清除成功后，该用户无法从服务端获取已清除的历史消息。 • **聊天室会话**：表示执行删除操作的操作者用户 ID。清除成功后，所有用户均无法从服务端获取已清除的历史消息。
targetId	String	是	需要清除的目标会话 ID。
msgTimestamp	String	否	清除该时间戳之前的所有历史消息，精确到毫秒。为空时清除该会话的所有历史消息。

请求示例

```
POST /conversation/message/history/clean.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

fromUserId=1&targetId=8888&conversationType=3&msgTimestamp=1566281295943
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

单群聊消息扩展概述

更新时间:2024-08-30

单/群聊消息扩展功能可用于为原始消息增加状态标识（扩展数据为 KV 键值对），提供添加、删除、查询扩展信息的接口。

适用场景

对单聊、群聊会话中的原始消息增加状态标识的业务场景均可使用消息扩展。以下是业务场景示例：

- 消息评论。可通过设置原始消息扩展信息的方式添加评论信息。
- 礼物领取、订单状态变化。通过此功能可改变消息显示状态。例如：向用户发送礼物，默认为未领取状态，用户点击后可设置消息扩展为已领取状态。

> 提示
>
> 单/群聊消息扩展接口仅支持单聊、群聊会话类型。
>
> - 聊天室业务不支持消息扩展功能，您无法在聊天室中使用消息扩展。
> - 超级群业务已支持消息扩展功能，并提供 Server API。请另行参见设置超级群消息扩展。

长期存储消息扩展数据

消息扩展操作实际上是通过一条特殊的消息实现的，可认为是“扩展操作消息”。如需保证在长期不登录客户端等情况下仍可获取扩展操作消息，建议同时在 IM 服务管理 页面开通**单群聊消息云端存储**功能。

单群聊消息云端存储功能可保证发送者可在长期不登录客户端、或本地不存储消息等情况下仍可获取到历史消息（包括扩展操作消息）。

- 未开通时，即时通讯服务端默认会保存 7 天的离线消息，供客户端再次上线时收取。例如，假设原始消息已存储在本地，并且离线期间原始消息已经过多次扩展操作，那么客户端再次上线可收取 7 天内的扩展操作消息。**注意**，客户端无法收取早于 7 天的扩展操作消息。
- 开通后，消息将存入服务端历史消息库，默认保存 6 个月。长时间未收取的消息、本地已删除的消息均可通过客户端 SDK 的从服务端拉取历史消息的接口获取。

前往控制台开通单群聊消息云端存储 。生产环境下，**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。

设置单群聊消息扩展

更新时间:2024-08-30

根据消息 ID ，对单聊会话或群聊会话中已经发送的消息设置扩展信息。每次最多可以设置 100 个扩展属性信息，最多可设置 300 个。

通过 Server API 操作消息扩展，默认不会向操作者的客户端同步，会导致扩展信息不一致。如有需要，建议设置 `isSyncSender` 参数为 `1`，并了解使用细节。

前提条件

需要设置消息扩展的原始消息必须在发送时已打开可扩展属性。

例如，通过即时通讯服务端 API 发送消息时，需要设置 `expansion` 为 `true`，该条消息才能支持扩展信息设置。

客户端 SDK 从特定版本开始支持该功能（移动端 4.0.3 版本、Web 端 3.0.7 版本），详见各客户端消息扩展文档。

请求方法

POST https://数据中心域名/message/expansion/set.json

频率限制： 每秒钟限 100 次。**注意**，100 次请求中最多设置群聊消息扩展 20 次。另见已知问题 1。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
userId	String	是	操作者用户 ID，即需要为指定消息（`msgUID`）设置扩展信息的用户 ID。
conversationType	String	是	会话类型。支持的会话类型包括：`1`（二人会话）、`3`（群组会话）。
targetId	String	是	目标 ID，根据不同的 conversationType，可能是用户 ID 或群组 ID。
extraKeyVal	Object	是	消息扩展的内容，JSON 结构，以 Key、Value 的方式进行设置， 如：`{"type":"3"}`。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符`+ = - _`的组合方式，不支持汉字。Value 最大 4096 个字符。单条消息可设置 300 个扩展信息，一次最多可以设置 100 个。
isSyncSender	Int	否	删除操作会生成一条「扩展操作消息」。该字段指定是否将该「扩展操作消息」同步到发件人（扩展操作者）的客户端。`1` 表示同步，默认值为 `0`，即不同步。**注意**，仅设置该参数无法确保发件人客户端一定能获取到该条已发消息，您可能还需要启用其他服务。详见发件人客户端如何同步已发消息。

请求示例

```
POST /message/expansion/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

msgUID=BRGM-DEN2-01E4-
BN66&userId=WNYZbMqpH&targetId=tjw3zbMrU&conversationType=1&extraKeyVal=%7B%22type%22%3A%223%22%7D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

删除单群聊消息扩展

更新时间:2024-08-30

根据消息 ID ，为已发送的单聊或群聊消息删除已设置的扩展信息。

通过 Server API 操作消息扩展，默认不会向操作者的客户端同步，会导致扩展信息不一致。如有需要，建议设置 `isSyncSender` 参数为 `1`，并了解使用细节。

请求方法

POST： https://数据中心域名/message/expansion/delete.json

频率限制： 每秒钟限 100 次。**注意**，100 次请求中最多删除群聊消息扩展 20 次。另见已知问题 1。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
userId	String	是	操作者用户 ID，即需要为指定消息（`msgUID`）删除扩展信息的用户 ID。
conversationType	String	是	会话类型。支持的会话类型包括：`1`（二人会话）、`3`（群组会话）。
targetId	String	是	目标 ID，根据不同的 conversationType，可能是用户 ID 或群组 ID。
extraKey	String	是	需要删除的扩展信息的 Key 值，一次最多可以删除 100 个扩展信息。
isSyncSender	Int	否	删除操作会生成一条「扩展操作消息」。该字段指定是否将该「扩展操作消息」同步到发件人（扩展操作者）的客户端。`1` 表示同步，默认值为 `0`，即不同步。**注意**，仅设置该参数无法确保发件人客户端一定能获取到该条已发消息，您可能还需要启用其他服务。详见发件人客户端如何同步已发消息。

请求示例

```
POST /message/expansion/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

msgUID=BRGM-DEN2-01E4-
BN66&userId=tjw3zbMrU&targetId=WNYZbMqpH&conversationType=1&extraKey=%5B%2211111%22%5D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

获取单群聊消息扩展

更新时间:2024-08-30

根据消息 ID 获取指定消息扩展信息。

请求方法

POST： https://数据中心域名/message/expansion/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
pageNo	Int	否	页数，默认返回 300 个扩展信息。

请求示例

```
POST /message/expansion/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/x-www-form-urlencoded

msgUID=BRGM-DEN2-01E4-BN66
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
extraContent	Object	消息扩展的内容，JSON 结构的 Key、Value 对，如：{"type":"3"}。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符+ = - _ 的组合方式，不支持汉字。Value 最大 4096 个字符。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"extraContent": {
"kkk314": {
"v": "vvv314",
"ts": 33831605613
},
"kkk313": {
"v": "vvv313",
"ts": 33831605588
}
}
}
```

获取历史消息日志

更新时间:2024-08-30

即时通讯服务端可以保存 APP 内所有会话的历史消息记录，历史消息记录以日志文件方式提供，并已经过压缩。您可以使用服务端 API 获取指定 App 的历史消息日志。

服务端 API 接口仅返回消息记录文件下载地址，获取地址后请您自行下载。

开通服务

使用获取历史消息日志功能前，请确认已为当前 **App Key** 开通相关服务。详见消息管理服务配置。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

可获取日志的时间范围

历史消息日志中包含以下会话类型的消息数据：单聊、讨论组、群组、超级群、聊天室、客服、系统通知。

在控制台开通历史消息日志下载服务后，服务端即开始保存当前一小时的消息日志数据。例如您在 10:00 ～ 11:00 之间开通服务，则即时通讯服务端可提供从 10:00 开始的历史消息日志。

> 提示
>
> - 即时通讯服务端每小时打包一次历史消息日志数据，仅支持按小时获取数据。具体请参见 API 接口 `date` 参数的说明。
> - 获取数据有一定延迟。首先，10～11 点的数据文件在 11 点以后才能生成。另外，因文件压缩打包等耗时，一般 1 小时内（即 12 点前）可获取到下载地址。
> - 获取的历史消息日志为指定时间段内所有会话历史消息的全量日志；服务端 API 接口不支持按用户或按会话等方式获取。

日志保存时限

服务端保存的消息记录日志文件仅保留 3 天。3 天后服务端自动删除该日志文件。

如果需要获取图片、视频等文件信息，可通过消息中地址进行下载。如果文件保存在即时通讯服务，文件存储有效期为 6 个月。

您可以通过服务端 API 主动删除历史消息日志。删除接口返回成功后，日志文件会在 10 分钟内被永久删除。详见删除历史消息日志。

日志格式

名称	类型	说明
appId	String	App Key。
fromUserId	String	发送者 ID。
targetId	String	接收者 ID，在消息路由中为 `toUserId`，当发送聊天室广播消息、全量用户落地通知时此字段为空。
targetType	Int	会话类型。`1`（单聊会话）、`2`（讨论组会话）、`3`（群组会话）、`4`（聊天室会话）、`5`（客服会话）、`6`（系统通知）、`7`（应用公众服务）、`8`（公众服务）、`10`（超级群会话）。`targetType` 在 SDK 中为 `ConversationType`。
GroupId	String	根据不同的 `targetType`，可能是讨论组 Id、群组 ID、超级群 ID 或聊天室 ID ，如 `targetType` 为 1 时可忽略 `GroupId`。
busChannel	String	超级群频道 ID。
classname	String	消息类型，例如文本消息 `RC:TxtMsg`、图片消息 `RC:ImgMsg`。详见消息类型概述。
content	String	消息内容。详见消息类型概述 中各类消息内容的结构。
extraContent	Object	消息扩展的内容，JSON 结构的 Key、Value 对，如：`{"type":"3"}`。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符`+ = - _` 的组合方式，不支持汉字。Value 最大 4096 个字符。
dateTime	String	消息时间。
source	String	消息来源，包括：iOS、Android、Websocket、MiniProgram（小程序）、PC、Server。
isDiscard	String	是否被丢弃，`true` 为是，`false` 为否，只针对聊天室会话类型存在。
isSensitiveWord	String	是否含有屏蔽敏感词，`true` 为含有、`false` 为不含有。只针对聊天室会话类型存在。
isForbidden	String	是否为被禁言后发送的消息，只针对聊天室会话类型存在。
isNotForward	String	消息是否不分发，`true` 为不分发、`false` 为分发。只针对聊天室会话类型存在。
msgUID	String	可通过 `msgUID` 确定消息唯一。
groupUserIds	String[]	`targetType` 为 3 时此参数有效，显示为群组中指定接收消息的用户 ID 数组，该条消息为群组定向消息。非定向消息时内容为空，如指定的用户不在群组中内容也为空。

请求方法

POST： https://数据中心域名/message/history.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必填	说明

参数	类型	必填	说明
date	String	是	指定时间，精确到某天某小时，格式为 YYYYMMDDHH。例如 2014010101 表示需要获取 2014 年 1 月 1 日凌晨 1 点至 2 点的数据。 注意：date 的值与应用所属数据中心有关。如您的 App 业务使用新加坡数据中心，则获取消息日志时使用的时间（date），及日志中的消息时间（dateTime）均为 UTC 时间。如您仍需根据北京时间下载数据，请自行转换处理。如要下载北京时间 2019120109 的日志，需要输入 2019120101。

返回参数

返回值	返回类型	说明
code	Int	返回码，200 为正常。
url	String	历史记录下载地址。如没有消息记录数据时，则 url 值为空。
date	String	历史记录时间。

请求示例

```
POST /message/history.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

date=2014010101
```

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"url":"http://aa.com/1/c6720eea-452b-4f93-8159-7af3046611f1.gz",
"date":"2014010101"
}
```

删除历史消息日志

更新时间:2024-08-30

删除服务端保存的 App 下指定某天某小时内的所有会话的历史消息记录日志文件。

- 服务端保存的消息记录日志文件仅保留 3 天。3 天后服务端自动删除该日志文件。
- 如果通过当前 API 主动删除，删除接口返回成功后，日志文件会在 10 分钟内被永久删除。删除后，无法再从服务端获取历史消息日志文件。

> 提示
>
> 该功能仅删除服务端保存的历史消息日志文件。如果 App 已开通单群聊消息云存储功能，删除历史消息日志不影响从服务端保存的历史消息。

请求方法

POST： https://数据中心域名/message/history/delete.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必填	说明
date	String	是	指定时间，精确到某天某小时，格式为 YYYYMMDDHH。例如 `2014010101` 表示需要删除 2014 年 1 月 1 日凌晨 1 点至 2 点的数据。返回成功后，消息记录文件将在随后的 10 分钟内被永久删除。 **注意**：`date` 的值与应用所属数据中心有关。如您的 App 业务使用新加坡数据中心，则获取消息日志时使用的时间（`date`），及日志中的消息时间（`dateTime`）均为 UTC 时间。如您仍需根据北京时间下载数据，请自行转换处理。如要下载北京时间 `2019120109` 的日志，需要输入 `2019120101`。

请求示例

```
POST /message/history/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

date=2014010101
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

全量消息路由

更新时间:2024-08-30

全量消息路由服务支持将单聊、群组、超级群、聊天室、讨论组（废弃）、客服（废弃）的消息数据同步到指定的应用服务器。当接收到的消息为图片、视频类消息时，如果需要获取图片、视频等文件信息，可通过地址进行下载，文件存储有效期为 6 个月。

适用场景举例：

- **存储聊天记录**：可通过开启此服务，实时同步用户发送的消息至应用服务器，由应用服务器进行存储。
- **数据迁移**：实现从第三方通讯云服务的平滑迁移，您的新客户端（集成客户端 SDK）向老客户端（即集成原第三方 SDK）发送消息时，服务端会通过消息路由服务调用原第三方的相应服务端接口，实现向指定老客户端（即集成原第三方 SDK）用户发送消息。详细请查看平滑迁移。

开通服务

使用**全量消息路由**功能前，请确认已为当前 **App Key** 开通相关服务。详见消息管理服务配置。

开通服务时，请配置可正常访问的回调接收地址。如果您的网络有 IP 访问限制，请务必配置 IP 白名单，否则无法正常接收服务端回调。

限制与注意事项说明

- **消息中包含敏感词**：
  - 内置敏感词服务与第三方审核服务（如数美）可设置用于屏蔽消息的敏感词。如果消息因包含此类敏感词导致不下发给收件人，服务端默认也不会将该消息同步到应用服务器。如需要将含有屏蔽敏感词的消息也路由到应用服务器，可提交工单申请开通。
  - 如果在内置敏感词服务的配置中设置了**替换敏感词**，会将消息中的敏感词替换成设置的内容再同步到应用服务器。如需获取原始内容，可参考知识库使用内置敏感词、IM 内容审核、消息回调服务时，如何获取消息原始内容？。
- **Server API 发送的消息默认不进行消息路由**：
  - 通过调用 Server API 接口发送的消息，默认不会通过消息路由服务。如需修改默认行为，请在控制台免费基础功能页面开通 **Server API 发送消息实时路由**服务。
- **单聊状态消息默认不进行消息路由**：单聊会话中的状态类消息，默认不会被同步到应用服务器。包含以下情况：
  - 客户端自动往单聊会话中发送即时通讯服务预定义的状态类消息，例如“正在输入状态”消息，详见消息类型概述。
  - 客户端主动调用发送消息接口发送状态类消息，包括即时通讯服务预定义的状态类消息或客户自定义的状态消息。
  - 通过调用 Server API 接口发送单聊状态消息发送的任何消息。
  - 如果需要上述状态类消息启用消息路由，可提交工单申请开通**单聊状态消息消息路由功能**。注意，如果客户端使用了单聊会话「正在输入的状态」功能，该正在输入的状态消息也会进行路由和消息日志存储，消息量较大请谨慎开通。如果

未开通 Server API 发送消息实时路由服务，则仅对客户端发送的状态类消息启用全量消息路由。

- **始终不支持进行消息路由的情况**：无论是否开通 Server API 发送消息实时路由服务，以下接口发送的消息始终不支持进行消息路由：
  - `/message/online/broadcast.json`，详见在线用户广播。
  - `/message/broadcast.json`，详见全量用户落地通知。
  - `/push.json`，详见标签用户通知、应用包名用户通知、全量用户不落地通知。
  - `/push/user.json`，详见单个用户不落地通知。
- **单群聊消息扩展变更会进行全量消息路由**：
  - 如果变更单群聊消息扩展，服务端会生成一条类型为 `RC:MsgExMsg` 的消息。新消息仅用于消息路由同步，不会同步给客户端 SDK。新消息 ID 由服务端生成，`originalMsgUID` 字段中会携带原消息的 `MsgUID`。
  - 注意，在发消息时即携带的扩展数据无法通过全量消息路由获取。
- **超级群消息扩展变更默认不支持全量消息路由**：
  - 2022.08.11 之后，超级群业务中对消息扩展的变更操作可以同步到您指定的应用服务器。考虑到消息扩展频率较大，可能产生大量消息，因此即时通讯服务端默认不进行全量消息路由。如有需要，请提交工单申请调整超级群消息扩展是否进行路由。开通服务后，如果变更超级群消息扩展，服务端会生成一条类型为 `RC:MsgExMsg` 的消息。新消息仅用于消息路由同步，不会同步给客户端 SDK。新消息 ID 由服务端生成，`originalMsgUID` 字段中会携带原消息的 `MsgUID`。
  - 注意，在发消息时即携带的扩展数据无法通过全量消息路由获取。
- **超级群消息内容修改会进行全量消息路由**：2022.08.11 之后，超级群业务中 App 用户修改的消息内容将会自动同步到应用服务器。用户在超级群中对指定消息进行修改后，服务端会生成一条新消息，同时保留原消息内容与原消息 ID。新消息用于消息路由同步，消息类型同原消息相同。新消息 ID 由服务端生成，不会同步给客户端 SDK。新消息的 `originalMsgUID` 字段中会携带原消息的 `MsgUID`。

回调方法

请求方法： POST

数据格式： `application/x-www-form-urlencoded`

即时通讯服务端会在 POST 请求 URL 中添加签名参数，您可通过签名验证调用者身份和数据有效性，详细参见 服务端回调签名。**注意**，全量消息路由的回调请求路径中同时包含 `timestamp` 与 `signTimestamp` 路径参数。两者值相同，建议仅解析 `signTimestamp` 参数。`signTimestamp` 格式为 Unix 时间戳。

回调正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，包含以下 HTTP 表单参数：

参数	类型	说明
fromUserId	String	发送用户 ID。

参数	类型	说明
toUserId	String	目标 ID，即为客户端 targetId，根据会话类型 channelType 的不同，可能为二人目标 ID、群聊 Id、聊天室 ID、客服 Id 等。
objectName	String	消息类型，例如文本消息 `RC:TxtMsg`、图片消息 `RC:ImgMsg`。参见消息类型概述。
content	String	发送的消息内容。消息内容结构参考用户内容类消息格式或其他内置消息类型的消息内容格式。
channelType	String	会话类型。`PERSON`（二人会话）、`PERSONS`（讨论组会话）、`GROUP`（群组会话）、`TEMPGROUP`（聊天室会话）、`CUSTOMERSERVICE`（客服会话）、`NOTIFY`（系统通知）、`MC`（应用公众服务）、`MP`（公众服务）、`ULTRAGROUP`（超级群服务）。 该字段对应客户端 SDK 中 `ConversationType` 类型：`1`（二人会话）、`2`（讨论组会话）、`3`（群组会话）、`4`（聊天室会话）、`5`（客服会话）、`6`（系统通知）、`7`（应用公众服务）、`8`（公众服务）、`10`（超级群服务）。
msgTimestamp	String	服务端收到客户端发送消息时的服务器时间（1970年到现在的毫秒数）。
msgUID	String	可通过 msgUID 确定消息唯一。
originalMsgUID	String	原始消息 ID，仅针对超级群会话有效。在修改消息、扩展消息、删除消息时，此字段携带有效值。可通过此字段查询原始消息内容。修改超级群消息后，如果需要在服务端撤回消息，必须使用该 ID。
sensitiveType	Int	消息中是否含有敏感信息。`0` 为不包含，`1` 为含有屏蔽敏感词，`2` 为含有替换敏感词。 注意：含有屏蔽敏感词的消息默认不进行路由，可提交工单申请开通。 • 如果未开通（默认），`sensitiveType` 值默认为 `0`，不代表任何意义。 • 如果已开通，可通过该属性判断文本消息中是否含有敏感词。图片、语言消息审核不通过时标识为 `1`。支持单聊、群聊、聊天室会话类型，其他会话类型默认为 `0`，开通后含有屏蔽敏感词的消息也不会进行下发，只会进行消息路由。
source	String	标识消息的发送源头，包括：iOS、Android、Websocket、MiniProgram（小程序）、PC、Server（通过 Server API 发送，需要开通 Server API 发送消息进行消息路由功能）。
busChannel	String	会话频道 ID，使用超级群频道功能时，可通过 `buschannel` 获取频道 ID，未使用时该内容为空。
groupUserIds	String[]	接收群定向消息的群组成员用户 ID 数组。仅群组会话（`channelType` 为 `GROUP`）支持群定向消息，如果指定的用户不在群组中，该字段为空。非群定向消息时，该字段为空。

回调请求示例

假设您在开通服务页面配置的接收地址：`http://example.com/receive_message.php`

```
POST /receive_message.php?
appKey=someappKey&timestamp=1681202504348&signTimestamp=1681202504348&nonce=14314&signature=45beb7cc7307
e711219a47b7cf6a5b000e8 HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
User-Agent: RongCloud/1.0

fromUserId=123&toUserId=456&objectName=RC%3ATxtMsg&content=%7B"content"%3A"hello"%7D&channelType=PERSON&
mestamp=1408710653491&msgUID=596E-P5PG-4FS2-7OJK&groupUserIds=["543","567"]
```

响应回调请求

> 提示

- 只要有 HTTP `200 OK` 成功响应，服务端会认为状态已经同步。
- 如果应答超时 5 秒，服务端会再尝试推送 2 次，如果仍然失败，将不再同步此条状态。
- 如短时间内有大面积超时，将暂时停止请求您的服务器，1 分钟后会继续发送回调请求。异常断网情况下的会延迟 5 分钟同步。

会话置顶

更新时间:2024-08-30

设置会话是否置顶。

服务端会保存会话置顶状态。如果客户端 SDK 4.0.0 及之后版本，通过客户端设置的会话置顶状态会被同步到服务端，在切换设备时可自动同步。

您还可以使用服务端 API 接口进行设置会话是否置顶。设置生效后，最新的置顶状态信息会被自动同步到客户端 SDK。您可以使用最新状态数据更新 UI。

> 提示
>
> 如果客户端 SDK 版本早于 4.0.0 版本，您通过服务端 API 接口设置的会话置顶状态不会被同步到客户端。

请求方法

POST： https://数据中心域名/conversation/top/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，会话所属的用户
conversationType	String	是	会话类型。支持的会话类型包括：1（二人会话）、3（群组会话）、6（系统会话）。
targetId	String	是	需要设置的目标 ID，根据会话类型不同为单聊用户 ID、群聊 ID、系统目标 ID
setTop	String	是	`true` 表示置顶，`false` 表示取消置顶。

请求示例

```
POST /conversation/top/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=XivuFwkcl&targetId=RfqHbcjes&conversationType=1&setTop=true
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送系统通知普通消息

更新时间:2024-08-30

App 下指定用户可以向其他用户发送系统通知消息。单次最多向 100 个用户发送消息。属于落地通知？。

- 支持发送即时通讯服务预定义的消息类型（见消息类型概述）。消息的类型（`objectName` 字段）决定了消息是否支持离线推送通知，以及客户端在收到该消息后，是否展示在聊天界面、会话列表，是否存入本地数据库。
- 支持发送客户自定义类型的消息。客户端在收到自定义消息后，是否展示在聊天界面、会话列表，是否存入本地数据库取决于客户端注册的自定义消息类型定义。
- 系统通知消息只能通过即时通讯服务端 API 进行发送，会话类型为 SYSTEM。该类型的会话不支持终端用户在收到消息后进行回复。

例如，对于一般的 App 业务通知来说，假设发送一条文本消息（objectName 为 RC:TxtMsg，属于客户端 SDK 会存储的内置消息类型，且可触发离线推送），效果如下：

- 客户端如果在线，会即时收到一条消息，所在会话的 Target ID 为调用接口时传入的 `fromUserId`，会话类型为系统会话（类型为 SYSTEM）。
- 客户端如果离线，消息会触发服务端远程推送，客户端如已集成并启用推送服务，则会收到推送通知。

请求方法

POST： https://数据中心域名/message/system/publish.json

频率限制： 每秒钟限发送 100 条消息，每次最多同时向 100 人发送，如：一次发送 100 人时，视为 100 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
toUserId	String	是	接收用户 ID，提供多个本参数可以实现向多用户发送系统消息，上限为 100 人。
objectName	String	是	接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。

参数	类型	必传	说明
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见消息类型概述中各内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • 自定义消息类型（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
pushContent	String	否	指定收件人离线时触发的远程推送通知中的通知内容。注意：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。 • 如果消息类型（objectName 字段）为即时通讯服务预定义消息类型中的用户内容类消息格式，可不填写该字段，远程推送通知默认使用服务端预置的推送通知内容。 • 如果消息类型（objectName 字段）为即时通讯服务预定义消息类型中通知类、信令类（"撤回命令消息" 除外），且需要支持远程推送通知，则必须填写 pushContent，否则收件人不在线时无法收到远程推送通知。如无需触发远程推送，可不填该字段。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 pushContent 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 pushContent 字段留空禁用远程推送通知。
pushData	String	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 appData（提示：rc 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 appData。
isPersisted	Int	否	是否需要为收件人在历史消息云端存储服务中存储此条消息。0 表示不存储；1 表示存储。默认值为 1，存储（系统消息是否存入服务端历史消息也依赖单群聊消息云端存储服务）。此属性不影响离线消息功能，用户未在线时都会转为离线消息存储。 一般情况下（第 1、2 种情况），客户端是否存储消息不依赖此参数。以下第 3 种情况属于例外： 1. 如果消息属于内置消息类型，客户端 SDK 会根据消息类型本身的存储属性标识判断是否存入本地数据库。详见消息类型概述。 2. 如果消息属于自定义消息类型，则客户端 SDK 会根据该类型在客户端上注册时的存储属性标识判断是否需要存入本地数据库。 3. 如果消息属于客户端 App 上未注册自定义消息类型（例如客户端使用的 App 版本过旧），则客户端 SDK 会根据当前参数值确定是否将消息存储在本地。但因消息类型未注册，客户端无法解析显示该消息。
contentAvailable	Int	否	针对 iOS 平台，对 SDK 处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
disablePush	Boolean	否	是否为静默消息，默认为 false，设为 true 时终端用户离线情况下不会收到通知提醒。

参数	类型	必传	说明
pushExt	String	否	配置消息的推送通知，如推送通知的标题等。`disablePush` 属性为 `true` 时此属性无效。具体请查看下方 `pushExt` 参数说明。

- `pushExt` 参数说明

pushExt 参数	类型	必传	说明
title	String	否	通知栏显示标题，最长不超过 50 个字符，默认情况下通知标题单聊会话显示用户名称，群聊会话显示群名称。
templateId	String	否	推送模板 ID，设置后根据目标用户通过 SDK 设置的语言环境，匹配模板中设置的语言内容进行推送，未匹配成功时使用默认内容进行推送，模板内容在“控制台-自定义推送文案”中进行设置。详情请查看自定义多语言推送模板。
forceShowPushContent	Number	否	是否越过客户端配置，强制在推送通知内显示通知内容（`pushContent`）。默认值 `0` 表示不强制，`1` 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置`forceShowPushContent` 为 `1` 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
pushConfigs	Array	否	按厂商设置不同推送属性。支持的推送通道值为 `MI`（小米）、`HONOR`（荣耀）、`HW`（华为）、`OPPO`、`VIVO`、`APNs`、`FCM`。
pushConfigs.HONOR.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
pushConfigs.HONOR.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.channelId	String	否	华为推送通知渠道的 ID，详细请参见 华为自定义通知渠道。更多通知渠道信息，请参见 Android 官方文档。

pushExt 参数	类型	必传	说明
pushConfigs.HW.importance	String	否	华为通知栏消息优先级，取值： • NORMAL（默认值，重要信息） • LOW
pushConfigs.HW.image	String	否	华为推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
pushConfigs.MI.channelId	String	否	小米推送通知渠道的 ID，详细请参见 小米推送消息分类新规。
pushConfigs.MI.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息的右侧图标 URL，若不设置，则不展示通知栏右侧图标。 • 国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。 • 图片要求：大小 120 * 120px，格式为 png 或者 jpg 格式。
pushConfigs.OPPO.channelId	String	否	OPPO 推送通知渠道的 ID，详细请参见 OPPO PUSH 通道升级说明。
pushConfigs.VIVO.classification	Number	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
pushConfigs.VIVO.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category`，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。

pushExt 参数	类型	必传	说明
pushConfigs.APNs.thread-id	String	否	iOS 平台通知栏分组 ID。相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示。
pushConfigs.APNs.apns-collapse-id	String	否	适用于 iOS 平台。设置后设备收到有相同 ID 的消息，会合并成一条。 仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.richMediaUri	String	否	适用于 iOS 平台。iOS 推送自定义的通知栏消息右侧图标 URL，需要 App 自行解析 `richMediaUri` 并实现展示。仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
pushConfigs.FCM.channelId	String	否	FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见安卓官方文档。
pushConfigs.FCM.collapse_key	String	否	适用于 Android 平台。可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
pushConfigs.FCM.imageUrl	String	否	适用于 Android 平台。FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。

- `pushExt` 结构示例

```
{
"title":"you have a new message.",
"templateId":"123456",
"forceShowPushContent":0,
"pushConfigs": [
{
"HW": {
"channelId": "hw-123",
"importance": "NORMAL",
"image":"https://example.com/image.png"
}
},
{
"MI": {
"channelId": "mi-123",
"large_icon_uri":"https://example.com/image.png"
}
},
{
"OPPO": {"channelId": "oppo-123"}
},
{
"VIVO" : {"classification": "0"}
},
{
"APNs": {
"thread-id": "123",
"apns-collapse-id":"123456",
"richMediaUri":"https://example.com/image.png"
}
},
{
"FCM": {
"channelId":"rongcloud_channelid",
"collapse_key":"1234",
"imageUrl":"https://example.com/image.png"
}
}
]
}
```

请求示例

```
POST /message/system/publish.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&toUserId=2191
erId=2192&objectName=RC:TxtMsg&pushContent=thisisapush&pushData=hello
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送系统通知模板消息

更新时间:2024-08-30

App 下指定用户可以向其他用户发送系统通知模板消息。单次最多向 100 个用户发送消息。属于落地通知[?]。

- 支持发送即时通讯服务预定义的消息类型（见消息类型概述）。消息的类型（`objectName` 字段）决定了消息是否支持离线推送通知，以及客户端在收到该消息后，是否展示在聊天界面、会话列表，是否存入本地数据库。
- 支持发送客户自定义类型的消息。客户端在收到自定义消息后，是否展示在聊天界面、会话列表，是否存入本地数据库取决于客户端注册的自定义消息类型定义。
- 系统通知消息只能通过即时通讯服务端 API 进行发送，会话类型为 SYSTEM。该类型的会话不支持终端用户在收到消息后进行回复。
- 消息的 `content`、`pushContent`、`pushData` 字段可以通过模板控制。

例如，对于一般的 App 业务通知来说，假设发送一条文本消息（objectName 为 RC:TxtMsg，属于客户端 SDK 会存储的内置消息类型，且可触发离线推送），效果如下：

- 客户端如果在线，会即时收到一条消息，所在会话的 Target ID 为调用接口时传入的 `fromUserId`，会话类型为系统会话（类型为 SYSTEM）。
- 客户端如果离线，消息会触发服务端远程推送，客户端如已集成并启用推送服务，则会收到推送通知。

如何配置与使用消息模板

消息模板通过字段内容模板（带标识位），和按接收者提供的标识位定义，实现单次向多个接收者发送不同内容的效果。

定义内容模板

发送消息时，可在 content、pushContent、pushData 字段中传入带标识位的字段内容，例如：

```
"toUserId":["21","22","23"],
"content":"{\"content\":\"{c}{d}\",\"extra\":\"bb\"}",
"pushContent":["hello {c}","hello {c}","hello {c}"],
```

上例中的 {c}、{d} 为自定义的标识位。

- 当前支持定义模板的字段包括：`content`、`pushContent`、`pushData`。
- 消息内容（`content`）字段仅支持插入一个模板，所有接收者共用该模板。
- 消息推送通知内容（`pushContent`）、推送附加数据（`pushData`）字段类型为数组，必须按照 `toUserId` 中的用户 ID 列表逐个定义模板。即使不在 `pushContent`、`pushData` 中使用模板标识位，或所有接收者接收相同内容，都必须按照 `toUserId` 中的用户 ID 列表逐个定义各自的 `pushContent`、`pushData`。

指定模板内容标识位的值

按照收件人列表（toUserId）在 values 字段提供标识位的定义，即为各个收件人指定需要接收的个性化内容。

```
"values":[
{
"{c}":"Tom",
"{d}":"：2"
},
{
"{c}":"Jerry",
"{d}":"：5"},
{
"{c}":"Rose",
"{d}":"：10"}
],
```

上面示例中，各接收者在线时收到的消息内容如下：

- 用户 ID 为 21 收到：Tom：2
- 用户 ID 为 22 收到：Jerry：5
- 用户 ID 为 23 收到：Rose：10

如果接收者离线，各自收到的推送通知的内容（pushContent）也不同，分别为 Hello Tom，Hello Jerry，Hello Rose。

> 提示
>
> 如 content 中定义了标识 {d} ，则在 values 中必须设置 {d} 的值，否则 {d} 会以文本方式随消息发送给用户。

请求方法

POST： https://数据中心域名/message/system/publish_template.json

频率限制： 每秒钟限 100 次。请注意，如果同时向 100 人发送消息，视为 100 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/json，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
toUserId	String[]	是	接收用户 ID，提供多个本参数可以实现向多人发送消息，上限为 100 人。

参数	类型	必传	说明
objectName	String	是	接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
values	Array of objects	是	为消息内容（content）、推送通知内容（pushContent）、推送数据（pushData）中的标识位（标识位示例：{d}）提供对应的值。
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见消息类型概述中各内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • 自定义消息类型（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
pushContent	String[]	否	指定收件人离线时触发的远程推送通知中的通知内容。注意：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。支持定义模板标识位，使用 values 中的值进行替换。 • 如果消息类型（objectName 字段）为即时通讯服务预定义的消息类型，填写该字段后，离线推送通知中显示模板定义的推送内容，而非消息类型的默认推送内容。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 pushContent 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 pushContent 字段对应数组传空值禁用离线推送。
pushData	String[]	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 appData（提示：rc 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 appData。
contentAvailable	Int	否	针对 iOS 平台，对 SDK 处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
disablePush	Boolean	否	是否为静默消息，默认为 false，设为 true 时终端用户离线情况下不会收到通知提醒。

请求示例

```
POST /message/system/publish_template.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/json

{
"fromUserId":"fromuser",
"objectName":"RC:TxtMsg",
"content":"{\"content\":\"{c}{d}{e}\",\"extra\":\"bb\"}",
"toUserId":["21","22"],
"values":[{"{c}":"1","{d}":"2","{e}":"3"},{"{c}":"4","{d}":"5","{e}":"6"}],
"pushContent":["push{c}","push{c}"],
"pushData":["pushd","pushd{e}"]
}
```

上面示例中用户 ID 为 21 的用户，收到信息为 123，上面示例中用户 ID 为 22 的用户，收到信息为 456。

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

撤回单条系统通知

更新时间:2024-08-30

撤回指定消息本质是发送一条撤回命令消息。移动端收到撤回命令消息后，原目标消息将被删除，同时生成一条 objectName 是 RC:RcNtf 的通知消息，会话界面中可根据 RC:RcNtf 展示。

撤回命令消息也会存储到历史消息中，SDK 在获取历史消息时，会获取到撤回命令消息和被撤回的原始消息。移动端 SDK 已内部实现删除逻辑，开发者无需额外处理。

接口说明

请使用撤回消息接口撤回系统通知消息。**注意**，输入参数中的 conversationType 为 6（系统会话）。

```
POST /message/recall.json
```

以上接口的详细使用说明请参见消息撤回。

支持撤回的系统通知

以下接口发送的系统会话消息支持撤回：

- `/message/system/publish.json`，详见发送系统通知普通消息。
- `/message/system/publish_template.json`，详见发送系统通知模板消息。

全量用户通知服务配置

更新时间:2024-08-30

即时通讯服务提供**全量用户通知服务**，允许 App 向应用中所有或指定用户发送消息或推送，实现将活动预告、内容动态精准及时的发送给目标用户的运营需求。

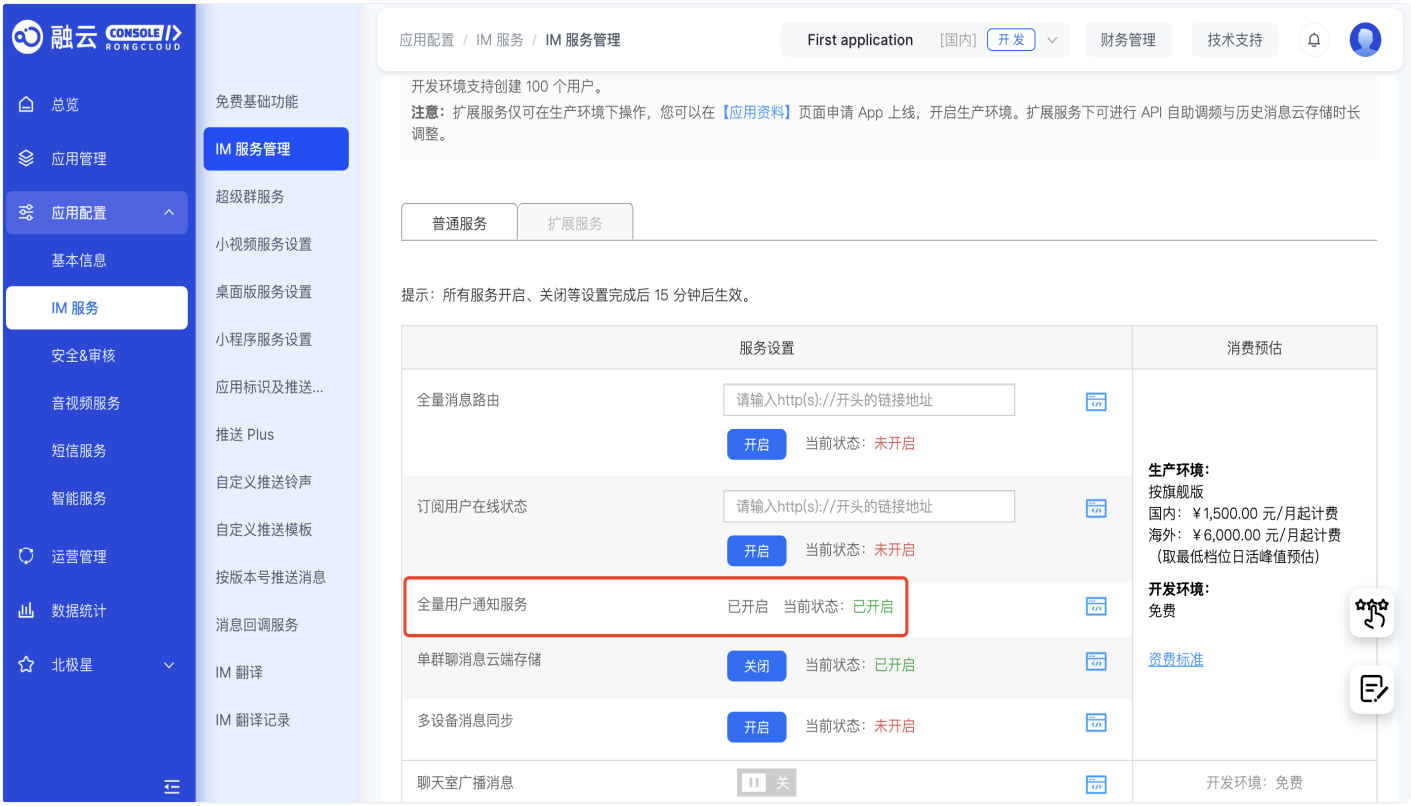
服务能力及相关 API 接口

在使用即时通讯服务端的以下能力及 API 接口前，必须启用**全量用户通知服务**。

名称	接口
发送全量用户落地通知	`/message/broadcast.json` 接口用于 **App 用户**向全部用户发送系统通知消息的场景，本质上是指向 App 中所有用户广播一条系统会话消息，在客户端使用**定时拉取**机制接收。发送消息后，最长 3 分钟内可收到此条消息。属于落地通知。
发送在线用户广播	`/message/online/broadcast.json` 接口用于 **App 用户**向当前在线用户发送系统通知消息的场景，本质上是指向 App 中所有在线用户广播一条系统会话消息，在客户端使用**定时拉取**机制接收。发送消息后，最长 3 分钟内可收到此条消息。属于落地通知。
发送全量用户不落地通知	`/push.json` 用于 **App 业务端**向全部用户或指定用户发送「推送通知」，本质上是直接通过第三方或即时通讯服务自建的推送通道向 App 中所有用户推送一条通知。属于不落地通知。
标签用户通知	`/push.json` 用于 **App 业务端**向携带指定标签的用户发送「推送通知」及系统会话消息，本质上是直接通过第三方或即时通讯服务自建的推送通道向 App 中所有用户推送一条通知。发送的数据携带消息类型和消息内容，属于落地通知。
应用包名通知	`/push.json` 用于 **App 业务端**向指定应用包名的用户发送「推送通知」及系统会话消息，本质上是直接通过第三方或即时通讯服务自建的推送通道向 App 中所有用户推送一条通知。发送的数据携带消息类型和消息内容，属于落地通知。

开通服务

- 在开发环境下，默认已开启**全量用户通知服务**开关，可免费使用。
- 在生产环境下，可以在控制台 IM 服务管理 页面的**普通服务**标签下开通**全量用户通知服务**。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。



开通云存储服务

如果需要即时通讯服务端长期存储系统会话消息，请开通对应的云存储服务。开通成功后，客户端可通过获取远端历史消息的 API 获取服务端存储的历史消息记录。

名称	接口	服务端消息存储服务
发送全量用户落地通知	/message/broadcast.json	默认不存入即时通讯服务端历史消息记录。如需存储，请提交工单 申请开通广播消息云存储。
发送在线用户广播	/message/online/broadcast.json	不支持存入服务端历史消息记录。
发送全量用户不落地通知	/push.json	仅产生推送通知，不产生消息，不支持在服务端存储。
标签用户通知	/push.json	仅依赖开通单群聊消息云存储服务。您可以前往控制台 IM 服务管理 页面，在普通服务标签下开通服务。
应用包名通知	/push.json	仅依赖开通单群聊消息云存储服务。您可以前往控制台 IM 服务管理 页面，在普通服务标签下开通服务。

发送全量用户落地通知

更新时间:2024-08-30

App 用户可以向 App 下全部用户发送系统会话消息，该功能称为「全量用户落地通知」。

- 支持发送即时通讯服务预定义的消息类型（见消息类型概述）。消息的类型（`objectName` 字段）决定了消息是否支持离线推送通知，以及客户端在收到该消息后，是否展示在聊天界面、会话列表，是否存入本地数据库。
- 支持发送客户自定义类型的消息。客户端在收到自定义消息后，是否展示在聊天界面、会话列表，是否存入本地数据库取决于客户端注册的自定义消息类型定义。
- 只能通过即时通讯服务端 API 进行发送，会话类型为 SYSTEM。该类型的会话不支持终端用户在收到消息后进行回复。
- 全量用户通知消息默认不存入即时通讯服务端历史消息记录。因此，客户端如果获取服务端的系统会话历史消息记录，其中不包含全量落地通知的消息。如需存储，参见全量用户通知服务配置。

例如，对于一般的 App 业务通知来说，假设发送一条文本消息（objectName 为 RC:TxtMsg，属于客户端 SDK 会存储的内置消息类型，且可触发离线推送），效果如下：

- 客户端如果在线，会即时收到一条消息，所在会话的 Target ID 为调用接口时传入的 `fromUserId`，会话类型为系统会话（类型为 SYSTEM）。
- 客户端如果离线，消息会触发服务端远程推送，客户端如已集成并启用推送服务，则会收到推送通知。

> 提示
>
> 广播消息在客户端使用定时拉取机制接收。发送消息后，最长 3 分钟内可收到此条消息。

开通服务

使用全量用户落地通知功能前，请确认已为当前 App Key 开通相关服务。详见全量用户通知服务配置。

如未开通服务，Server API 将返回 1009 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 1008）。

请求方法

POST： https://数据中心域名/message/broadcast.json

频率限制： 每小时限发送 2 次，每天（自然日）最多发送 3 次。即时通讯服务端的消息处理默认为每秒钟 2000 条，即每秒钟可以向 2000 个目标用户下发消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见消息类型概述中各内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • 自定义消息类型（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
pushContent	String	否	指定收件人离线时触发的远程推送通知中的通知内容。注意：对于部分消息类型，该字段是否有值决定了是否触发远程推送通知。 • 如果消息类型（objectName 字段）为即时通讯服务预定义消息类型中的用户内容类消息格式，可不填写该字段，远程推送通知默认使用服务端预置的推送通知内容。 • 如果消息类型（objectName 字段）为即时通讯服务预定义消息类型中通知类、信令类（"撤回命令消息" 除外），且需要支持远程推送通知，则必须填写 pushContent，否则收件人不在线时无法收到远程推送通知。如无需触发远程推送，可不填该字段。 • 如果消息类型为自定义消息类型，且需要支持远程推送通知，则必须填写 pushContent 字段，否则收件人不在线时无法收到远程推送通知。 • 如果消息类型为自定义消息类型，但不需要支持远程推送通知（例如通过自定义消息类型实现的 App 业务层操作指令），可将 pushContent 字段留空禁用远程推送通知。
pushData	String	否	iOS 平台收到推送消息时，可从 payload 中获取 APNs 推送数据，对应字段名为 appData（提示：rc 字段中默认携带了消息基本信息）。Android 平台收到推送消息时对应字段名为 appData。
contentAvailable	Int	否	针对 iOS 平台，对 SDK 处于后台暂停状态时为静默推送，是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
pushExt	String	否	配置消息的推送通知，如推送通知的标题等。具体请查看下方 pushExt 参数说明。

- pushExt 参数说明

pushExt 参数支持设置消息推送通知的标题、推送内容模板、是否强制通知及推送 ChannelID 等。pushExt 为 JSON 结构请求时需要做转义处理。

pushExt 参数	类型	必传	说明
title	String	否	通知栏显示标题，最长不超过 50 个字符，默认情况下通知标题单聊会话显示用户名称，群聊会话显示群名称。
templateId	String	否	推送模板 ID，设置后根据目标用户通过 SDK 设置的语言环境，匹配模板中设置的语言内容进行推送，未匹配成功时使用默认内容进行推送，模板内容在“控制台-自定义推送文案”中进行设置。详情请查看自定义多语言推送模板。
forceShowPushContent	Number	否	是否越过客户端配置，强制在推送通知内显示通知内容（pushContent）。默认值 0 表示不强制，1 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置forceShowPushContent 为 1 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
pushConfigs	Array	否	按厂商设置不同推送属性。支持的推送通道值为 MI（小米）、HONOR（荣耀）、HW（华为）、OPPO、VIVO、APNs、FCM。
pushConfigs.HONOR.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
pushConfigs.HONOR.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.channelId	String	否	华为推送通知渠道的 ID，详细请参见 华为自定义通知渠道。更多通知渠道信息，请参见 Android 官方文档。
pushConfigs.HW.importance	String	否	华为通知栏消息优先级，取值： • NORMAL（默认值，重要信息） • LOW

pushExt 参数	类型	必传	说明
pushConfigs.HW.image	String	否	华为推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
pushConfigs.HW.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
pushConfigs.MI.channelId	String	否	小米推送通知渠道的 ID，详细请参见 小米推送消息分类新规。
pushConfigs.MI.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息的右侧图标 URL，若不设置，则不展示通知栏右侧图标。 • 国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。 • 图片要求：大小 120 * 120px，格式为 png 或者 jpg 格式。
pushConfigs.OPPO.channelId	String	否	OPPO 推送通知渠道的 ID，详细请参见 OPPO PUSH 通道升级说明。
pushConfigs.VIVO.classification	Number	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
pushConfigs.VIVO.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category`，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
pushConfigs.APNs.thread-id	String	否	iOS 平台通知栏分组 ID。相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示。

pushExt 参数	类型	必传	说明
pushConfigs.APNs.apns-collapse-id	String	否	适用于 iOS 平台。设置后设备收到有相同 ID 的消息，会合并成一条。 仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.richMediaUri	String	否	适用于 iOS 平台。iOS 推送自定义的通知栏消息右侧图标 URL，需要 App 自行解析 `richMediaUri` 并实现展示。仅支持 iOS 10.0 及以上版本。
pushConfigs.APNs.interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
pushConfigs.FCM.channelId	String	否	FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见安卓官方文档。
pushConfigs.FCM.collapse_key	String	否	适用于 Android 平台。可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
pushConfigs.FCM.imageUrl	String	否	适用于 Android 平台。FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。

- `pushExt` 结构示例

```
{
"title":"you have a new message.",
"templateId":"123456",
"forceShowPushContent":0,
"pushConfigs": [
{
"HW": {
"channelId": "hw-123",
"importance": "NORMAL",
"image":"https://example.com/image.png"
}
},
{
"MI": {
"channelId": "mi-123",
"large_icon_uri":"https://example.com/image.png"
}
},
{
"OPPO": {"channelId": "oppo-123"}
},
{
"VIVO" : {"classification": "0"}
},
{
"APNs": {
"thread-id": "123",
"apns-collapse-id":"123456",
"richMediaUri":"https://example.com/image.png"
}
},
{
"FCM": {
"channelId":"rongcloud_channelid",
"collapse_key":"1234",
"imageUrl":"https://example.com/image.png"
}
}
]
}
```

请求示例

```
POST /message/broadcast.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&objectName=RC
sg&pushContent=thisisapush&pushData=%7B%22pushData%22%3A%22hello%22%7D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送在线用户广播

更新时间:2024-08-30

App 用户可以向 App 下当前在线的所有用户发送系统会话消息，该功能称为「在线用户广播」。

- 支持发送即时通讯服务预定义的消息类型（见消息类型概述）。消息的类型（`objectName` 字段）决定了客户端在收到该消息后，是否展示在聊天界面、会话列表，是否存入本地数据库。
- 支持发送客户自定义类型的消息。客户端在收到自定义消息后，是否展示在聊天界面、会话列表，是否存入本地数据库取决于客户端注册的自定义消息类型定义。
- 只能通过即时通讯服务端 API 进行发送，会话类型为 SYSTEM。该类型的会话不支持终端用户在收到消息后进行回复。

> 提示
>
> - 广播消息在客户端使用定时拉取机制接收。发送消息后，最长 3 分钟内 SDK 会拉取到此条消息。
> - 处于离线状态的客户端如果在 10 分钟内上线，可以收到广播消息。如果在发送消息后 10 分钟内一直处于离线状态，则无法收到该消息。
> - 在线广播消息暂不支持撤回。

例如，App 业务端希望向当前时间点正在使用 App 的所有用户发送一条消息。假设发送一条文本消息（objectName 为 RC:TxtMsg，属于客户端 SDK 会存储的内置消息类型，且可触发离线推送），效果如下：

- 在线用户即时收到一条消息，所在会话的 Target ID 为调用接口时传入的 `fromUserId`，会话类型为系统会话（类型为 SYSTEM）。
- 离线用户在 10 分钟内上线，仍可以收到广播消息。如果在发送消息后 10 分钟内用户一直处于离线状态，则无法收到该消息。

开通服务

使用在线用户广播功能前，请确认已为当前 **App Key** 开通相关服务。详见全量用户通知服务配置。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/message/online/broadcast.json

频率限制： 每分钟限 60 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	发送人用户 ID。
objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 "RC:" 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见消息类型概述中各内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 {"content":"Hello world!"} 序列化后的结果作为此处 content 字段的值。 • 自定义消息类型（objectName 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。

请求示例

```
POST /message/online/broadcast.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&fromUserId=2191&objectName=RC
sg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送全量用户不落地通知

更新时间:2024-08-30

App 业务端可以向全部用户或指定用户群体发送「推送通知」，本质上是直接通过第三方或即时通讯服务自建的推送通道向 App 中所有用户推送一条通知。该功能只触发推送，不发送消息，不产生会话，称为「全量用户不落地通知」。

该功能使用 `/push.json` 接口。

- 只能通过即时通讯服务端 API 进行发送。
- 「推送通知」的所有内容仅展现在通知栏。无论客户端 App 是否在前台，始终以通知形式展示在系统通知栏中。该功能不发送消息，不产生会话，因此用户无法在任何聊天会话中看到不落地通知的内容。
- 客户端不会存储不落地通知。
- 始终使用推送通道，因此不受客户端与即时通讯服务端之间的连接状态的影响。即使客户端不在前台、不在线（例如 App 被杀死），只要可正常接收推送，就可以收到不落地通知。
- 通过 `audience` 字段可以控制推送条件，支持按用户标签（`tag`）、用户 ID（`userid`）、应用包名（`packageName`）、全量用户（`is_to_all`）。其中用户标签需要 App 进行设置，详见设置用户标签、批量设置用户标签。

> 提示
>
> - 即时通讯服务将 30 天内连接过 IM 服务的设备作为推送的目标。30 天内未打开过应用的设备，被视为应用已被卸载。
> - 如果客户端设备不允许推送（断开连接时设置不允许推送），或遇到其他原因导致推送失败，则无法接收到不落地通知。

开通服务

使用全量用户不落地通知功能前，请确认已为当前 **App Key** 开通相关服务。详见全量用户通知服务配置。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/push.json

频率限制： 共享 `/push.json` 的限频配额，即每小时限发送 2 次，每天（自然日）最多发送 3 次。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/json，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
platform	String[]	是	目标操作系统，iOS、Android 最少传递一个。如果需要给两个系统推送消息时，则需要全部填写。
audience	Object	是	推送条件，包括：tag、userid 、packageName 、 is_to_all。
audience.tag	String[]	否	用户标签。每次发送时最多发送 20 个标签，标签之间为 AND 的关系，is_to_all 为 true 时参数无效。
audience.tag_or	String[]	否	用户标签。每次发送时最多发送 20 个标签，标签之间为 OR 的关系，is_to_all 为 true 时参数无效，tag_or 同 tag 参数可以同时存在。
audience.userid	String[]	否	用户 ID，每次发送时最多发送 1000 个用户，如果 tag 和 userid 两个条件同时存在时，则以 userid 为准，如果 userid 有值时，则 platform 参数无效，is_to_all 为 true 时参数无效。
audience.packageName	String	否	应用包名，is_to_all 为 true 时，此参数无效。与 tag、tag_or 同时存在时为 And 的关系，向同时满足条件的用户推送。与 userid 条件同时存在时，以 userid 为准进行推送。
audience.is_to_all	Boolean	是	是否全部推送，false 表示按 tag 、tag_or 或 userid 条件推送，true 表示向所有用户推送，tag、tag_or 和 userid 条件无效。
notification	Object	是	按操作系统类型推送消息内容，如 platform 中设置了给 iOS 和 Android 系统推送消息，而在 notification 中只设置了 iOS 的推送内容，则 Android 的推送内容为最初 alert 设置的内容，详细查看 notification 参数结构说明。
notification.title	String	否	通知栏显示标题，最长不超过 50 个字符。
notification.forceShowPushContent	Int	否	是否越过客户端配置，强制在推送通知内显示通知内容（pushContent）。默认值 0 表示不强制，1 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置forceShowPushContent 为 1 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
notification.alert	String	否	默认推送通知内容。如填写了 iOS 或 Android 下的 alert 时，则推送内容以对应平台系统的 alert 为准。
notification.ios	Object	否	设置 iOS 平台下的推送及附加信息。详细查看 ios 参数结构说明。
notification.android	Object	否	设置 Android 平台下的推送及附加信息。详细查看 android 参数结构说明。

- notification.ios 参数结构说明

参数	类型	必传	说明
title	String	否	通知栏显示的推送标题，仅针对 iOS 平台，支持 iOS 8.2 及以上版本。该属性优先级高于 notification.title。
contentAvailable	Int	否	针对 iOS 平台，静默推送是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
alert	String	否	推送消息内容，传入后默认的推送消息内容失效，不能为空。

参数	类型	必传	说明
badge	int	否	应用角标，仅针对 iOS 平台；不填时，表示不改变角标数；为 0 或负数时，表示 App 角标上的数字清零；否则传相应数字表示把角标数改为指定的数字，最大不超过 9999，参数在 `ios` 节点下设置，详细可参考“设置 iOS 角标数 HTTP 请求示例”。
thread-id	String	否	iOS 平台通知栏分组 ID，相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示
apns-collapse-id	String	否	iOS 平台，从 iOS10 开始支持，设置后设备收到有相同 ID 的消息，会合并成一条
category	String	否	iOS 富文本推送的类型开发者自己定义，自己在 App 端进行解析判断，与 richMediaUri 一起使用，当设置 category 后，推送时默认携带 mutable-content 进行推送，属性值为 1。
richMediaUri	String	否	iOS 富文本推送内容的 URL，与 category 一起使用。
interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的 “定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

- `notification.android` 结构说明

参数	类型	必传	说明
alert	String	否	Android 平台下推送消息内容，传入后默认的推送消息内容失效。
honor.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
honor.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。

参数	类型	必传	说明
hw.channelId	String	否	华为推送通知渠道的 ID。详见自定义通知渠道。
hw.importance	String	否	华为推送通知栏消息优先级，取值 NORMAL、LOW，默认为 NORMAL 重要消息。
hw.image	String	否	华为推送自定义的通知栏消息右侧大图标 URL，如果不设置，则不展示通知栏右侧图标。URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp，超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
mi.channelId	String	否	小米推送通知渠道的 ID。详见小米推送消息分类新规。
mi.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。图片要求：大小120 * 120px，格式为 png 或者 jpg 格式。
oppo.channelId	String	否	oppo 推送通知渠道的 ID。详见推送私信通道申请。
vivo.classification	String	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
vivo.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category` ，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
fcm.channelId	String	否	Google FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见 Android 官方文档。
fcm.collapse_key	String	否	Google FCM 推送中可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。

参数	类型	必传	说明
fcm.imageUrl	String	否	Google FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

请求示例

```
POST /push.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{
"platform":["ios","android"],
"audience":{
"tag":["女","年轻"],
"tag_or":["北京","上海"],
"userid":["123","456"],
"is_to_all":true
},
"notification":{
"title":"标题",
"forceShowPushContent":0,
"alert":"this is a push",
"ios":
{
"alert": "override alert",
"thread-id":"223",
"apns-collapse-id":"111",
"extras": {"id": "1","name": "2"}
},
"android": {
"alert": "override alert",
"hw":{
"channelId":"NotificationKanong",
"importance": "NORMAL",
"image":"https://example.com/image.png"
},
"mi":{
"channelId":"rongcloud_kanong",
"large_icon_uri":"https://example.com/image.png"
},
"oppo":{
"channelId":"rc_notification_id"
},
"vivo":{
"classification":"0"
},
"extras": {"id": "1","name": "2"}
}
}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
id	String	推送唯一标识。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"id":"50whSR6kQiHb7YgFwQzXIb"}
```

发送标签用户通知

更新时间:2024-08-30

App 用户可以向 App 下携带指定标签的用户发送系统会话消息，该功能称为「标签用户通知」。

该功能使用 /push.json 接口。

- 用户标签需要通过设置用户标签、批量设置用户标签 接口进行添加或移除。
- 支持发送即时通讯服务预定义的消息类型（见消息类型概述）。消息的类型（objectName 字段）决定了消息是否支持离线推送通知，以及客户端在收到该消息后，是否展示在聊天界面、会话列表，是否存入本地数据库。
- 支持发送客户自定义类型的消息。客户端在收到自定义消息后，是否展示在聊天界面、会话列表，是否存入本地数据库取决于客户端注册的自定义消息类型定义。
- 只能通过即时通讯服务端 API 进行发送，会话类型为 SYSTEM。该类型的会话不支持终端用户在收到消息后进行回复。

例如，对于一般的 App 业务通知来说，假设发送一条文本消息（objectName 为 RC:TxtMsg，属于客户端 SDK 会存储的内置消息类型，且可触发离线推送），效果如下：

- 客户端如果在线，会即时收到一条消息，所在会话的 Target ID 为调用接口时传入的 fromUserId，会话类型为系统会话（类型为 SYSTEM）。
- 客户端如果离线，消息会触发服务端远程推送，客户端如已集成并启用推送服务，则会收到推送通知。

开通服务

使用标签用户通知功能前，请确认已为当前 **App Key** 开通相关服务。详见全量用户通知服务配置。

如未开通服务，Server API 将返回 1009 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 1008）。

请求方法

POST： https://数据中心域名/push.json

调用频率：共享 /push.json 的限频配额，即每小时限发送 2 次，每天（自然日）最多发送 3 次。即时通讯服务端按标签推送的处理能力为每秒钟 1500 条，即每秒钟可以向 1500 个目标用户下发消息。如需超过此限制，请联系商务。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/json，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
platform	String[]	是	目标操作系统，iOS、Android 最少传递一个。如果需要给两个系统推送消息时，则需要全部填写，发送时如目标用户在 Web 端登录也会收到此条消息。
fromuserid	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
audience	Object	是	推送条件。支持按用户 ID 推送，按用户标签推送（tag、tag_or）、按应用包名推送（packageName）和按指定平台全部推送（is_to_all）。注意：如果推送条件中 is_to_all 为 true，则忽略其他推送条件。
audience.tag	String[]	否	用户标签，每次发送时最多发送 20 个标签，标签之间为 AND 的关系，is_to_all 为 true 时参数无效。
audience.tag_or	String[]	否	用户标签，每次发送时最多发送 20 个标签，标签之间为 OR 的关系，is_to_all 为 true 时参数无效，tag_or 同 tag 参数可以同时存在。
audience.userid	String[]	否	用户 ID，每次发送时最多发送 1000 个用户，如果 tag 和 userid 两个条件同时存在时，则以 userid 为准，如果 userid 有值时，则 platform 参数无效，is_to_all 为 true 时参数无效。
audience.is_to_all	Boolean	是	是否全部推送，false 表示按 tag、tag_or 或 userid 条件推送，true 表示向所有用户推送，tag、tag_or 和 userid 条件无效。
message.content	String	是	所发送消息的内容，单条消息最大 128k。 内置消息以 JSON 方式进行数据序列化，消息中可选择是否携带用户信息。您可以在内置消息类型（见消息类型概述）中找到消息结构示例。 如果 objectName 为自定义消息类型，该参数可自定义格式，不限于 JSON。
message.objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 “RC:” 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
notification	Object	是	按操作系统类型推送通知内容，如 platform 中设置了给 iOS 和 Android 系统推送消息，而在 notification 中只设置了 iOS 的推送内容，则 Android 的推送内容为最初 alert 设置的内容。
notification.title	String	否	通知栏显示标题，最长不超过 50 个字符。
notification.forceShowPushContent	Int	否	是否越过客户端配置，强制在推送通知内显示通知内容（pushContent）。默认值 0 表示不强制，1 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置forceShowPushContent 为 1 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
notification.alert	String	否	推送通知内容。注意，如果此处 notification.alert 不传，则必须在 notification.ios.alert 和 notification.android.alert 分别指定 iOS 和 Android 下的推送通知内容，否则无法正常推送。一旦指定了各平台推送内容，则推送内容以对应平台系统的 alert 为准。如果都不填写，则无法发起推送。
notification.ios	Object	否	设置 iOS 平台下的推送及附加信息。详细查看 ios 参数结构说明。
notification.android	Object	否	设置 Android 平台下的推送及附加信息。详细查看 android 参数结构说明。

- notification.ios 参数结构说明

参数	类型	必传	说明
title	String	否	通知栏显示的推送标题，仅针对 iOS 平台，支持 iOS 8.2 及以上版本。该属性优先级高于 `notification.title`。
contentAvailable	Int	否	针对 iOS 平台，静默推送是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
alert	String	否	推送通知内容，传入后默认的推送通知内容失效。
badge	int	否	应用角标，仅针对 iOS 平台；不填时，表示不改变角标数；为 0 或负数时，表示 App 角标上的数字清零；否则传相应数字表示把角标数改为指定的数字，最大不超过 9999，参数在 `ios` 节点下设置，详细可参考“设置 iOS 角标数 HTTP 请求示例”。
thread-id	String	否	iOS 平台通知栏分组 ID，相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示
apns-collapse-id	String	否	iOS 平台，从 iOS10 开始支持，设置后设备收到有相同 ID 的消息，会合并成一条
category	String	否	iOS 富文本推送的类型开发者自己定义，自己在 App 端进行解析判断，与 richMediaUri 一起使用，当设置 category 后，推送时默认携带 mutable-content 进行推送，属性值为 1。
richMediaUri	String	否	iOS 富文本推送内容的 URL，与 category 一起使用。
interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的 “定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

- **`notification.android` 结构说明**

参数	类型	必传	说明
alert	String	否	Android 平台下推送通知内容，传入后默认的推送通知内容失效。
honor.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。

参数	类型	必传	说明
honor.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.channelId	String	否	华为推送通知渠道的 ID。详见自定义通知渠道。
hw.importance	String	否	华为推送通知栏消息优先级，取值 NORMAL、LOW，默认为 NORMAL 重要消息。
hw.image	String	否	华为推送自定义的通知栏消息右侧大图标 URL，如果不设置，则不展示通知栏右侧图标。URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp，超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
mi.channelId	String	否	小米推送通知渠道的 ID。详见小米推送消息分类新规。
mi.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。图片要求：大小120 * 120px，格式为 png 或者 jpg 格式。
oppo.channelId	String	否	oppo 推送通知渠道的 ID。详见推送私信通道申请。
vivo.classification	String	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
vivo.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category` ，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。

参数	类型	必传	说明
fcm.channelId	String	否	Google FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见 Android 官方文档。
fcm.collapse_key	String	否	Google FCM 推送中可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
fcm.imageUrl	String	否	Google FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

请求示例

```
POST /push.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{
"platform":["ios","android"],
"fromuserid": "fromuseId1",
"audience":{
"tag":["女","年轻"],
"tag_or":["北京","上海"],
"userid":["123","456"],
"is_to_all":false
},
"message": {
"content": "{\"content\":\"1111\",\"extra\":\"aa\"}",
"objectName": "RC:TxtMsg"
},
"notification":{
"title":"标题",
"forceShowPushContent":0,
"alert":"this is a push",
"ios":
{
"alert": "override alert",
"thread-id":"223",
"apns-collapse-id":"111",
"extras": {"id": "1","name": "2"}
},
"android": {
"alert": "override alert",
"hw":{
"channelId":"NotificationKanong",
"importance": "NORMAL",
"image":"https://example.com/image.png"
},
"mi":{
"channelId":"rongcloud_kanong",
"large_icon_uri":"https://example.com/image.png"
},
"oppo":{
"channelId":"rc_notification_id"
},
"vivo":{
"classification":"0"
},
"extras": {"id": "1","name": "2"}
}
}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送应用包名通知

更新时间:2024-08-30

App 用户可以向 App 下指定应用包名的全部用户发送系统会话消息，该功能称为「应用包名通知」。

例如，客户通过控制台创建了一个应用，该应用下有两个客户端：学生端、教师端（通过创建两个应用包名实现），当需要给所有使用学生端或教师端的用户发送运营消息时，可使用此功能接口实现。

该功能使用 `/push.json` 接口。

- 支持发送即时通讯服务预定义的消息类型（见消息类型概述）。消息的类型（`objectName` 字段）决定了消息是否支持离线推送通知，以及客户端在收到该消息后，是否展示在聊天界面、会话列表，是否存入本地数据库。
- 支持发送客户自定义类型的消息。客户端在收到自定义消息后，是否展示在聊天界面、会话列表，是否存入本地数据库取决于客户端注册的自定义消息类型定义。
- 只能通过即时通讯服务端 API 进行发送，会话类型为 SYSTEM。该类型的会话不支持终端用户在收到消息后进行回复。

假设发送一条文本消息（`objectName` 为 `RC:TxtMsg`，属于客户端 SDK 会存储的内置消息类型，且可触发离线推送），效果如下：

- 客户端如果在线，会即时收到一条消息，所在会话的 Target ID 为调用接口时传入的 `fromUserId`，会话类型为系统会话（类型为 SYSTEM）。
- 客户端如果离线，消息会触发服务端远程推送，客户端如已集成并启用推送服务，则会收到推送通知。

> 提示
>
> Web 端用户不支持此功能，终端用户收到系统消息后，不支持消息回复功能。

开通服务

使用应用包名通知功能前，请确认已为当前 **App Key** 开通相关服务。详见全量用户通知服务配置。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/push.json

频率限制：共享 `/push.json` 的限频配额，即每小时限发送 2 次，每天（自然日）最多发送 3 次。即时通讯服务端按应用包名推送的处理能力为每秒钟 2000 条，即每秒钟可以向 2000 个目标用户下发消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
platform	String[]	是	目标操作系统，iOS、Android 最少传递一个。如果需要给两个系统推送消息时，则需要全部填写，发送时如目标用户在 Web 端登录也会收到此条消息。
fromuserid	String	是	发送人用户 ID。 注意：发送消息所使用的用户 ID 必须已获取过用户 Token，否则消息一旦触发离线推送，通知内无法正确显示发送者的用户信息。
audience	String	是	推送条件。使用「应用包名用户通知」时，请设置为 `packageName`。
audience.tag	String[]	否	用户标签，每次发送时最多发送 20 个标签，标签之间为 AND 的关系，is_to_all 为 true 时参数无效。
audience.tag_or	String[]	否	用户标签，每次发送时最多发送 20 个标签，标签之间为 OR 的关系，is_to_all 为 true 时参数无效，tag_or 同 tag 参数可以同时存在。
audience.userid	String[]	否	用户 ID，每次发送时最多发送 1000 个用户，如果 tag 和 userid 两个条件同时存在时，则以 userid 为准，如果 userid 有值时，则 platform 参数无效，is_to_all 为 true 时参数无效。
audience.packageName	String	否	应用包名，is_to_all 为 true 时，此参数无效。与 tag、tag_or 同时存在时为 And 的关系，向同时满足条件的用户推送。与 userid 条件同时存在时，以 userid 为准进行推送。
audience.is_to_all	Boolean	是	是否全部推送，false 表示按 tag 、tag_or 或 userid 条件推送，true 表示向所有用户推送，tag、tag_or 和 userid 条件无效。
message.content	String	是	所发送消息的内容，单条消息最大 128k。 • 内置消息类型：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见消息类型概述中各内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 `{"content":"Hello world!"}` 序列化后的结果作为此处 `content` 字段的值。 • 自定义消息类型（`objectName` 字段必须指定为自定义消息类型）：如果发送自定义消息，该参数可自定义格式，不限于 JSON。
message.objectName	String	是	消息类型，接受内置消息类型（见消息类型概述）或自定义消息的消息类型值。 注意：在自定义消息时，消息类型不可以 “RC:” 开头，以免与系统内置消息类型重名；消息类型长度不可超过 32 个字符。SDK 中必须已注册过该自定义消息，否则 SDK 收到该消息后将无法解析。
notification	Object	是	按操作系统类型推送通知内容，如 platform 中设置了给 iOS 和 Android 系统推送消息，而在 notification 中只设置了 iOS 的推送内容，则 Android 的推送内容为最初 alert 设置的内容。
notification.title	String	否	通知栏显示标题，最长不超过 50 个字符。

参数	类型	必传	说明
notification.forceShowPushContent	Int	否	是否越过客户端配置，强制在推送通知内显示通知内容（pushContent）。默认值 0 表示不强制，1 表示强制。 说明：客户端设备可设置在接收推送通知时仅显示类似「您收到了一条通知」的提醒。从服务端发送消息时，可通过设置forceShowPushContent 为 1 越过该配置，强制客户端针在此条消息的推送通知中显示推送内容。
notification.alert	String	否	推送通知内容。注意，如果此处 notification.alert 不传，则必须在 notification.ios.alert 和 notification.android.alert 分别指定 iOS 和 Android 下的推送通知内容，否则无法正常推送。一旦指定了各平台推送内容，则推送内容以对应平台系统的 alert 为准。如果都不填写，则无法发起推送。
notification.ios	Object	否	设置 iOS 平台下的推送及附加信息。详细查看 ios 参数结构说明。
notification.android	Object	否	设置 Android 平台下的推送及附加信息。详细查看 android 参数结构说明。

- notification.ios 参数结构说明

参数	类型	必传	说明
title	String	否	通知栏显示的推送标题，仅针对 iOS 平台，支持 iOS 8.2 及以上版本。该属性优先级高于 notification.title。
contentAvailable	Int	否	针对 iOS 平台，静默推送是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
alert	String	否	推送通知内容，传入后默认的推送通知内容失效。
badge	int	否	应用角标，仅针对 iOS 平台；不填时，表示不改变角标数；为 0 或负数时，表示 App 角标上的数字清零；否则传相应数字表示把角标数改为指定的数字，最大不超过 9999，参数在 ios 节点下设置，详细可参考“设置 iOS 角标数 HTTP 请求示例”。
thread-id	String	否	iOS 平台通知栏分组 ID，相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示
apns-collapse-id	String	否	iOS 平台，从 iOS10 开始支持，设置后设备收到有相同 ID 的消息，会合并成一条
category	String	否	iOS 富文本推送的类型开发者自己定义，自己在 App 端进行解析判断，与 richMediaUri 一起使用，当设置 category 后，推送时默认携带 mutable-content 进行推送，属性值为 1。
richMediaUri	String	否	iOS 富文本推送内容的 URL，与 category 一起使用。

参数	类型	必传	说明
interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式”都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

- `notification.android` 结构说明

参数	类型	必传	说明
alert	String	否	Android 平台下推送通知内容，传入后默认的推送通知内容失效。
honor.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
honor.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.channelId	String	否	华为推送通知渠道的 ID。详见自定义通知渠道。
hw.importance	String	否	华为推送通知栏消息优先级，取值 NORMAL、LOW，默认为 NORMAL 重要消息。
hw.image	String	否	华为推送自定义的通知栏消息右侧大图标 URL，如果不设置，则不展示通知栏右侧图标。URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp，超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
mi.channelId	String	否	小米推送通知渠道的 ID。详见小米推送消息分类新规。

参数	类型	必传	说明
mi.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。图片要求：大小120 * 120px，格式为 png 或者 jpg 格式。
oppo.channelId	String	否	oppo 推送通知渠道的 ID。详见推送私信通道申请。
vivo.classification	String	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
vivo.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category` ，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
fcm.channelId	String	否	Google FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见 Android 官方文档。
fcm.collapse_key	String	否	Google FCM 推送中可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
fcm.imageUrl	String	否	Google FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

请求示例

```
POST /push.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{
"platform":["ios","android"],
"fromuserid": "fromuseId1",
"audience":{
"packageName":"xxx.rong.xxx",
"is_to_all":false
},
"message": {
"content": "{\"content\":\"1111\",\"extra\":\"aa\"}",
"objectName": "RC:TxtMsg"
},
"notification":{
"title":"标题",
"forceShowPushContent":0,
"alert":"this is a push",
"ios":
{
"alert": "override alert",
"thread-id":"223",
"apns-collapse-id":"111",
"extras": {"id": "1","name": "2"}
},
"android": {
"alert": "override alert",
"hw":{
"channelId":"NotificationKanong",
"importance": "NORMAL",
"image":"https://example.com/image.png"
},
"mi":{
"channelId":"rongcloud_kanong",
"large_icon_uri":"https://example.com/image.png"
},
"oppo":{
"channelId":"rc_notification_id"
},
"vivo":{
"classification":"0"
},
"extras": {"id": "1","name": "2"}
}
}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

撤回全量用户落地通知

更新时间:2024-08-30

撤回指定消息本质是发送一条撤回命令消息。移动端收到撤回命令消息后，原目标消息将被删除，同时生成一条 objectName 是 RC:RcNtf 的通知消息，会话界面中可根据 RC:RcNtf 展示。

支持撤回以下功能发送的消息：

- 发送全量用户落地通知：`/message/broadcast.json`
- 发送标签用户通知：`/push.json`
- 发送应用包名用户通知：`/push.json`

> 提示
>
> 撤回命令消息也会存储到历史消息中，SDK 在获取历史消息时，会获取到撤回命令消息和被撤回的原始消息。移动端 SDK 已内部实现删除逻辑，开发者无需额外处理。

请求方法

POST： https://数据中心域名/message/broadcast/recall.json

频率限制：每小时限发送 2 次，每天（自然日）最多发送 3 次。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
fromUserId	String	是	消息发送人用户 ID。
messageUID	String	是	消息唯一标识。目前只能通过历史消息日志获取，对应字段名称为 msgUID。 • 获取历史消息日志数据有一定延迟，详见历史消息日志。 • 通过全量用户落地通知发送的消息不支持全量消息路由服务，因此无法通过全量消息路由服务获取该字段的值。

参数	类型	必传	说明
sentTime	Long	是	消息发送时间。目前只能通过历史消息日志获取，对应字段名称为 dateTime。 • 获取历史消息日志数据有一定延迟，详见历史消息日志。 • 通过全量用户落地通知发送的消息不支持全量消息路由服务，因此无法通过全量消息路由服务获取该字段的值。
isAdmin	Int	否	是否为管理员，默认为 0，设为 1 时，IMKit 收到此条消息后，小灰条默认显示为“管理员 撤回了一条消息”。
isDelete	Int	否	默认为 0 撤回该条消息同时，用户端将该条消息删除并替换为一条小灰条撤回提示消息；为 1 时，该条消息删除后，不替换为小灰条提示消息。
extra	String	否	扩展信息，可以放置任意的数据内容。

请求示例

```
POST /message/recall.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

fromUserId=fDR2cVpxxR5zSMUNh3yAwh&messageUID=5FGT-7VA9-G4DD-4V5P&sentTime=1507778882124
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

发送指定用户不落地通知

更新时间:2024-08-30

App 业务端可以向指定用户发送「推送通知」，本质上是直接通过第三方或即时通讯服务自建的推送通道向 App 用户推送一条通知。该功能只触发推送，不发送消息，不产生会话，称为「单个用户不落地通知」。

- 只能通过即时通讯服务端 API 进行发送。单次最多同时向 100 人发送推送。
- 「推送通知」的所有内容仅展现在通知栏。无论客户端 App 是否在前台，始终以通知形式展示在系统通知栏中。该功能不发送消息，不产生会话，因此用户无法在任何聊天会话中看到不落地通知的内容。
- 客户端不会存储不落地通知。
- 始终使用推送通道，因此不受客户端与即时通讯服务端之间的连接状态的影响。即使客户端不在前台、不在线（例如 App 被杀死），只要可正常接收推送，就可以收到不落地通知。

> 提示
>
> - 即时通讯服务将 30 天内连接过 IM 服务的设备作为推送的目标。30 天内未打开过应用的设备，被视为应用已被卸载。
> - 如果客户端设备不允许推送（断开连接时设置不允许推送），或遇到其他原因导致推送失败，则无法接收到不落地通知。

请求方法

POST： https://数据中心域名/push/user.json

频率限制： 每秒钟限推送 100 条消息，每次最多同时向 100 人发送，如：一次发送 100 人时，视为 100 条消息。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	必传	说明
userIds	String[]	是	用户 ID，每次发送时最多发送 100 个用户。
notification	Object	是	按操作系统类型推送消息内容。
notification.title	String	否	通知栏显示标题，最长不超过 50 个字符。
notification.pushContent	String	是	推送消息内容。
notification.ios	Object	否	设置 iOS 平台下的推送及附加信息，详见 `ios` 结构说明。
notification.android	Object	否	设置 Android 平台下的推送及附加信息，详见 `android` 结构说明。

- `notification.ios` 结构说明：

参数	类型	必传	说明
contentAvailable	Int	否	针对 iOS 平台，静默推送是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0
badge	int	否	应用角标，仅针对 iOS 平台；不填时，表示不改变角标数；为 0 或负数时，表示 App 角标上的数字清零；否则传相应数字表示把角标数改为指定的数字，最大不超过 9999，参数在 `ios` 节点下设置，详细可参考“设置 iOS 角标数 HTTP 请求示例”。
thread-id	String	否	iOS 平台通知栏分组 ID，相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示
apns-collapse-id	String	否	iOS 平台，从 iOS10 开始支持，设置后设备收到有相同 ID 的消息，会合并成一条
category	String	否	iOS 富文本推送的类型开发者自己定义，自己在 App 端进行解析判断，与 richMediaUri 一起使用，当设置 category 后，推送时默认携带 mutable-content 进行推送，属性值为 1。
richMediaUri	String	否	iOS 富文本推送内容的 URL，与 category 一起使用。
interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的 “定时推送摘要”、“专注模式” 都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

- `notification.android` 结构说明

参数	类型	必传	说明
honor.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。

参数	类型	必传	说明
honor.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.channelId	String	否	华为推送通知渠道的 ID。详见自定义通知渠道。
hw.importance	String	否	华为推送通知栏消息优先级，取值 NORMAL、LOW，默认为 NORMAL 重要消息。
hw.image	String	否	华为推送自定义的通知栏消息右侧大图标 URL，如果不设置，则不展示通知栏右侧图标。URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp，超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
mi.channelId	String	否	小米推送通知渠道的 ID。详见小米推送消息分类新规。
mi.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。图片要求：大小120 * 120px，格式为 png 或者 jpg 格式。
oppo.channelId	String	否	oppo 推送通知渠道的 ID。详见推送私信通道申请。
vivo.classification	String	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
vivo.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category` ，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。

参数	类型	必传	说明
fcm.channelId	String	否	Google FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见 Android 官方文档。
fcm.collapse_key	String	否	Google FCM 推送中可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。
fcm.imageUrl	String	否	Google FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

请求示例

```
POST /push/user.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{
"userIds":["123","456"],
"notification":{
"title":"标题",
"pushContent":"this is a push",
"ios":
{
"thread-id":"223",
"apns-collapse-id":"111",
"extras": {"id": "1","name": "2"}
},
"android": {
"hw":{
"channelId":"NotificationKanong",
"importance": "NORMAL",
"image":"https://example.com/image.png"
},
"mi":{
"channelId":"rongcloud_kanong",
"large_icon_uri":"https://example.com/image.png"
},
"oppo":{
"channelId":"rc_notification_id"
},
"vivo":{
"classification":"0"
},
"extras": {"id": "1","name": "2"}
}
}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

群组业务概述

更新时间:2024-08-30

群聊是即时通讯类应用中常见的多人通讯方式，一般包含两个及以上的用户。群组业务支持丰富的群组成员管理、禁言管理等特性，支持离线消息推送和历史消息记录漫游，可用于兴趣群、办公群、客服服务沟通等。

群组业务要点如下：

- 即时通讯服务只负责将消息传达给群组中的所有用户，不维护群组成员的资料（头像、名称、群成员名片等），需要由开发者应用服务器维护。
- 创建/解散/加入/退出群组等群组管理操作，必须由 App 服务器请求即时通讯服务端 API 实现。即时通讯客户端 SDK 不提供相应方法。详见下方群组管理功能。
- App Key 下可创建的群组数量没有限制，单个群组默认成员上限为 3000 人。如果您是专有云用户，您可以联系融云商务人员来修改群成员上限。
- 单个用户可加入的群组数量无限制。
- 从控制台 IM 服务管理页面为 App Key 开启**单群聊消息云端存储**服务后，可使用即时通讯服务提供的消息存储服务，实现消息历史记录漫游。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明页面及计费说明文档为准。

服务配置

在控制台创建应用后默认支持群组业务，不需要申请开通。

群组业务的部分基础功能与增值服务可以在控制台的免费基础功能和 IM 服务管理页面进行开通和配置。

群组管理功能

即时通讯服务不会托管用户，也不管理群组的业务逻辑，因此群的业务逻辑全部需要在 App 服务器进行实现。

> 提示
>
> **群主、群管理员、群公告、邀请入群、群号搜索等均为群组业务逻辑，需在 App 侧自行实现。**

对于客户端开发人员来说，创建群组、解散等基础管理操作只需要与 App 自身的业务服务端交互即可，由 App 后端负责调用相应的即时通讯服务端 API（Server API）接口完成相关操作。

下表列出了即时通讯服务端提供的群组基础管理接口。注意，群组管理需要由 App 后端调用相应的即时通讯服务端 API（Server API）接口完成。客户端不提供群组管理的 API。

功能分类	功能描述	即时通讯服务端 API
创建、解散群组	提供创建者用户 ID、群组 ID、和群名称，向即时通讯服务端申请建群。如解散群组，则群成员关系不复存在。	创建群组、解散群组
加入、退出群组	加入群组后，默认可查看入群以后产生的新消息。退出群组后，不再接收该群的新消息。	加入群组、退出群组
修改即时通讯服务端的群组信息	修改在即时通讯的推送服务中使用的群组信息。	刷新群组信息
查询群组成员	查询指定群组所有成员的用户 ID 信息。	查询群组成员
查询用户所在群组	根据用户 ID 查询该用户加入的所有群组，返回群组 ID 及群组名称。即时通讯服务不存储群组资料信息，群组资料及群成员信息需要开发者在应用服务器自行维护，如应用服务端维护的用户群组关系有缺失时，可通过此接口来核对校验。	查询用户所在群组
同步用户所在群组	向即时通讯服务端同步指定用户当前所加入的所有群组，防止应用中的用户群组信息与即时通讯服务端的用户所属群信息不一致。如果在集成即时通讯服务前 App Server 上已有群组及成员数据，第一次连接即时通讯服务时，可使用此接口向即时通讯服务端同步已有的用户与群组对应关系。	同步用户所在群组
群组单人禁言	在指定的单个群组中或全部群组中，禁言一个或多个用户。被禁言用户可以接收查看群组中其他用户消息，但不能通过客户端 SDK 发送消息。	单人禁言
群组全体禁言	将群组全体成员禁言。被禁言群组的所有成员均不能发送消息，需要某些用户可以发言时，可将此用户加入到群禁言用户白名单中。	全体禁言
群组禁言用户白名单	群组被整体禁言后，禁言白名单中用户可以发送群消息。	全体成员禁言白名单

群聊消息功能

群聊消息功能与单聊业务类似，共用部分 API 及配置。

功能	描述	客户端 API	即时通讯服务端 API
发送消息	可发送普通消息与媒体消息，例如文本、图片、GIF 等。支持在发送消息时添加 @ 信息。	支持	发送群聊消息
发送群聊定向消息	可发送普通消息与媒体消息给群组中的指定的一个或多个成员，其他成员不会收到该消息。	支持	发送群聊定向消息
接收消息	监听并实时接收消息，或在客户端上线时接收离线消息。	支持	不提供该 API
群聊消息已读回执	发送群消息后如需要查看消息的阅读状态，需要先发送回执请求，在通过接受者的响应获取已读数据。	支持	不提供该 API
离线消息	支持离线消息存储，存储时间可设置（1 ～ 7 天），默认存储 7 天内的所有群消息，支持调整存储时长与存储的群消息数量。	支持	不提供该 API
离线消息推送	离线状态下，群组中有新消息时，支持 Push 通知。	支持	不提供该 API
撤回消息	消息发送成功后可撤回该条消息。	支持	撤回消息
本地搜索消息	消息存储在本地（移动端），支持按关键字或用户搜索本地指定会话的消息内容。	支持	不提供该 API
获取历史消息	从本地数据库或远端获取历史消息。注意，从远端获取历史消息需要开通单群聊消息云存储服务，默认存储时长为 6 个月。	支持	不提供该 API

功能	描述	客户端 API	即时通讯服务端 API
获取历史消息日志	即时通讯服务端可以保存 APP 内所有会话的历史消息记录，历史消息记录以日志文件方式提供，并已经过压缩。您可以使用服务端 API 获取、删除指定 App 的历史消息日志	不提供该 API	获取历史消息日志
本地插入消息	在本地数据库中插入消息。本地插入的消息不会实际发送给服务器和对方。	支持	不提供该 API
删除消息	支持按会话删除本地和存储在服务器的指定消息或会话中全部历史消息。	支持	消息清除
单群聊消息扩展	为原始消息增加状态标识（扩展数据为 KV 键值对），提供添加、删除、查询扩展信息的接口。	支持	单/群聊消息扩展
自定义消息类型	如果内置消息类型满足不了您的需求，可以自定义消息类型。支持自定义普通消息类型与自定义媒体消息类型。服务端直接发送自定义消息即可，注意消息内容结构需要与客户端定义的消息类型保持一致。	支持	发送群聊消息

默认新入群的成员的用户仅可接收加入群组后产生的消息。如果需要查看入群之前的历史消息，请为 App Key 开启以下两项服务（请注意区分开发/生产环境）：

从控制台 IM 服务管理 页面开启**单群聊消息云端存储**服务。开启该服务后，可使用即时通讯服务提供的消息存储服务，实现消息历史记录漫游。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。

- 从控制台免费基础功能 页面开启**新用户获取加入群组前历史消息**。

群聊会话功能

群聊会话功能与单聊业务类似，共用部分 API 及配置。

功能	描述	客户端 API	即时通讯服务端 API
获取会话	SDK 会根据收发的消息在本地数据库中生成对应会话。您可以从本地数据库获取 SDK 生成的会话列表。	支持	不提供该 API
获取会话未读消息	从指定会话中获取未读消息，可满足 App 跳转到第一条未读消息、展示全部未读 @ 消息的需求。	支持	不提供该 API
处理会话未读消息数	获取或清除会话中的未读消息数，可用于 UI 展示。	支持	不提供该 API
删除会话	从 SDK 生成的会话列表中删除一个会话或多个会话	支持	不提供该 API
会话草稿	保存一条草稿内容至指定会话。	支持	不提供该 API
输入状态	可设置指定的群聊会话，收到新的消息后是否进行提醒，默认进行新消息提醒。	支持	不提供该 API
管理会话标签	创建和管理标签信息数据，用于对会话进行标记分组。每个用户最多可以创建 20 个标签。App 用户创建的标签信息数据会同步即时通讯服务端。	支持	不提供该 API
设置与使用会话标签	使用会话标签对会话进行分组。	支持	不提供该 API
会话置顶	在会话列表中将指定会话置顶。	支持	会话置顶
会话免打扰	控制用户在客户端设备离线时，是否可针对离线消息接收推送通知。支持按照会话或按会话类型设置免打扰。	支持	免打扰功能概述

功能	描述	客户端 API	即时通讯服务端 API
多端同步会话免打扰/置顶状态	SDK 提供了会话状态（置顶或免打扰）同步机制，通过设置会话状态同步监听器，当在其它端修改会话状态时，可在本端实时监听到会话状态的改变。	支持	不提供该 API
多端同步阅读状态	在同一用户账户的多个设备间主动同步会话的阅读状态。	支持	不提供该 API

群组与聊天室的区别

即时通讯服务提供群组与聊天室业务，其主要区别如下，客户可根据自己的业务场景进行选择：

功能	群组（group）	聊天室（Chatroom）
场景	类似微信的群组，无论是否在线都会接收消息	只有在线用户可接收消息，可用于直播、社区、游戏、广场交友、兴趣讨论等场景。
离线缓存消息	支持离线消息存储，存储时间可设置（1～7 天），默认存储 7 天。	无离线消息，只有在线用户才可收到聊天室消息
人数限制	默认一个群上限为 3000 人	聊天室人数无上限
消息提醒	离线状态，群组中有新消息时，支持远程推送（PUSH）通知	离开聊天室后不再接收消息
本地存储	移动端本地数据库存储，提供本地消息搜索接口	退出聊天室后同时删除本地聊天室消息，不支持消息搜索功能
云端存储	需开通单群聊消息云存储，可以提供 6 - 36 个月存储服务	需开通聊天室消息云存储，可以提供 2 - 36 个月存储服务
用户加入限制	一个用户可加入多个群组，无限制	默认一个用户只能加入一个聊天室，加入多个聊天室功能可在控制台自行开通
加入后消息获取逻辑	默认加入群组后，只能查看加入后群组中产生的消息。如需要查看群历史消息，则需要开通单群聊消息云存储后，再开通“查看加入前群消息”功能	加入后可获取聊天室中最新的 50 条消息。
销毁/解散逻辑	需要通过 AppServer 自行调用解散群组接口。	提供销毁聊天室接口，可通过 AppServer 调用。同时聊天室中 1 小时内没有消息产生时，将自动销毁聊天室。
消息可靠度	100% 可靠，不丢消息。	消息量较大时，超出服务端消费上限的消息将被主动抛弃。您可通过用户白名单、消息白名单、自定义消息级别等服务，改变消息抛弃策略。如果用户在聊天室的用户白名单内，该用户所发送的消息在消息量大时也不会被抛弃。 如需了解服务端消费上限与如何改变消息抛弃策略，可参见服务端文档消息优先级服务、聊天室白名单服务。
相关接口调用	SDK 不提供群组管理功能接口，通过 Server API 提供群组功能接口。	SDK 和 Server API 同时提供功能接口，销毁聊天室操作只能通过 Server API 方式调用。
发送消息频率	每个客户端 5 条/秒；服务端调用，20 条/秒	每个客户端 5 条/秒；服务端调用，100 条/秒

创建群组

更新时间:2024-08-30

即时通讯服务不管理群组的业务逻辑，因此群组的业务逻辑全部在 App 服务器进行实现，对于客户端开发人员来说，创建群组只需要与 App 后端交互即可。

> 提示
>
> 每个群群成员上限 3000 人，App 内的群组数量没有限制，每个用户加入和创建的群组数量没有限制。

基本流程：

1. App 客户端需要创建群组时，向 App 后端发起请求。
2. App 后端自行生成群组 ID，并调用即时通讯服务端 API 接口创建群组。同时可以传入需要入群的用户 ID。
3. 即时通讯服务创建群组成功后，返回给 App 后端。
4. App 后端通知客户端群组创建成功。用户已加入该群组，可以在群组中收发消息。

创建群组时序图

请求方法

POST： https://数据中心域名/group/create.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
userId	String	必传	要加入群的用户 ID，最多不超过 1000 个。
groupId	String	必传	创建群组 ID，最大长度 64 个字符。支持大小写英文字母与数字的组合。
groupName	String	必传	群组 ID 对应的名称，用于在发送群组消息显示远程 Push 通知时使用，如群组名称改变需要调用刷新群组信息接口同步。

请求示例

```
POST /group/create.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=1&userId=2&groupId=123&groupName=TestGroup
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

常见问题

即时通讯服务不维护群的基本信息（头像、名称、群成员名片等），需要由开发者应用服务器维护。

创建群组必须由 App 服务器请求即时通讯服务端 API 实现。即时通讯服务移动端 SDK 不提供创建群组的方法。

即时通讯服务在创建群组、用户加入、退出群组等完成群组操作后，不会发送通知消息。创建群组、成功加入及退出群组等群通知消息，需要由客户根据自身的业务场景决定，是否发送、什么时候发送。

以创建群组为例，发送群组通知消息的流程如下：

1. App 请求自己的 App Server 创建群组。
2. App Server 调用即时通讯服务端 API，申请创建群组。
3. 返回 200 成功后，使用发送群消息 API 接口，发送创建群组通知消息。

解散群组

更新时间:2024-08-30

将群组解散，所有用户都无法再接收该群的消息。

对本地历史消息记录的影响：

- **群组解散后，客户端仍可查看已保存到本地的群组历史消息。**

对远端历史消息记录的影响（仅适用于已开通单群聊消息云存储[?]服务的情况）：

- **解散群组不会删除该群组在服务端的历史消息。如需清除，可使用即时通讯服务端删除消息 API 接口。**
- **群组关系解散后，App 客户端无权再拉取云存储服务中的群组历史消息。**

> 提示
>
> **如果单群聊消息云存储服务中的历史消息未被清除，只要创建同 ID 的群，客户端仍可拉取到之前的历史消息。**

解散群组的基本流程：

1. **用户在 App 进行解散群组操作，并向 App 服务器发起解散群组请求。**
2. **App 后端调用即时通讯服务端 API 解散群组。**

解散群组时序图：

请求方法

POST： https://数据中心域名/group/dismiss.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	操作解散群的用户 ID，可以为任何用户 ID ，非群组创建者也可以解散群组。
groupId	String	是	要解散的群的群组 ID。

请求示例

```
POST /group/dismiss.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=1&groupId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

常见问题

1. 调用解散群组接口不会主动向 SDK 发送通知消息，如果希望给用户发送解散群组通知消息，可发送系统消息方式进行通知。

加入群组

更新时间:2024-08-30

即时通讯服务不管理群组的业务逻辑，因此群组的业务逻辑全部在 App 服务器进行实现，对于客户端开发人员来说，创建群组只需要与 App 后端交互即可。

> 提示
>
> 每个群群成员上限 3000 人，App 内的群组数量没有限制，每个用户加入和创建的群组数量没有限制。

基本流程：

1. 客户端请求 App 后端加入群组。
2. App 后端调用即时通讯服务端 API，申请加入群组。群组 ID 由 App 服务器管理，创建成功后，同时返回给客户端。
3. 用户成功加入该群组后，可以在该群组内收发消息。

> 用户加入群组后，默认只能查看加入时间点后的群消息。如果希望用户入群后可查看群历史记录，需在控制台开通以下功能：
>
> - 单群聊历史消息云存储
> - 新用户获取加入群组前历史消息

加入群组时序图

请求方法

POST： https://数据中心域名/group/join.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	要加入群的用户 ID，可提交多个。最多不超过 1000 个。
groupId	String	是	要加入的群的群组 ID。
groupName	String	否	要加入的群的名称。 注意：加入群组时，如果传入 groupName，则会修改推送通知中携带的群组名称，效果与调用刷新群组信息一致。

请求示例

```
POST /group/join.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=1&userId=2&groupId=123&groupName=TestGroup
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

退出群组

更新时间:2024-08-30

将用户从群中移除，不再接收该群组的消息。

基本流程：

1. 用户在 App 进行退出群组操作，该退出操作请求 App 服务器。
2. 由 App 服务器调用即时通讯服务端 API，将该用户移除群组，移除后不再接收该群组的消息。

退出群组时序图

请求方法

POST： https://数据中心域名/group/quit.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	要退出群的用户 ID，可提交多个，最多不超过 1000 个。
groupId	String	是	要退出的群的群组 ID。

请求示例

```
POST /group/quit.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=1&userId=2&groupId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

常见问题

1. 即时通讯服务不维护群的基本信息（头像、名称、群成员名片等），需要由开发者应用服务器维护。

2. 调用退出群组接口不会主动向 SDK 发送通知消息，如果希望给用户发送退出群组通知消息，可发送系统消息方式进行通知。

查询群组成员

更新时间:2024-08-30

查询指定群组所有成员的用户 ID 信息。

请求方法

POST： https://数据中心域名/group/user/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	群组 ID。

请求示例

```
POST /group/user/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
users	String	群成员数组。
id	String	群成员的用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"users":[
{
"id":"10001"
},
{
"id":"10002"
},
{
"id":"10000"
},
{
"id":"10003"
}
]
}
```

同步用户所在群组

更新时间:2024-08-30

向即时通讯服务端同步指定用户当前所加入的所有群组，防止应用中的用户群组信息与即时通讯服务端的用户所属群信息不一致。

如果在集成即时通讯服务前 App Server 上已有群组数据，第一次连接即时通讯服务时，可使用此接口向即时通讯服务端同步已有的用户与群组对应关系。

请求方法

POST： https://数据中心域名/group/sync.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	被同步群信息的用户 ID。
group[id]=name	String	否	该用户所属的群信息，如群组 ID 已经存在，则同时刷新对应群组名称。此参数可传多个，参见下面示例。

> 提示
>
> 当不提交 group[id]=name 参数时，表示解除 userId 对应群的绑定关系。

请求示例

```
POST /group/sync.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2014&group[10001]=TestGroup1&group[10002]=TestGroup2&group[10003]=TestGroup3
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询用户所在群组

更新时间:2024-08-30

根据用户 ID 查询该用户加入的所有群组，返回群组 ID 及群组名称。默认获取用户 ID 加入的前 5000 个群组列表。

> 提示
>
> 即时通讯服务不存储群组资料信息，群组资料及群成员信息建议开发者在应用服务器自行维护，如应用服务端维护的用户群组关系有缺失时，可通过此接口来核对校验。

请求方法

POST： https://数据中心域名/user/group/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
page	Int	否	当前页数，在分页查询时使用。如果进行分页查询，页面大小默认为 50，可使用 `size` 调整页面大小。如无需分页可不传（或传 `0`），可获得用户加入的前 5000 个群组列表。
size	Int	否	页面大小，仅在 `page` 传入有效值时生效。默认每页 50 行，最大值 1000。

请求示例

```
POST /user/group/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
groups	String	用户加入的群信息数组。
name	String	群名称。
id	String	群组 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"groups":[
{
"name":"GroupName_1",
"id":"GroupId_1"
},
{
"name":"GroupName_8",
"id":"GroupId_8"
}]
}
```

刷新群组信息

更新时间:2024-08-30

向即时通讯服务端同步指定群组的名称，防止应用中的群组名称与即时通讯服务端的群组名称不一致。

当群组名称变更时，应调用此接口刷新在即时通讯服务端保存的群组名称信息，以保证在触发远程推送（PUSH）提醒的时候，能够正确显示群组名称。

> 提示
>
> 刷新群组信息的接口仅更新远程推送（PUSH）提醒中的携带的群组名称。

刷新群组信息后 5 分钟内生效，生效之后 推送时将显示刷新后的群组名称。

请求方法

POST： https://数据中心域名/group/refresh.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	群组 ID。
groupName	String	是	群组名称。

请求示例

```
POST /group/refresh.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=123&groupName=new
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

禁言指定群成员

更新时间:2024-08-30

群组业务支持将指定用户在群组中禁言。App 可以调用该 API，将用户加入禁言用户列表。具体支持以下能力：

- 在指定的群组中禁言一个或多个用户。
- 如不指定群组（`groupId` 可传空），可将用户在其加入的所有群组中禁言。最多支持设置 100 个用户。如需查询是否已达上限，可调用查询群成员禁言列表接口（`/group/user/gag/list.json`），将 `groupId` 传空。
- 支持设置禁言时长，可以设置为固定时长（上限 43200 分钟）或永久。
- 将用户添加到禁言列表后，禁言状态立即生效。群成员被禁言后可以接收并查看群组中用户聊天信息，但不能通过客户端 SDK 往该群组内发送消息。

用户退出群组后，禁言数据不会清除：

- 被设置为单人禁言（永久）的用户退出群组，再次加入时，禁言状态仍然有效。
- 被设置为单人禁言（指定时长）的用户退出群组，在禁言时长内再次加入时，禁言状态仍然有效。

> 提示
>
> 服务端（Server API）发送群聊消息接口不受群组全体成员禁言状态的限制，被禁言用户可通过 Server API 往该群组中发送消息。详见发送群聊消息。

请求方法

POST： https://数据中心域名/group/user/gag/add.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，每次添加最多不超过 20 个用户。
groupId	String	否	群组 ID，为空时则设置用户在加入的所有群组中都不能发送消息。
minute	String	是	禁言时长，以分钟为单位，最大值为 43200 分钟，为 0 表示永久禁言。

请求示例

```
POST /group/user/gag/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2582&userId=2583&groupId=16&minute=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消指定群成员禁言

更新时间:2024-08-30

在指定的单个群组中，解除一个或多个用户的禁言状态。

如果用户被设置为在全部群组中禁言，可通过将 groupId 传空为用户解除该状态。

请求方法

POST： https://数据中心域名/group/user/gag/rollback.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，每次最多移除 20 个用户。
groupId	String	否	群组 ID，为空时则移除用户在所有群组中的禁言设置。

请求示例

```
POST /group/user/gag/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2582&userId=2583&groupId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询群成员禁言列表

更新时间:2024-08-30

获取在指定群组中被禁言的用户，返回用户 ID 列表。

不指定群组时（groupId 为空）则获取所有群组禁言用户列表。

请求方法

POST： https://数据中心域名/group/user/gag/list.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	否	群组 ID，为空时则获取所有群组禁言用户列表。

请求示例

```
POST /group/user/gag/list.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	Array of objects	禁言成员列表。

返回值	返回类型	说明
users[i].startTime	String	禁言开始时间。精确到秒，格式为 YYYY-MM-DD HH:MM:SS，例如 2022-09-25 16:12:38。注意：time 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 time 为北京时间。如您的 App 业务使用海外数据中心，则 time 为 UTC 时间。
users[i].time	String	解禁时间。精确到秒，格式为 YYYY-MM-DD HH:MM:SS，例如 2022-09-25 16:12:38。注意：time 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 time 为北京时间。如您的 App 业务使用海外数据中心，则 time 为 UTC 时间。
users[i].userId	String	群成员的用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"users": [
{
"startTime": "2024-02-07 18:25:12",
"time": "2024-02-07 18:55:12",
"userId": "u02"
},
{
"startTime": "2024-02-07 18:25:12",
"time": "2024-02-07 18:55:12",
"userId": "u01"
}
]
}
```

设置群组全体禁言

更新时间:2024-08-30

群组业务支持将群成员全体禁言。App 可以调用该 API，将指定的群组加入服务端的全体禁言群组列表。

- 将群组加入全体禁言列表后，群组全体禁言立即生效。该群组的所有成员均不能通过客户端 SDK 往该群组内发送消息。
- 即使群组解散再创建，全体禁言状态仍然有效。如有需要，请将指定群组移出全体禁言列表。详见取消群组全体禁言。

> 提示
>
> - 在群禁言用户白名单中用户仍可通过客户端 SDK 往该群组中发消息。详见 加入群组全体禁言白名单。
> - 服务端（Server API）发送群聊消息接口不受群组全体成员禁言状态的限制，被禁言用户可通过 Server API 往该群组中发送消息。

请求方法

POST： https://数据中心域名/group/ban/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	群组 ID，支持一次设置多个，最多不超过 20 个。

请求示例

```
POST /group/ban/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=XklWnVzXm&groupId=LklWnVzXq
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消群组全体禁言

更新时间:2024-08-30

解除指定的一个或多个群组的全体成员禁言状态。

请求方法

POST： https://数据中心域名/group/ban/rollback.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	群组 ID，支持一次设置多个，最多不超过 20 个。

请求示例

```
POST /group/ban/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=L6c85orxt&groupId=X6c95orxt
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询群组全体禁言

更新时间:2024-08-30

查询群组是否设置为全部成员禁言，具体功能如下：

- 传入有效的群组 ID 时，查询指定的单个群组或多个群组是否已设置全部成员禁言
- 不传群组 ID 时，查询 App Key 下所有已被设置为全体成员禁言的群组列表

请求方法

POST： https://数据中心域名/group/ban/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	否	群组 ID。单次可查询指定单个或多个群组，单次查询最多不超过 20 个群组。不传时，表示查询所有设置禁言的群组列表。

请求示例

```
POST /group/ban/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
groupinfo	String[]	禁言群组信息数据。
groupId	String	群组 ID。

返回值	返回类型	说明
stat	Int	禁言状态，0 表示为未禁言、1 表示为禁言。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"groupinfo":[
{
"groupId": "groupId1",
"stat":1
},
{
"groupId": "groupId2",
"stat":1
}
]
}
```

加入群组全体禁言白名单

更新时间:2024-08-30

添加一个或多个群成员到指定群组的全体成员禁言白名单。

- 加入群组的禁言白名单后，在群组已被设置为全体禁言时，该成员仍可通过客户端 SDK 往该群组发送消息。
- 在白名单的用户如果退出群组，再次加入，白名单状态仍然有效。
- 群组解散再创建，白名单状态仍然有效。

> 提示
>
> 群组单人禁言功能具有更高优先级。如单人禁言已生效，即使用户已被添加到全体禁言白名单，仍无法往群组发送消息。

请求方法

POST： https://数据中心域名/group/user/ban/whitelist/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，支持一次添加多个用户，最多不超过 20 个。
groupId	String	是	群组 ID。

请求示例

```
POST /group/user/ban/whitelist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=2MqOJa1Un&userId=3212&userId=2583
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移出群组全体禁言白名单

更新时间:2024-08-30

从指定群组的群禁言白名单中移除一个或多个用户。

请求方法

POST： https://数据中心域名/group/user/ban/whitelist/rollback.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，支持同时移除多个用户，每次最多不超过 20 个。
groupId	String	是	群组 ID。

请求示例

```
POST /group/user/ban/whitelist/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2582&userId=2583&groupId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询群组全体禁言白名单

更新时间:2024-08-30

获取指定群组的群全体禁言白名单。

请求方法

POST： https://数据中心域名/group/user/ban/whitelist/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	群组 ID。

请求示例

```
POST /group/user/ban/whitelist/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
userIds	String[]	用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"userIds":["2111","2582"]}
```

超级群概述

更新时间:2024-08-30

超级群（UltraGroup）提供了一种新的群组业务形态。超级群不设置群成员人数上限，允许用户在超级社群中建立社交关系、在海量信息中聚焦自己感兴趣的内容，帮助开发者打造高用户黏性的群体。超级群组成员最多可加入 100 个超级群，每个超级群下的不同频道之间共享一份超级群成员关系。App 内的超级群数量没有限制。

服务端一般用 `busChannel` 参数表示超级群频道 ID，对应客户端的 `channelId` 参数。

开通服务

超级群功能需要在控制台超级群服务 页面开通。

如何使用频道

超级群支持在群会话中创建独立的频道（`busChannel`），超级群的会话、消息、未读数等消息数据和群组成员支持分频道进行聚合，各个频道之间消息独立。

频道按类型区分为公有频道与私有频道。公有频道对所有超级群成员开放（无需加入）。该超级群的所有成员都会接收公有频道下的消息。私有频道仅对该频道成员列表上的用户开放。有关私有频道的详细介绍，可参见超级群私有频道概述。

超级群业务提供一个 ID 为 `RCDefault` 的默认频道。`RCDefault` 频道对所有超级群成员开放，不可转为私有频道。

对于 App 业务来说，如果仅需实现类似群聊的业务，可以利用超级群无成员上限的特性构建大于 3000 人的超大群。这种场景下，可以让所有消息都在 `RCDefault` 默认频道中进行收发。建议在调用客户端、服务端 API 时指定频道 ID 为 `RCDefault`。

如果仅需实现类似 Discord 类业务，通过超级群频道功能构建子社区，推荐全部使用您自行创建的频道实现您的业务特性。默认频道（`RCDefault`）与自建频道的行为存在差异，全部使用自建频道可避免这种差异在实现 App 业务逻辑时造成限制。

> 提示
>
> 如果您的 App / 环境在 2022.10.13 日之前开通超级群服务，则您的超级群服务中不存在 `RCDefault` 频道。在调用客户端、服务端 API 时如果不指定频道 ID，一般仅作用于不属于任何频道的消息，具体行为需参见各功能文档。即时通讯服务支持客户调整服务至最新行为。该行为调整将影响客户端、服务端收发消息、获取会话、清除历史消息、禁言等多个功能。如有需要，请提交工单咨询详细方案。

超级群管理接口

即时通讯服务不会托管用户，也不管理群组的业务逻辑，因此超级群的业务逻辑全部需要在 App 服务器进行实现。

对于客户端开发人员来说，创建群组、频道等基础管理操作只需要与 App 后端交互即可。App 后端需要调用相应的即时通讯服务端 API（Server API）接口相关接口创建超级群、创建频道等其他管理操作。

下表列出了超级群部分基础管理接口。更多接口可参见API 接口列表。

功能分类	功能描述	即时通讯服务端 API
创建、解散超级群	提供创建者用户 ID、超级群 ID、和群名称，向即时通讯服务端申请建群。如解散超级群，则群成员关系不复存在。	创建超级群、解散超级群
加入、退出超级群	加入超级群后，默认可查看入群以后产生的新消息。退出超级群后，不再接受该群的新消息。	加入超级群、退出超级群
修改即时通讯服务端的超级群信息	修改在即时通讯的推送服务中使用的超级群信息。	更新超级群信息
创建、删除群频道	在超级群会话中创建独立沟通的频道。如删除频道，将无法在频道中发送消息。	创建频道、删除频道
查询群频道列表	加入超级群后，默认可查看入群以后产生的新消息。退出超级群后，不再接受该群的新消息。	查询频道列表
变更群频道类型	超级群频道可以随时切换为公有频道或私有频道。	变更频道类型
添加、删除私有频道成员	将超级群成员加入或移出指定频道的私有频道成员列表。在频道类型为私有频道时启用该成员列表的数据。	添加私有频道成员、删除私有频道成员

创建超级群

更新时间:2024-08-30

创建单个超级群。App 下的超级群数量没有限制。

创建超级群流程

即时通讯服务不托管超级群的业务逻辑，因此超级群的业务逻辑全部在 App 服务器进行实现，对于客户端开发人员来说，创建群组只需要与 App 后端交互即可。

如需在 App 客户端创建超级群，App 应请求自身业务服务端，由 App 后端调用即时通讯服务端 API，创建超级群。超级群 ID 须由 App 服务器自行生成。创建成功后，同时返回给客户端。

时序图如下：

重要

- 如果您的 App / 环境在 2022.10.13 日之后开通超级群服务，在创建超级群时，即时通讯服务默认会同时创建一个频道 ID 为 `RCDefault` 的默认频道。`RCDefault` 频道对所有超级群成员开放，不可转为私有频道。
- 对于 App 业务来说，如果仅需构建一个超大群，可以让所有消息都在 `RCDefault` 默认频道中进行收发。在调用客户端、服务端 API 时，强烈建议显式指定频道 ID 为 `RCDefault`。如果需要通过超级群频道功能构建子社区，推荐全部采用自建频道的方式实现您的业务特性。详见超级群概述。

请求方法

POST： https://数据中心域名/ultragroup/create.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	需要加入的用户 ID，创建后同时加入超级群。仅支持传入一个用户 ID。
groupId	String	是	超级群 ID，最大长度 64 个字符。支持大小写英文字母与数字的组合。

参数	类型	必传	说明
groupName	String	是	超级群 ID 对应的名称，用于在发送群组消息显示远程 Push 通知时使用，如超级群名称改变需要调用刷新超级群信息接口同步。

请求示例

```
POST /ultragroup/create.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&groupId=abcdefg&groupName=群名称
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

加入超级群

更新时间:2024-08-30

将单个或多个用户加入指定超级群。

- 单个超级群无成员上限限制，但每个用户最多可加入 100 个超级群。
- 用户加入该群组后，用户将可以收到该群的消息。
- 新入群的用户默认只能查看加入成功后该超级群产生的消息。如果需要允许用户查看加入超级群之前的历史消息，可在控制台超级群服务 页面开通新用户加入群后是否拉取加入群之前的消息。

加入超级群流程

即时通讯服务不托管超级群的业务逻辑，因此超级群的业务逻辑全部在 App 服务器进行实现，对于客户端开发人员来说，加入群组只需要与 App 后端交互即可。

如果需要在 App 客户端加入超级群，App 应请求自身业务服务端，由 App 后端调用即时通讯服务端 API，将用户加入超级群。超级群 ID 由 App 服务器自行维护生成。加入成功后，同时返回给客户端。

时序图如下：

请求方法

POST： https://数据中心域名/ultragroup/join.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	要加入群的用户 ID。单次仅支持 1 个用户。
groupId	String	是	要加入的群 ID。

请求示例

```
POST /ultragroup/join.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jzk4j6x5&groupId=abcdefg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

退出超级群

更新时间:2024-08-30

将用户从指定超级群中移除。用户退出超级群后，不再接收该群组的消息。

退出超级群流程

即时通讯服务不托管超级群的业务逻辑，因此超级群的业务逻辑全部在 App 服务器进行实现，对于客户端开发人员来说，将用户移出超级群只需要与 App 后端交互即可。

如果需要在 App 客户端将用户退出超级群，App 需要请求 App 服务器，由 App 服务器调用即时通讯服务端 API，退出超级群。

时序图如下：

请求方法

POST： https://数据中心域名/ultragroup/quit.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	要退出群的用户 ID。
groupId	String	是	要退出的群 ID。

请求示例

```
POST /ultragroup/quit.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=jlk456j5&groupId=abcdefg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

解散超级群

更新时间:2024-08-30

解散指定单个超级群。

- 将群组解散，所有用户都无法再接收该群的消息。
- 超级群解散后，群的成员关系已经不存在，通过客户端无法在获取群存储在服务端的历史消息，保存到客户端本地的历史消息仍然可以查看。

解散超级群流程

即时通讯服务不托管超级群的业务逻辑，因此超级群的业务逻辑全部在 App 服务器进行实现，对于客户端开发人员来说，解散超级群只需要与 App 后端交互即可。

如果需要在 App 客户端解散超级群，App 需要请求 App 服务器，由 App 服务器调用即时通讯服务端 API，解散超级群。

时序图如下：

请求方法

POST： https://数据中心域名/ultragroup/dis.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	要解散的超级群 ID。

请求示例

```
POST /ultragroup/dis.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

刷新超级群信息

更新时间:2024-08-30

当超级群名称变更时，App 业务服务端应该调用此接口刷新在即时通讯服务端保存的超级群信息，以保证在触发远程推送（PUSH）提醒的时候，能够正确显示相关内容信息。

刷新超级群信息后 5 分钟内生效，Push 推送时将显示刷新后的超级群名称。

请求方法

POST： https://数据中心域名/ultragroup/refresh.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	群组 ID。
groupName	String	是	群组名称。

请求示例

```
POST /ultragroup/refresh.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=123&groupName=new
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

创建频道

更新时间:2024-08-30

在指定超级群下创建独立的沟通频道（busChannel）。一个超级群中最多可以创建 50 个频道。

超级群（UltraGroup）提供了一种新的群组业务形态，支持在群会话中创建独立的频道，超级群的消息数据（会话、消息、未读数）和群组成员支持分频道进行聚合，各个频道之间消息独立。

频道分为公有频道和私有频道，在创建频道时可指定频道类型，也可通过 API 接口进行切换。每个频道共用所属群的成员关系，不同频道间的消息相互隔离。

- **公有频道**：公有频道对所有超级群成员开放（无需加入）。所有超级群成员可随意在任何公有频道中发送消息。所有群成员都会接收公有频道下的消息内容，暂不支持设置不接收指定公有频道的消息。
- **私有频道**：用户必须已被添加到私有频道成员列表，才能在私有频道中收发消息。

方法说明

POST： https://数据中心域名/ultragroup/channel/create.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID，请确保该超级群 ID 已存在。
busChannel	String	是	频道 ID，支持大小写字母、数字的组合方式，长度不超过 20 个字符。
type	Number	否	频道类型。0 表示公有频道（默认）。1 表示私有频道。

请求示例

```
POST /ultragroup/channel/create.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&busChannel=channel001&type=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

删除频道

更新时间:2024-08-30

在指定超级群下删除（解散）频道。删除频道后，超级群用户将无法往频道中发送消息。

- 删除频道后，频道内的历史消息不会被删除，如需恢复已被删除的频道，可使用创建频道接口并传入相同的频道 ID。
- 删除频道时，会删除该频道下的私有频道成员列表。

方法说明

POST： https://数据中心域名/ultragroup/channel/del.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID。
busChannel	String	是	频道 ID，支持大小写字母、数字的组合方式，长度不超过 20 个字符。

请求示例

```
POST /ultragroup/channel/del.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&busChannel=channel001
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询频道列表

更新时间:2024-08-30

获取指定超级群下的频道。一个超级群最多可以有 50 个频道。

方法说明

POST： https://数据中心域名/ultragroup/channel/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID。
page	Int	否	当前页数，默认为 1。
limit	Int	否	每页条数，默认为 20 条，最大不超过 100 条。

请求示例

```
POST /ultragroup/channel/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&page=1&size=20
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常
channelList	JSONArray	频道列表
channelId	String	频道 ID
createTime	String	频道创建时间

返回值	返回类型	说明
type	Number	频道类型。`0` 表示公有频道。`1` 表示私有频道。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"channelList": [
{
"channelId": "channel001",
"createTime": "2022-02-14 11:19:42",
"type" : 0
}
]
}
```

获取指定超级群消息内容

更新时间:2024-08-30

根据消息 UID 获取超级群消息，单次最多获取 20 条。

App 后端使用该接口查询消息的内容与消息扩展数据。如果消息已超过超级群历史消息记录存储期限（默认 7 天）则返回内容为空。查询消息所需传入的消息全局唯一 ID（消息 UID）与频道 ID 可以由 App 自动从客户端上传至 App 后端，或通过即时通讯服务提供的全量消息路由等回调服务获取。

方法说明

POST： https://数据中心域名/ultragroup/msg/get.json

频率限制： 每秒钟限 5 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
msgs	Array of Objects	是	消息的查询参数，单次请求最多获取 20 条消息。
msgs[i].msgUID	String	是	全局唯一 ID，即消息 UID。
msgs[i].busChannel	String	否	消息所在的超级群频道 ID。

请求示例

```
POST /ultragroup/msg/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxabx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e739219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=BWSZNmXBD&msgs=%5B%7B%22busChannel%22%3A%22001%22%2C%22msgUID%22%3A%22C16R-VBGG-1IE5-
SD0C%22%7D%2C%7B%22busChannel%22%3A%22002%22%2C%22msgUID%22%3A%22C16R-VBGG-1IE5-SD5C%22%7D%5D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
data	JSONArray	消息数组。
data[i].fromUserId	String	发送人用户 ID。
data[i].groupId	String	超级群 ID。
data[i].sentTime	Number	消息发送时间。Unix 时间戳，单位为毫秒。
data[i].busChannel	String	频道 ID。
data[i].msgUID	String	全局唯一消息 ID，即消息 UID。
data[i].objectName	String	消息类型的唯一标识。
data[i].content	String	消息的内容。
data[i].expansion	Boolean	是否为扩展消息。
data[i].extraContent	String	消息扩展的内容，JSON 结构的 Key、Value 对，如："{\"key1\":{\"v\":\"value\",\"ts\":110908544521}}"。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符+ = - _ 的组合方式，不支持汉字。Value 最大 4096 个字符。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"data":[
{
"fromUserId":"user1",
"groupId":"g1",
"sentTime":1654591498853,
"busChannel":"001",
"msgUID":"C16R-VBGG-1IE5-SD0C",
"objectName":"RC:TxtMsg",
"content":"哈喽",
"expansion":"false",
"extraContent":null
}
]
}
```

修改超级群消息

更新时间:2024-08-30

修改一条已发送成功的超级群消息。消息类型无法修改。如果改前为文本消息，则传入的新消息内容必须同样为文本消息。

请求方法

POST： https://数据中心域名/ultragroup/msg/modify.json

频率限制： 每分钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	消息所属的超级群 ID。
busChannel	String	否	消息所属的超级群的会话频道 ID。仅在消息属于超级群默认频道（`RCDefault`）时可以不传该参数。
fromUserId	String	是	消息发送者 ID。
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
content	String	是	修改后的消息内容，单条消息最大 128k。消息类型无法修改。如果改前为文本消息内容体，则传入的新消息内容体必须同样为文本消息内容体。 • **内置消息类型**：将消息内容体 JSON 对象序列化为 JSON 字符串传入。消息内容 JSON 结构体详见用户内容类消息格式或其他内置消息类型的消息内容格式。 例如，文本消息内容 JSON 结构体内部包含 content 字段（此为 JSON 结构体内的 key 值，注意区分），则需要将 `{"content":"Hello world!"}` 序列化后的结果作为此处 `content` 字段的值。 • **自定义消息类型（`objectName` 字段必须指定为自定义消息类型）**：如果发送自定义消息，该参数可自定义格式，不限于 JSON。

请求示例

```
POST /ultragroup/msg/modify.json HTTP/1.1
Host: api-cn.ronghub.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307338a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=BWSZNmXBD&fromUserId=2191&msgUID=C16R-VBGG-1IE5-
SD0C&content=%7B%22content%22%3A%22hello%22%2C%22extra%22%3A%22helloExtra%22%7D&busChannel=channel1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

搜索超级群消息

更新时间:2024-08-30

即时通讯服务端默认保存 7 天内 App 下超级群所有会话频道的历史消息记录。您可以使用服务端 API 主动获取指定时段内的频道历史消息。支持仅查询由指定用户 ID 发送的消息。

该接口一般由超级群 App 运营方使用，可用于实现后台搜索消息。具体可用于查询：

- 指定时段内超级群频道中的消息
- App 的指定用户发送的消息

> 提示
>
> 即时通讯服务同时提供服务端回调服务，支持将消息副本实时抄送给您指定的应用服务器。开通及配置方式详见详见全量消息路由。

可查询范围

- **时间范围**：API 接口支持获取最大时间跨度不超过 14 天的频道的消息。开通超级群业务后，超级群消息云端存储默认设置为 7 天，即仅保存 7 天的历史消息存储。支持付费开通为 3 个月、6 个月、 1 年。如何调整保存时长详见开通超级群服务 。
- **频道类型**：API 接口支持查询公有频道和私有频道的历史消息。公有频道指 App 自建频道和超级群服务默认的 `RCDefault` 频道。API 不对查询私有频道历史消息作额外限制，App 后端可能需要自行区分频道类型。关于私有频道的更多说明请参见私有频道概述。
- **限制条件**：超级群业务从 2022.10.13 日开始提供 ID 为 `RCDefault` 的默认频道。在此之前开通超级群业务的 App/环境中不存在 `RCDefault` 频道。即时通讯服务支持客户调整服务至最新行为。该行为调整将影响客户端、服务端收发消息、获取会话、清除历史消息、禁言等多个功能。如有需要，请提交工单咨询详细方案。如果在调用客户端、服务端 API 发送消息时不指定频道 ID，消息不属于任何频道，无法通过此 API 查询。

方法说明

POST： https://数据中心域名/ultragroup/hismsg/query.json

频率限制： 每分钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID，请确保该超级群 ID 已存在。
busChannel	String	是	频道 ID。
startTime	Number	是	查询开始时间。Unix 时间戳（毫秒）。
endTime	Number	是	查询结束时间。Unix 时间戳（毫秒）。需要保证比 `startTime` 大，且两者之间时间跨度最大14 天。
fromUserId	String	否	发送者用户 ID。不传该字段，查所有用户发送的历史消息。如果传入该参数，表示只查该用户 ID 发的历史消息。
pageSize	Number	否	分页返回的页面大小。默认 20 条，最大 100 条。

请求示例

```
POST /ultragroup/hismsg/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=wxlGroup&busChannel=wxlBusChannel&startTime=1666251303434&endTime=1666251311996&fromUserId=123&p
ze=10
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
data	Array of objects	查询结果。按消息时间戳升序排列。
data[i].groupId	String	超级群 ID。
data[i].busChannel	String	超级群频道 ID。
data[i].fromUserId	String	消息发送方用户 ID。
data[i].msgUID	String	消息 UID。
data[i].msgTime	Number	发送消息的时间戳。Unix 时间戳（毫秒）。
data[i].objectName	String	消息类型。详见消息类型概述。
data[i].conversationType	Number	会话类型。超级群的会话类型为 `10`。
data[i].content	String	消息内容，JSON 格式。具体结构可通过 `objectName` 字段在 消息类型概述 中查询。
data[i].expansion	Boolean	消息是否已被设置为可扩展消息。`true` 表示可扩展。`false` 表示不可扩展。
data[i].extraContent	String	消息扩展的内容，JSON 结构的 Key、Value 对，如："{\"key1\":{\"v\":\"value\",\"ts\":110908544521}}"。Key 最大 32 个字符，支持大小写英文字母、数字、特殊字符+ = - _ 的组合方式，不支持汉字。Value 最大 4096 个字符。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"data": [
{
"groupId": "wxlGroup",
"busChannel": "wxlBusChannel",
"fromUserId": "123",
"msgUID": "C3UH-8KMU-O038-22SK",
"msgTime": 1666251311995,
"objectName": "RC:TxtMsg",
"conversionType": 10,
"content": "{\"content\":\"abce\"}",
"expansion": false,
"extraContent": ""
},
{
"groupId": "wxlGroup",
"busChannel": "wxlBusChannel",
"fromUserId": "123",
"msgUID": "C3UH-8IK2-G018-22SK",
"msgTime": 1666251303434,
"objectName": "RC:TxtMsg",
"conversionType": 10,
"content": "{\"content\":\"abc\"}",
"expansion": false,
"extraContent": ""
}
]
}
```

搜索超级群消息上下文

更新时间:2024-08-30

即时通讯服务端会保存（默认 7 天）App 下超级群所有会话频道的历史消息记录。您可以调用服务端 API 通过消息 UID（msgUID）主动获取超级群频道内指定消息的历史消息上下文。

该接口一般由超级群 App 运营方使用，可用于实现后台搜索消息。

> 提示
>
> 即时通讯服务同时提供服务端回调服务，支持将消息副本实时抄送给您指定的应用服务器。开通及配置方式详见全量消息路由。

可查询范围

- **数量范围**：API 接口支持通过指定消息 UID 搜索 100 条上下文消息，其中上文消息最多 50 条，下文消息最多 50 条。
- **时间范围**：API 接口支持获取在历史消息存储服务中的上下文历史消息。如果消息已超过保存期限，则无法获取。开通超级群业务后，超级群消息云端存储默认设置为 7 天，即仅保存 7 天的历史消息存储。支持付费开通为 3 个月、6 个月、 1 年。如何调整保存时长详见开通超级群服务 。
- **频道类型**：API 接口支持查询公有频道和私有频道的历史消息。公有频道指 App 自建频道和超级群服务默认的 `RCDefault` 频道。API 不对查询私有频道历史消息作额外限制，App 后端可能需要自行区分频道类型。关于私有频道的更多说明请参见私有频道概述。
- **限制条件**：超级群业务从 2022.10.13 日开始提供 ID 为 `RCDefault` 的默认频道。在此之前开通超级群业务的 App/环境中不存在 `RCDefault` 频道。即时通讯服务支持客户调整服务至最新行为。该行为调整将影响客户端、服务端收发消息、获取会话、清除历史消息、禁言等多个功能。如有需要，请提交工单咨询详细方案。如果在调用客户端、服务端 API 发送消息时不指定频道 ID，消息不属于任何频道，无法通过此 API 查询。

方法说明

POST： https://数据中心域名/ultragroup/hismsg/msgid/query.json

频率限制： 每分钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID，请确保该超级群 ID 已存在。

参数	类型	必传	说明
busChannel	String	是	频道 ID。
msgUID	String	是	消息的 UID。返回结果中会包含该 `msgUID` 的消息。例如，该接口默认查询该消息前面 10 条和后面 10 条消息，默认情况下返回 21 条消息。
prevNum	Number	否	需要查询的上文消息数量。例如传入 `20`，即表示需要获取 `msgUID` 前的 20 消息数量。取值范围：`0-50`，默认 `10`。`0` 表示不需要获取 `msgUID` 前的消息。
lastNum	Number	否	需要查询的下文消息数量。例如传入 `20`，即表示需要获取 `msgUID` 前的 20 消息数量。取值范围：`0-50`，默认 `10`。`0` 表示不需要获取 `msgUID` 后的消息。

请求示例

```
POST ultragroup/hismsg/msgid/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=wxlGroup&busChannel=wxlBusChannel&msgUID=AAAA-BBBB-CCCC-DDDD&prevNum=10&lastNum=10
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
data	Array of objects	查询结果。按消息时间戳升序排列。
data[i].groupId	String	超级群 ID。
data[i].busChannel	String	超级群频道 ID。
data[i].fromUserId	String	消息发送方用户 ID。
data[i].msgUID	String	消息 UID。
data[i].msgTime	Number	发送消息的时间戳。Unix 时间戳（毫秒）。
data[i].objectName	String	消息类型。详见消息类型概述。
data[i].conversationType	Number	会话类型。超级群的会话类型为 `10`。
data[i].content	String	消息内容，JSON 格式。具体结构可通过 `objectName` 字段在 消息类型概述 中查询。
data[i].expansion	Boolean	消息是否已被设置为可扩展消息。`true` 表示可扩展。`false` 表示不可扩展。
data[i].extraContent	String	消息扩展的内容，JSON 结构的 Key、Value 对，如："{\"key1\":{\"v\":\"value\",\"ts\":110908544521}}"。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符`+ = - _` 的组合方式，不支持汉字。Value 最大 4096 个字符。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"data": [
{
"groupId": "wxlGroup",
"busChannel": "wxlBusChannel",
"fromUserId": "123",
"msgUID": "C3UH-8KMU-0038-22SK",
"msgTime": 1666251311995,
"objectName": "RC:TxtMsg",
"conversionType": 10,
"content": "{\"content\":\"abce\"}",
"expansion": false,
"extraContent": ""
},
{
"groupId": "wxlGroup",
"busChannel": "wxlBusChannel",
"fromUserId": "123",
"msgUID": "C3UH-8IK2-G018-22SK",
"msgTime": 1666251303434,
"objectName": "RC:TxtMsg",
"conversionType": 10,
"content": "{\"content\":\"abc\"}",
"expansion": false,
"extraContent": ""
}
]
}
```

设置超级群消息扩展

更新时间:2024-08-30

超级群消息扩展功能可用于为原始消息增加状态标识（扩展数据为 KV 键值对），提供添加、删除、查询扩展信息的接口。

使用消息扩展的场景

对原始消息增加状态标识的业务场景均可使用消息扩展。以下是业务场景示例：

- 消息评论。可通过设置原始消息扩展信息的方式添加评论信息。
- 礼物领取、订单状态变化。通过此功能可改变消息显示状态。例如：向用户发送礼物，默认为未领取状态，用户点击后可设置消息扩展为已领取状态。

消息扩展支持的会话类型

超级群消息扩展接口仅可用于超级群会话类型。

注意事项

消息扩展操作实际上是通过一条特殊的消息实现的，可认为是“扩展操作消息”。如需保证在长期（超过 7 天）不登录客户端等情况下仍可获取扩展操作消息，建议在超级群服务 页面延长超级群消息云端存储的时长。

前提条件

原始消息必须在发送时已设置为可扩展。每次最多可以设置 100 个扩展属性信息，最多可设置 300 个。

> 提示
>
> 例如，通过即时通讯服务端 API 发送消息时，需要设置 `expansion` 为 `true`，该条消息才能支持扩展信息设置。客户端 SDK 从特定版本开始支持该功能（移动端 4.0.3 版本、Web 端 3.0.7 版本）

请求方法

POST： https://数据中心域名/ultragroup/message/expansion/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

> 提示
>
> 扩展信息设置超过 300 个时，设置不成功，返回设置失败错误码，如：已设置 290 个，再次设置 100 个时，本次请求全部失败，设置不成功。

参数	类型	必传	说明
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
userId	String	是	操作者用户 ID，即需要为指定消息（msgUID）设置扩展信息的用户 ID。
busChannel	String	否	超级群频道 ID。具体使用要求如下： • 如果发送消息时指定了频道 ID，则必传频道 ID，否则无法成功设置扩展。 • 如果发送消息时未指定频道 ID，则不可传入频道 ID，否则无法成功设置扩展。 客户端发送超级群消息时，频道 ID 对应字段名称为 channelId。
groupId	String	是	超级群 ID。
extraKeyVal	Object	是	消息扩展的内容，JSON 结构，以 Key、Value 的方式进行设置，如：{"type":"3"}。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符+ = - _ 的组合方式，不支持汉字。Value 最大 4096 个字符。单条消息可设置 300 个扩展信息，一次最多可以设置 100 个。

请求示例

```
POST /ultragroup/message/expansion/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

msgUID=BRGM-DEN2-01E4-
BN66&userId=WNYZbMqpH&groupId=tjw3zbMrU&busChannel=&extraKeyVal=%7B%22type%22%3A%223%22%7D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

删除超级群消息扩展

更新时间:2024-08-30

根据消息 ID 删除指定消息的扩展信息，一次可删除多个扩展 KV 键值对。

超级群消息扩展功能可用于为原始消息增加状态标识（扩展数据为 KV 键值对），提供添加、删除、查询扩展信息的接口。

请求方法

POST： https://数据中心域名/ultragroup/message/expansion/delete.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
userId	String	是	操作者用户 ID，即需要为指定消息（`msgUID`）删除扩展信息的用户 ID。
busChannel	String	否	频道 Id 字段。
groupId	String	是	超级群频道 ID。具体使用要求如下： • 如果发送消息时指定了频道 ID，则必传频道 ID，否则无法成功删除扩展。 • 如果发送消息时未指定频道 ID，则不可传入频道 ID，否则无法成功删除扩展。 客户端发送超级群消息时，频道 ID 对应字段名称为 `channelId`。
extraKey	String	是	需要删除的扩展信息的 Key 值，一次最多可以删除 100 个扩展信息

请求示例

```
POST /ultragroup/message/expansion/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1585127132438
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

msgUID=BRGM-DEN2-01E4-BN66&userId=tjw3zbMrU&groupId=WNYZbMqpH&busChannel=&extraKey=%5B%2211111%22%5D
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

获取超级群消息扩展

更新时间:2024-08-30

根据消息 ID 获取指定消息的扩展信息。

超级群消息扩展功能可用于为原始消息增加状态标识（扩展数据为 KV 键值对），提供添加、删除、查询扩展信息的接口。

请求方法

POST： https://数据中心域名/ultragroup/message/expansion/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
msgUID	String	是	消息唯一标识 ID，可通过全量消息路由功能获取。详见全量消息路由。
busChannel	String	否	超级群频道 ID。具体使用要求如下： • 如果发送消息时指定了频道 ID，则必传频道 ID，否则无法成功获取扩展。 • 如果发送消息时未指定频道 ID，则不可传入频道 ID，否则无法成功获取扩展。 客户端发送超级群消息时，频道 ID 对应字段名称为 `channelId`。
groupId	String	是	超级群 ID。
pageNo	Int	否	页数，默认返回 300 个扩展信息。

请求示例

```
POST /ultragroup/message/expansion/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/x-www-form-urlencoded

msgUID=BRGM-DEN2-01E4-BN66&groupId=WNYZbMqpH&busChannel=abc
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
extraContent	Object	消息扩展的内容，JSON 结构，以 Key、Value 的方式进行设置，如：`{"type":"3"}`。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符+ = - _ 的组合方式，不支持汉字。Value 最大 4096 个字符。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"extraContent": {
"kkk314": {
"v": "vvv314",
"ts": 33831605613
},
"kkk313": {
"v": "vvv313",
"ts": 33831605588
}
}
}
```

查询用户是否为群成员

更新时间:2024-08-30

查询指定用户是否在指定的超级群中。

请求方法

POST： https://数据中心域名/ultragroup/member/exist.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	是否必传	说明
userId	String	必填	要查询的用户 ID
groupId	String	必填	要查询的超级群 ID

请求示例

```
POST /ultragroup/member/exist.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=1&groupId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	200：成功。
status	Boolean	用户是否在超级群中。`true` 表示在超级群中，`false` 表示不在超级群中。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"status":true}
```

免打扰功能概述

更新时间:2024-08-30

即时通讯业务支持多维度、多级别精细化的免打扰设置。

IM 的免打扰设置维度

即时通讯业务支持对免打扰功能进行多维度地控制。开发者可从 App Key、特定的 IM 细分业务（目前仅超级群）、用户级别进行免打扰功能配置。

> 提示
>
> - Android/iOS 客户端 SDK 从 5.2.2 开始提供对以下维度免打扰设置的完整支持。
> - Web 客户端 SDK 从 5.3.0 开始提供对以下维度免打扰设置的完整支持。

免打扰配置的维度	适用场景	说明	服务端 API
App Key 级设置	单聊、群聊、系统会话、超级群	以 App Key 为单位，设置整个应用的默认免打扰级别。默认未设置，等同于全部消息都接收通知。该级别的配置暂未在控制台开放，如有需要，请提交工单。	服务端不提供该 API。
超级群默认设置（全体群成员）	仅限超级群业务	可为指定的超级群的设置默认的免打扰级别，对全体群成员生效。	详见「超级群管理」下的设置群/频道默认免打扰
超级群频道默认设置（全体群成员）	仅限超级群业务	可为指定的超级群频道设置默认的免打扰级别，对全体群成员生效。	详见「超级群管理」下的设置群/频道默认免打扰
会话类型设置（用户级）	单聊、群聊、系统会话、超级群	允许用户设置指定类型会话的免打扰级别。	详见设置会话类型免打扰。
会话的设置（用户级）	单聊、群聊、系统会话、超级群	允许用户设置指定会话的免打扰级别。	详见设置会话免打扰。
超级群会话频道的设置（用户级）	仅限超级群业务	允许用户设置指定超级群频道的免打扰级别。	详见设置会话免打扰。
应用全局设置（用户级）	单聊、群聊、系统会话、超级群	允许用户设置指定时段内全局免打扰级别。	详见 设置用户免打扰时段。

免打扰设置的优先级

- 针对单聊、群聊、系统会话，即时通讯服务端会遵照以下顺序搜索免打扰配置。优先级从左至右依次降低，以优先级最高的配置为准判断是否需要触发推送：

  全局免打扰设置（用户级） > 指定会话的免打扰设置（用户级） > 指定会话类型的免打扰设置（用户级） > App 级的免打扰设置

- 针对超级群会话，即时通讯服务端会遵照以下顺序搜索免打扰配置。优先级从左至右依次降低，以优先级最高的配置为准判断是否需要触发推送：

  全局免打扰设置（用户级）> 指定超级群频道的免打扰设置（用户级）> 指定会话的免打扰设置（用户级）> 指定会话类型的免打扰设置（用户级）> 指定超级群频道的默认免打扰设置（超级群全员）> 指定超级群的免打扰设置（超级群全员）> App 级的免打扰设置

设置群/频道默认免打扰

更新时间:2024-08-30

超级群业务支持为指定的群，或群频道设置默认免打扰逻辑。默认免打扰逻辑对所有群成员生效，一般由超级群的管理员进行设置。

注意事项

- 在即时通讯服务端判断是否需要推送超级群消息时，指定的超级群，或群频道的默认免打扰配置优先级均低于用户级别配置。如果存在任何用户级别的免打扰配置，则优先以用户级别免打扰配置为准进行判断。

> 即时通讯业务免打扰功能的 用户级别设置 支持控制指定的单聊会话、群聊会话、超级群会话、超级群频道的免打扰级别，并可设置全局免打扰的时间段与级别。用户级别设置优先级如下：全局免打扰 > 按频道设置的免打扰 > 按会话设置的免打扰 > 按会话类型设置的免打扰。详见超级群免打扰功能概述。

- 为指定的超级群设置的默认免打扰逻辑，自动适用于群下的所有频道。如果针对频道另行设置了默认免打扰逻辑，则以该频道的默认设置为准。

请求方法

POST： https://数据中心域名/ultragroup/notdisturb/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	否	频道 ID，不传时为群的默认免打扰设置

参数	类型	必传	说明
unpushLevel	Int	是	• `-1`：全部消息通知 • `0`：未设置（用户未设置时为此状态，为全部消息都通知，在此状态下，如设置了超级群默认状态以超级群的默认设置为准） • `1`：仅针对 @ 消息进行通知，包括 @指定用户 和 @所有人 • `2`：仅针对 @ 指定用户消息进行通知，且仅通知被 @ 的指定的用户进行通知。 如：@张三 则张三可以收到推送，@所有人 时不会收到推送。 • `4`：仅针对 @群全员进行通知，只接收 @所有人 的推送信息 • `5`：不接收通知，即使为 @ 消息也不推送通知。

请求示例

```
POST /ultragroup/notdisturb/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&unpushLevel=2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询默认免打扰配置

更新时间:2024-08-30

获取指定的超级群，或指定频道设置默认的免打扰逻辑。

> 提示
>
> 该设置一般由管理员配置或修改。设置接口详见设置免打扰。

请求方法

POST： https://数据中心域名/ultragroup/notdisturb/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	否	频道 ID，不传时为群的默认免打扰设置

请求示例

```
POST /ultragroup/notdisturb/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
groupId	String	超级群 ID
busChannel	String	频道 ID，不传时为群的默认免打扰设置
unpushLevel	Int	• `-1`：全部消息通知 • `0`：未设置（用户未设置时为此状态，为全部消息都通知，在此状态下，如设置了超级群默认状态以超级群的默认设置为准） • `1`：仅针对 @ 消息进行通知，包括 @指定用户 和 @所有人 • `2`：仅针对 @ 指定用户消息进行通知，且仅通知被 @ 的指定的用户进行通知。 如：@张三 则张三可以收到推送，@所有人 时不会收到推送。 • `4`：仅针对 @群全员进行通知，只接收 @所有人 的推送信息 • `5`：不接收通知，即使为 @ 消息也不推送通知。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"groupId": "abcdefg",
"busChannel": "",
"unpushLevel": 2
}
```

私有频道概述

更新时间:2024-08-30

超级群服务支持创建私有频道，满足社区场景中只有指定用户可以在频道中沟通的业务需求。

已知限制

> 提示
>
> **私有频道功能目前有以下限制：**
>
> **如果用户非私有频道成员，消息发送失败后会进入本地数据库。如果用户未在私有频道成员列表中，但仍往频道中发送消息，此时消息无法成功发送到服务端，但会进入本地数据库（不适用于 Web 端），并可生成会话。**
>
> **建议解决方案：开发者在 App 业务侧维护一份私有频道成员列表数据，如果用户未在私有频道成员列表中，禁止用户往私有频道发送消息。**

了解私有频道

超级群的频道按类型区分为公有频道与私有频道。公有频道对所有超级群成员开放（无需加入）。该超级群的所有成员都会接收公有频道下的消息。私有频道仅对该频道的成员列表开放，仅频道成员可在该频道中收发消息、接收频道状态变化的通知。私有频道可以随时切换为公有频道，公有频道也可以切换为私有频道。

例如，管理员在社区中新创建一个频道，并定义为私有频道。接着邀请社区中部分成员加入私有频道。只有加入频道的成员才可以浏览此频道，并在频道中接收、发送消息。

频道变更会影响该频道下服务端历史消息的拉取权限，具体如下：

- 公有频道转换为私有频道：仅当前私有频道成员列表中的用户可以拉取该频道下的历史消息（含原公有频道消息）；
- 私有频道转换为公有频道：所有超级群用户均可拉取该频道下的历史消息（含原私有频道消息）。

您可以通过以下方式管理私有频道：

功能描述	客户端 API	即时通讯服务端 API
创建私有频道	不提供该 API	创建频道
删除私有频道	不提供该 API	删除频道
查询频道列表	不提供该 API	查询频道列表
变更频道类型	不提供该 API	变更频道类型

了解私有频道成员列表

频道类型为私有频道时，只有该列表中用户可在频道中的收发消息、接收频道内状态通知。

私有频道转为公有频道时，频道变更为对所有超级群成员开放。但该列表数据不会被删除。假设频道类型再次变更为私有，则启用该成员列表。

您也可以为公有频道创建私有频道成员列表。一旦该公有频道变更类型为私有频道，该列表即生效。

> 提示
>
> - 删除频道时，服务端会清除该频道的私有频道成员列表。
> - 一旦超级群成员退群，服务端会自动将用户从私有频道成员列表移除。

您可以通过以下方式管理私有频道成员列表：

功能描述	客户端 API	即时通讯服务端 API
加入私有频道成员列表	不提供该 API	添加私有频道成员
移出私有频道成员列表	不提供该 API	删除私有频道成员
查询私有频道成员列表	不提供该 API	查询私有频道成员列表

变更频道类型

更新时间:2024-08-30

变更超级群频道的类型。频道可以随时切换为公有频道或私有频道。

私有频道转为公有频道：频道变更为对所有超级群成员开放。客户端与服务端不再使用私有频道用户列表，但该列表数据不会被删除。假设频道类型再次变更为私有，则启用该成员列表。
公有频道转为私有频道：频道变更为仅对该频道的私有频道成员列表中的用户开放。

频道变更会影响该频道下历史消息的获取权限，详细请参见私有频道概述

方法说明

POST： https://数据中心域名/ultragroup/channel/type/change.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道 ID
type	Int	是	频道类型，0:公有频道，1:私有频道

请求示例

```
POST /ultragroup/channel/type/change.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
Content-Length: 40

groupId=abc&busChannel=channel001&type=0
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200
}
```

添加私有频道成员

更新时间:2024-08-30

添加用户到指定超级群频道的私有频道成员列表。

被添加的用户 ID 必须已经是超级群成员。如果添加多个用户 ID，请确保全部为超级群成员，否则会全部添加失败。

任意频道（公有或私有）均可创建私有频道成员列表。超级群业务会在频道被设置为私有频道时使用该成员列表。一旦频道变更为私有频道，仅有在私有频道成员列表的成员可在频道内收发消息。

方法说明

POST： https://数据中心域名/ultragroup/channel/private/users/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道 ID
userIds	String	是	群内的用户 ID。单次不超过 20 个。多个用户间用英文 ',' 分割

请求示例

```
POST /ultragroup/channel/private/users/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
Content-Length: 40

groupId=abc&busChannel=channel001&userIds=user1,userId2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。如果返回错误码 24406，请检查被添加的用户 ID 是否均为超级群内成员。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200
}
```

删除私有频道成员

更新时间:2024-08-30

在指定超级群频道的私有频道成员列表移除用户 ID。

被移除的用户 ID 必须已经是超级群成员。如果移除多个用户 ID，请确保全部为超级群成员，否则会全部移除失败。

> 提示
>
> 超级群成员退群时，即时通讯服务端会自动将用户从私有频道成员列表移除。

方法说明

POST： https://数据中心域名/ultragroup/channel/private/users/del.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道 ID
userIds	String	是	群内的用户 ID。单次不超过 20 个。多个用户间用英文 ',' 分割

请求示例

```
POST /ultragroup/channel/private/users/del.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
Content-Length: 40

groupId=abc&busChannel=channel001&userIds=user1,userId2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200
}
```

查询私有频道成员列表

更新时间:2024-08-30

查询在指定超级群频道的私有频道成员列表，返回用户 ID 列表。

方法说明

POST： https://数据中心域名/ultragroup/channel/private/users/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道 ID
page	Int	否	查询页码，默认值 1
pageSize	String	否	查询每页条数，可选值 1-100。默认 50。

请求示例

```
POST /ultragroup/channel/private/users/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
Content-Length: 40

groupId=abc&busChannel=channel001&page=1&pageSize=50
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常
users	JSONArray	用户 ID 列表

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"users": [
"qUser2"
]
}
```

查询用户所属的私有频道

更新时间:2024-08-30

分页查询用户所属的私有频道成员列表，返回超级群频道 ID。

> 提示
>
> **返回的超级群频道 ID 可能并非全部都是私有频道。超级群公有频道、私有频道均可创建私有频道成员列表。在频道类型为私有频道时，只有在私有频道成员列表上的用户可参与该频道的活动。如果需要查询查询指定用户所属的私有频道列表，可能需要在结果中剔除公有频道。**

请求方法

POST： https://数据中心域名/ultragroup/user/channel/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID。
userId	String	是	用户 ID。
page	int	否	查询页码，默认 1。
pageSize	int	否	每页条数，默认 10，最多为 50。

请求示例

```
POST /ultragroup/user/channel/query.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&userId=userId&page=1&pageSize=50
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功。
data	String[]	超级群频道 ID 列表。

返回结果示例

```
{
"code":200,
"data":[
"busChannel1",
"busChannel2"
]
}
```

用户组概述

更新时间:2024-08-30

超级群用户组（User Group）功能是超级群业务提供的群成员管理工具，结合超级群私有频道功能，可以帮助 App 实现更高效的超级群成员管理，沟通管理，和更精细的用户通知能力。

前提条件

超级群用户组功能主要通过与私有频道的绑定操作，提供了对用户在社区私有频道中沟通权限（收发消息、通知）的批量管理能力，提升了 App 集成效率与对超级群的运营管理能力。

在了解与使用超级群用户组功能前，需要先了解超级群私有频道功能。请参见超级群私有频道概述。

如何使用超级群用户组

App 后端可以在超级群中创建最多 50 个用户组（userGroup），每个用户组成员最多由 100 个超级群成员组成。单个用户可以存在于多个用户组中。

用户组创建后，可以与超级群频道绑定。如果用户组绑定了一个或多个私有频道，该用户组的所有成员即具有在绑定的私有频道中收发消息、接收通知的能力。

- App 将用户组绑定私有频道后，可认为组中所有用户均加入了该私有频道。与该私有频道成员列表中的用户类似，只有加入频道的成员才可以浏览此频道，并在频道中接收消息，发送消息，和接收通知。
- App 将用户组绑定公有频道后，不会影响组中用户可收发消息的范围。但一旦该公有频道转为私有频道，该组用户将具有在该私有频道中收发消息、获取通知的能力。
- App 可以在一个频道上绑定最多 10 个用户组，一个用户组可以与多个频道绑定（一个超级群最多 50 个频道）。

> 提示
>
> 超级群业务默认提供 RCDefault 频道，对所有超级群成员开放，不可转为私有频道。建议不要将用户组绑定到 RCDefault 频道。

混合使用用户组与私有频道成员列表

只要 App 用户在指定私有频道绑定的任意一个用户组中，或者在有该私有频道成员列表中，该用户就能在私有频道中收发消息接收通知。

如果混合使用私有频道成员列表与用户组，在 App 业务中可能存在以下情况：

- 私有频道配置了私有成员列表，并添加了多位用户。
- 该私有频道绑定了多个用户组。

在上述使用场景中，某个用户可能既在私有频道成员列表中，又同时在该频道绑定的多个用户组中。如果该用户不再使用该私有频道，App 进行以下操作，确保该用户无法继续在私有频道收发消息：

- 从该私有频道的成员列表中移除该用户。
- 检查该私有频道绑定的所有用户组，从绑定的所有用户组中移除该用户，或解绑用户组。

超级群用户组管理接口

超级群用户组的业务逻辑全部需要在 App 服务器进行实现。建议 App 后端同时维护一份用户、用户组、频道之间对应关系的数据。

对于客户端开发人员来说，创建用户组等基础管理操作只需要与 App 后端交互即可。App 后端需要调用相应的即时通讯服务端 API（Server API）接口相关接口创建用户组等其他管理操作。

下表列出了超级群用户组基础管理接口。其他相关接口可参见 API 接口列表。

功能分类	功能描述	即时通讯服务端 API
创建、删除用户组	在指定超级群下创建用户组。如删除用户组，则用户、用户组、频道之间的关系不复存在。	创建用户组、删除用户组
查询用户组列表	分页查询指定超级群下的用户组，返回用户组 ID 列表。	查询用户组列表
添加、删除用户	在超级群用户组中添加、删除用户。	添加用户、移出用户
查询用户所属用户组	分页查询指定单个用户在超级群下所属的用户组列表，返回用户组 ID 列表。	查询用户所属用户组
绑定频道与用户组	将指定单个超级群频道与用户组绑定，可绑定多个用户组，单个频道最多支持与 10 个用户组绑定。	绑定频道与用户组
解绑频道与用户组	将指定单个超级群频道与用户组解除绑定，单次请求可解绑最多 10 个用户组。	解绑频道与用户组
查询频道绑定的用户组	分页查询指定的超级群频道绑定的用户组列表，返回用户组 ID 列表。	查询频道绑定的用户组
查询用户组绑定的频道	分页查询指定单个用户组绑定的超级群频道列表，返回超级群频道 ID 列表。	查询用户组绑定的频道

创建用户组

更新时间:2024-08-30

创建用户组。每个超级群最多创建 50 个用户组。

请求方法

POST： https://数据中心域名/ultragroup/usergroup/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/json，包含具有以下结构的 JSON 对象：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID，请确保超级群 ID 存在。
userGroups	Object[]	是	用户组信息列表，最大长度为 10。
userGroups[i].userGroupId	String	是	用户组 ID，支持大小写字母、数字的组合方式，长度不超过 64 个字符。

请求示例

```
POST /ultragroup/usergroup/add.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/json
Content-Length: 141

{
"groupId":"groupId1",
"userGroups":[
{
"userGroupId":"id1"
},
{
"userGroupId":"id2"
}
]
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功。24418 表示超级群下用户组数量超出限制，24419 表示用户组 ID 重复

返回结果示例

```
{
"code":200
}
```

删除用户组

更新时间:2024-08-30

删除用户组。

- 删除用户组时，服务端会自动删除用户与用户组的成员关系。
- 删除用户组时，服务端会自动删除用户组与超级群频道的绑定关系。
- 被删除的用户组中的原有成员可通过客户端回调收到通知。

请求方法

POST： https://数据中心域名/ultragroup/usergroup/del.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID，请确保超级群 ID 存在。
userGroupIds	String	是	用户组 ID 列表，多个 ID 以逗号分隔。单次请求最大长度为 10 个，否则全部失败。

请求示例

```
POST /ultragroup/usergroup/del.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&userGroupIds=roleId1,roleId2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功

返回结果示例

```
{
"code":200
}
```

查询用户组列表

更新时间:2024-08-30

分页查询指定超级群下的用户组，返回用户组 ID 列表。

请求方法

POST： https://数据中心域名/ultragroup/usergroup/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID。
page	int	否	查询页码，默认 1。
pageSize	int	否	每页条数，默认 10，最多为 50。

请求示例

```
POST /ultragroup/usergroup/query.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&page=1&pageSize=10
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功
userGroups	Object[]	用户组列表
userGroups[].userGroupId	String	用户组 ID

返回结果示例


```
{
"code":200,
"userGroups":[
{
"userGroupId":"roleId1"
},
{
"userGroupId":"roleId2"
}
]
}
```

添加用户

更新时间:2024-08-30

在超级群用户组中添加用户。单个用户组中成员数量上限 100个。

- 新加入用户组的用户可通过客户端回调收到通知。

请求方法

POST： https://数据中心域名/ultragroup/usergroup/user/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID，请确保超级群 ID 存在。
userGroupId	String	是	用户组 ID，请确保用户组 ID 存在。
userIds	String	是	群内用户 ID 列表，多个 ID 以逗号分隔。单次不得超过 20人，请确保用户 ID 存在，否则全部失败。

请求示例

```
POST /ultragroup/usergroup/user/add.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&userGroupId=id&userIds=user1,user2,user3
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功。24419 表示用户组下用户数量超出限制。

移出用户

更新时间:2024-08-30

将用户移出指定用户组。

- 被移出用户组的用户可通过客户端回调收到通知。

请求方法

POST： https://数据中心域名/ultragroup/usergroup/user/del.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID
userGroupId	String	是	用户组 ID
userIds	String	是	群内用户 ID 列表，多个 ID 以逗号分隔。单次不得超过 20 人，否则全部失败。

请求示例

```
POST /ultragroup/usergroup/user/del.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&userGroupId=userGroupId1&userIds=user1,user2,user3
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功

查询用户所属用户组

更新时间:2024-08-30

分页查询指定单个用户在超级群下所属的用户组列表，返回用户组 ID 列表。

请求方法

POST： https://数据中心域名/ultragroup/user/usergroup/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID。
userId	String	是	用户 ID。
page	int	否	查询页码，默认 1。
pageSize	int	否	每页条数，默认 10，最多为 50。

请求示例

```
POST /ultragroup/user/usergroup/query.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&userId=userId1&page=2&pageSize=20
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功
data	String[]	用户组 ID 列表

返回结果示例

```
{
"code":200,
"data":[
"userGroupId1",
"userGroupId2"
]
}
```

绑定频道与用户组

更新时间:2024-08-30

将指定单个超级群频道与用户组绑定。单次请求可绑定多个用户组，单个频道最多支持与 10 个用户组绑定。

- 绑定成功的用户组中的所有用户可通过客户端回调收到通知。

请求方法

POST： https://数据中心域名/ultragroup/channel/usergroup/bind.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道 ID
userGroupIds	String	是	用户组 ID 列表，多个 ID 以逗号分隔。单次请求不能超过 10 个，否则全部失败。

请求示例

```
POST /ultragroup/channel/usergroup/bind.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&busChannel=channelid&userGroupIds=id1,id2,id3
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功。24420 表示频道绑定用户组数量超出限制。

返回结果示例

```
{
"code":200
}
```

解绑频道与用户组

更新时间:2024-08-30

将指定单个超级群频道与用户组解除绑定。单次请求可解绑最多 10 个用户组。

- 成功解除绑定的用户组中的所有用户可通过客户端回调收到通知。

请求方法

POST： https://数据中心域名/ultragroup/channel/usergroup/unbind.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道ID
userGroupIds	String	是	用户组 ID 列表，多个 ID 以逗号分隔。单次不能超过 10 个，否组全部失败。

请求示例

```
POST /ultragroup/channel/usergroup/unbind.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&busChannel=channelid&userGroupIds=id1,id2,id3
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功

返回结果示例

```
{
"code":200
}
```

查询频道绑定的用户组

更新时间:2024-08-30

分页查询指定的超级群频道绑定的用户组列表，返回用户组 ID 列表。

请求方法

POST： https://数据中心域名/ultragroup/channel/usergroup/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID
busChannel	String	是	频道ID
page	int	否	查询页码，默认1
pageSize	int	否	每页条数，默认10，最多为50

请求示例

```
POST /ultragroup/channel/usergroup/query.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&busChannel=buschannel1&page=2&pageSize=20
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功。
data	String[]	用户组 ID 列表。

返回结果示例

```
{
"code":200,
"data":[
"userGroupId1",
"userGroupId2"
]
}
```

查询用户组绑定的频道

更新时间:2024-08-30

分页查询指定单个用户组绑定的超级群频道列表，返回超级群频道 ID 列表。

请求方法

POST： https://数据中心域名/ultragroup/usergroup/channel/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	是否必传	说明
groupId	String	是	超级群 ID。
userGroupId	String	是	用户组 ID。
page	int	否	查询页码，默认1。
pageSize	int	否	每页条数，默认 10，最多为 50。

请求示例

```
POST /ultragroup/usergroup/channel/query.json HTTP/1.1
Host: api.rong-api.com
App-key: kj8swf7oksq89
Nonce: 1181222303
Signature: 6221193e0ffebc01366da6e4cf8950cf32d274d6
Timestamp: 1668430324
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

groupId=abc&userGroupId=userGroupId&page=2&pageSize=20
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

参数	类型	说明
code	int	返回码 200 表示成功。
data	String[]	超级群频道 ID 列表。

返回结果示例


```
{
"code":200,
"data":[
"busChannel1",
"busChannel2"
]
}
```

超级群禁言服务概述

更新时间:2024-08-30

即时通讯服务提供超级群禁言服务，具体提供以下功能：

- 超级群单人禁言：单人禁言服务支持在超级群（或频道）范围内禁言指定用户，或在指定用户加入的所有超级群中将其禁言。
- 超级群全体成员禁言：设置指定的超级群（或频道）为全体禁言状态。
- 超级群全体禁言白名单：在对 App 下的超级群或指定频道设置全体禁言后，可能还需要允许部分用户往群组或频道中发送消息。较简单的实现方式是设置对应级别的白名单。

禁言效果说明

禁言服务仅影响从客户端发送消息的行为。

- 如果用户被单人禁言，则无法通过客户端 SDK 往该群组中发消息。
- 如果用户所在群组、频道被全体禁言，则不可通过客户端 SDK 往该超级群（或频道）中发消息。
- 在设置禁言服务后，如需允许部分用户往群组或频道中发送消息，可以使用超级群全体禁言白名单服务。请注意，较简单的实现方式是设置对应级别（群或频道）的白名单。如果混用超级群、超级群频道级别的全体禁言与白名单，请参见下方说明。

> 提示
>
> 服务端（Server API）发送群聊消息接口不受群组全体成员禁言状态的限制，被禁言用户可通过 Server API 往该群组中发送消息。

混用不同级别的禁言与禁言白名单服务

超级群业务提供超级群级别和频道级别的全体禁言白名单服务。如果您的 App 业务同时使用了超级群单人禁言功能（在群范围或频道范围禁言指定成员），具体行为如下：

如果单人禁言范围与白名单生效的范围一致（同为群或频道）时，白名单优先生效。参见以下场景描述：

- **情况 1**：设置用户在超级群中单人禁言，如用户在全体禁言白名单中，可正常发送消息。
- **情况 2**：设置用户在超级群频道中单人禁言，如用户在频道的全体禁言白名单中，可正常发送消息。

如果单人禁言的范围与白名单生效的范围不一致时，超级群级别的禁言或白名单设置具有更高优先级：

- **情况 3**：设置用户在超级群中禁言，如用户仅在频道的白名单中，则不可发送消息。
- **情况 4**：设置用户在超级群频道中禁言，如用户已在超级群的全体禁言白名单中，则仍可正常发送消息。

关于超级群禁言服务行为变更的说明

> 提示
>
> 本节内容仅适用于在 2022.10.13 日前已开通超级群服务的客户。

如果您的应用/环境在 2022.10.13 日前已开通超级群服务，超级群级别的单人禁言、全体禁言、全体禁言白名单设置仅可控制发送不属于任何频道（指发消息时不携带频道 ID）的超级群消息的行为。

例如，该客户使用**超级群全体成员禁言白名单**接口将 App 用户加入超级群（groupId）级别的白名单后，白名单用户可发送不属于任何频道的消息（发送消息时未指定频道 ID），仍禁止往任何群频道中发送消息。如需要允许用户往群及任何频道中发送消息，请遍历所有超级群下所有频道，并将该用户加入频道的白名单。

即时通讯服务在 2022.10.13 对超级群禁言服务的行为进行了调整，实现一次调用即可对指定超级群下所有频道进行禁言、白名单设置。如果您的应用/环境在 2022.10.13 日及以后开通超级群服务，超级群业务中会包含一个 ID 为 RCDefault 的默认频道，且群级别的禁言相关服务设置会自动适用于所有频道。

即时通讯服务支持客户调整服务至最新行为。该行为调整将影响客户端、服务端收发消息、获取会话、清除历史消息、禁言等多个功能。如有需要，请提交工单咨询详细方案。

禁言指定超级群成员

更新时间:2024-08-30

在指定超级群（或指定频道）或全部频道中，禁言一个或多个用户。

- 设置用户在超级群（`groupId`）禁言后，该成员不能通过客户端 SDK 往该超级群及频道中发送消息。设置用户在超级群频道（`busChannel`）禁言后，该成员不能通过客户端 SDK 往该超级群频道中发送消息。
- 服务端（Server API）发送超级群消息接口不受禁言状态的限制，被禁言用户可通过 Server API 往该超级群中发送消息。
- 用户退出超级群，禁言数据不会清除。被设置为单人禁言的用户即使退出超级群，再次加入时，禁言状态仍然有效。
- 如果需要将用户在所有频道中都设置为禁言，可不指定频道 ID（`busChannel` 传空）。请注意，单个 App Key 下最多支持将 100 人设置为在超级群全部频道中禁言。如需查询是否已达上限，您可以在调用查询超级群成员禁言列表接口（`/ultragroup/userbanned/get.json`）时将 `busChannel` 传空。

> 提示
>
> 如果您的 App 业务需要同时使用超级群的单人禁言、全体禁言、全体禁言白名单功能（在群级别或频道级别禁言指定成员，以及设置群或频道的禁言白名单），可参考加入超级群禁言白名单中的同时使用超级群单人禁言服务。

如 API 行为不符合预期，请参见关于超级群禁言服务行为变更的说明。

请求方法

POST： https://数据中心域名/ultragroup/userbanned/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

请求示例

参数	类型	必传	说明
groupId	String	是	超级群 ID。
busChannel	String	否	频道 ID。
userIds	String	是	用户 ID 列表，每次最多不超过 20 个用户，以逗号分隔。

请求示例

```
POST /ultragroup/userbanned/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&userIds=userId_001,userId_002
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消指定超级群成员禁言

更新时间:2024-08-30

在指定超级群（或频道）中，解除一个或多个用户的禁言状态。

如果用户被设置为在超级群全部频道中禁言，可通过将 busChannel 传空为用户解除该状态。

请求方法

POST： https://数据中心域名/ultragroup/userbanned/del.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID。
busChannel	String	否	频道 ID。
userIds	String	是	用户 ID 列表，每次最多不超过 20 个用户，以逗号分隔。

请求示例

```
POST /ultragroup/userbanned/del.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&userIds=userId_001,userId_002
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询超级群成员禁言列表

更新时间:2024-08-30

获取在指定超级群（或频道）中被禁言的用户，返回用户 ID 列表。

不指定频道 ID 时（busChannel 为空）则获取所有群组禁言用户列表。

如 API 行为不符合预期，请参见关于超级群禁言服务行为变更的说明。

请求方法

POST： https://数据中心域名/ultragroup/userbanned/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	否	频道 ID。
page	String	否	当前页码，默认获取第一页。
pageSize	String	否	每页条数，默认每页 50 条。

请求示例

```
POST /ultragroup/userbanned/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	Array of objects	禁言成员列表。
users[i].userId	String	群成员 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"users": [
{
"id": "userId_001"
}
]
}
```

设置超级群全体禁言

更新时间:2024-08-30

设置指定的超级群（或频道）的禁言状态，支持设为禁言或取消禁言。

- 设置超级群全体禁言后，该超级群的所有成员均不能往该超级群及频道中发送消息。
- 设置超级群频道全体禁言后，该超级群的所有成员均不能往该超级群频道中发送消息。

> 提示
>
> - 超级群（或频道）解散再创建，禁言状态仍然有效。
> - 服务端（Server API）发送超级群消息接口不受全体禁言状态的限制，被禁言用户可通过 Server API 往该超级群中发送消息。

如需添加例外用户，请使用禁言白名单。详见加入超级群全体成员禁言白名单。

如果 API 行为不符合预期，请参考关于超级群禁言服务行为变更的说明。

请求方法

POST： https://数据中心域名/ultragroup/globalbanned/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	否	频道 ID。
status	String	是	`true` 为已禁言状态。`false` 为未被禁言状态。

请求示例

```
POST /ultragroup/globalbanned/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&status=true
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询超级群全体禁言

更新时间:2024-08-30

查询指定的超级群（或频道）的禁言状态。

请求方法

POST： https://数据中心域名/ultragroup/globalbanned/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID
busChannel	String	否	频道 ID 字段。

请求示例

```
POST /ultragroup/globalbanned/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
status	boolean	`true` 为已禁言状态。`false` 为未被禁言状态。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"status": true
}
```

加入超级群全体禁言白名单

更新时间:2024-08-30

App 可以添加一个或多个超级群成员到指定超级群（或频道）的禁言白名单。

在对 App 下的超级群或指定频道设置全体禁言后，可能需要允许部分用户通过客户端往群组或频道中发送消息。较简单的实现方式是设置对应级别的白名单，如下：

- 设置超级群全体禁言后，如希望允许用户往群组中发送消息，可将该用户加入超级群的白名单。
- 设置超级群频道全体禁言后，如希望允许用户往该频道发送消息，可将该用户加入该频道的白名单。

> 提示
> - 在白名单的用户如果退出超级群，再次加入，白名单状态仍然有效。
> - 超级群解散再创建，白名单状态仍然有效。

请求方法

POST： https://数据中心域名/ultragroup/banned/whitelist/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID。
busChannel	String	否	频道 ID 字段。
userIds	String	是	用户 ID 列表（以逗号分隔的字符串）。单次请求中最多不超过 20 个用户 ID。

请求示例

```
POST /ultragroup/banned/whitelist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb8j
NONCE: 97
TIMESTAMP: 1480479442
SIGNATURE: 9774e3d91656dc92df8aff294d46d6a506412538
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&userIds=userId_001,userId_002
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

同时使用超级群单人禁言服务

如果您的 App 业务同时使用了超级群的单人禁言功能（在群范围或频道范围禁言指定成员），注意在单人禁言范围与白名单生效的范围一致（同为群或频道）时，白名单优先生效。参见以下场景描述：

- **情况 1**：设置用户在超级群中单人禁言，如用户在全体禁言白名单中，可正常发送消息。
- **情况 2**：设置用户在超级群频道中单人禁言，如用户在频道的全体禁言白名单中，可正常发送消息。

单人禁言的范围与白名单生效的范围不一致时，超级群级别的禁言或白名单设置具有更高优先级：

- **情况 3**：设置用户在超级群中禁言，如用户仅在频道的白名单中，则不可发送消息。
- **情况 4**：设置用户在超级群频道中禁言，如用户已在超级群的全体禁言白名单中，则仍可正常发送消息。

移出超级群全体禁言白名单

更新时间:2024-08-30

将一个或多个超级群成员从指定超级群（或频道）的禁言白名单中移出。

请求方法

POST： https://数据中心域名/ultragroup/banned/whitelist/del.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID。
busChannel	String	否	频道 ID 字段。
userIds	String	是	用户 ID 列表（以逗号分隔的字符串）。单次请求中最多不超过 20 个用户 ID。

请求示例

```
POST /ultragroup/banned/whitelist/del.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg&userIds=userId_001,userId_002
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询群组全体禁言白名单

更新时间:2024-08-30

获取指定超级群（或频道）的禁言白名单。

请求方法

POST： https://数据中心域名/ultragroup/banned/whitelist/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
groupId	String	是	超级群 ID。
busChannel	String	否	频道 ID。
page	String	否	当前页码,默认1。
pageSize	String	否	每页条数，默认50。 上限 200

请求示例

```
POST /ultragroup/banned/whitelist/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=abcdefg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	String[]	用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"users": [
{
"id": "userId_001"
},
{
"id": "userId_002"
}
]
}
```

关于超级群禁言服务行为变更的说明

如果您的应用/环境在 2022.10.13 日前已开通超级群服务，群级别的单人禁言、全体禁言、全体禁言白名单设置仅可控制发送不属于任何频道（指发消息时不携带频道 ID）的超级群消息的行为。

即时通讯服务在 2022.10.13 对超级群禁言服务的行为进行了调整，实现一次调用即可对指定超级群下所有频道进行禁言、白名单设置。如果您的应用/环境在 2022.10.13 日及以后开通超级群服务，超级群业务中会包含一个 ID 为 `RCDefault` 的默认频道，且群级别的禁言相关服务设置会自动适用于所有频道。

即时通讯服务支持客户调整服务至最新行为。该行为调整将影响客户端、服务端收发消息、获取会话、清除历史消息、禁言等多个功能。如有需要，请提交工单咨询详细方案。

聊天室概述

更新时间:2024-08-30

聊天室（Chatroom）提供了支持高并发消息处理的业务形态，可用于直播、社区、游戏、广场交友、兴趣讨论等场景。App Key 下可创建的聊天室数量没有限制，单个聊天室成员数量没有限制。

服务配置

聊天室不需要申请开通。聊天室的部分基础功能与增值服务可以在控制台的免费基础功能和 IM 服务管理页面进行开通和配置。

聊天室自动销毁机制

聊天室具有自动销毁机制，默认情况下所有聊天室会在不活跃（连续时间段内无成员进出且无新消息）达到 1 小时后踢出所有成员并自动销毁，可修改时长，也可配置为定时自动销毁。

聊天室业务支持灵活控制每个聊天室的存活条件与时长。详见聊天室自动销毁机制。

聊天室离线成员自动退出机制

聊天室具有离线成员自动退出机制。用户离线后，如满足以下默认预设条件，即时通讯服务端会自动将该用户踢出聊天室：

- 从用户离线开始 30 秒内，聊天室中产生第 31 条消息时，触发自动踢出。
- 或用户已离线 30 秒后，聊天室有新消息产生时，触发自动踢出。

> 提示
>
> - 默认预设条件均要求聊天室中必须要有新消息产生，否则无法触发踢出动作。如果聊天室中没有消息产生，则无法将异常用户踢出聊天室。
> - 如需修改默认行为对新消息的依赖，请提交工单申请开通聊天室成员异常掉线实时踢出。开通该服务后，服务端会通过 SDK 行为（要求 Android/iOS IMLib SDK 版本 ≧ 5.1.6，Web IMLib 版本 ≧ 5.3.2）判断用户是否处于异常状态，最迟 5 分钟可以将异常用户踢出聊天室。
> - 如需保护特定用户，即不自动踢出指定用户（如某些应用场景下可能希望用户驻留聊天室），可使用 Server API 提供的聊天室用户白名单功能。

聊天室消息能力

- 聊天室不具备离线消息转推送功能，只有在线的聊天室成员可接收聊天室消息。
- 聊天室本地消息会在退出聊天室时删除。**IM 旗舰版** 与 **IM 尊享版**客户可选择启用聊天室消息云端存储功能。具体功能与费用以官方价格说明页面及计费说明文档为准。

聊天室管理接口

聊天室会话关系由即时通讯服务端负责建立并保持连接。SDK 提供加入、退出等部分聊天室管理接口。更多聊天室管理功能需要配合使用即时通讯服务端 API。下表描述了聊天室主要的功能接口。

功能分类	功能描述	即时通讯服务端 API
创建聊天室	手动创建聊天室。创建聊天室时可指定自动销毁类型（定时销毁或不活跃时销毁），可配置是否禁言全体成员，可设置聊天室自定义属性（KV）。	创建聊天室
销毁聊天室	手动销毁聊天室。	销毁聊天室
加入聊天室	加入聊天室。	不提供该 API
退出聊天室	在线用户可主动退出聊天室。离线用户可被自动踢出聊天室，不需要额外处理。	不提供该 API
查询聊天室房间与用户信息	查询聊天室房间的基础信息，包括聊天室 ID、创建时间、人数、自动销毁类型等。	查询房间信息
聊天室保活	添加一个或多个聊天室到聊天室保活列表。在保活列表中的聊天室不会被即时通讯服务端自动销毁。	保活聊天室
聊天室属性管理	在指定聊天室中设置自定义属性。比如在语音直播聊天室场景中，利用此功能记录聊天室中各麦位的属性；或在狼人杀等卡牌类游戏场景中记录用户的角色和牌局状态等。 聊天室属性以 Key-Value 的方式进行存储，支持设置、删除与查询属性，支持批量和强制操作。	属性管理（KV）
封禁/解封聊天室用户	封禁一个或多个聊天室成员。被封禁成员将被踢出指定聊天室，并在封禁时间内不能再进入此聊天室中。	成员封禁
聊天室用户白名单	需在 IM 服务管理 页面普通服务下开通后使用。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。 用户被加入某个聊天室的白名单后，在该聊天室消息量较大的情况下，该用户发送的消息不会被丢弃；并且用户也不会被即时通讯服务端自动踢出该聊天室。	聊天室白名单服务
发送聊天室消息	发送聊天室消息。	发送聊天室消息
撤回聊天室消息	撤回聊天室消息。	消息撤回
获取聊天室历史消息	获取聊天室历史消息。	历史消息日志
聊天室低级别消息	需在 IM 服务管理 页面普通服务下开通后使用。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。 如果消息类型在低级别消息列表中，该类型的消息全部视为低级别消息。当服务器负载高时，高级别的消息优先保留，低级别消息则优先丢弃。默认情况下，所有消息均为高级别消息。	聊天室消息优先级服务
聊天室消息白名单	需在 IM 服务管理 页面普通服务下开通后使用。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。 如果消息类型在聊天室消息白名单中，该类型的消息全部受到保护，在聊天室消息量较大的情况下也不会被丢弃。	聊天室白名单服务
聊天室成员禁言	在指定的某个聊天室中，禁言一个或多个成员。聊天室成员被禁言后，可以接收并查看聊天室中用户聊天信息，但不能通过往该聊天室内发送消息。	单人禁言
全体成员禁言	设置某一聊天室全部成员禁言，或取消指定聊天室全部成员禁言状态。设置全体群成员禁言后，该聊天室的所有成员均不能通过客户端 SDK 往该群组内发送消息。	全体禁言
全体禁言白名单	添加一个或多个群成员到聊天室全体成员禁言白名单。聊天室成员被添加到白名单后，即使该聊天室处于全体成员禁言状态，该成员仍可通过客户端 SDK 往该聊天室发送消息。	全体禁言
全局禁言聊天室成员	需在 IM 服务管理 页面普通服务下开通后使用。**IM 旗舰版**或**IM 尊享版**可开通该服务。具体功能与费用以官方价格说明 页面及计费说明 文档为准。 添加一个或多个用户到聊天室全局禁言列表中，列表中的用户在应用下的所有聊天室中都无法发送消息。	全局禁言

聊天室服务配置

更新时间:2024-08-30

聊天室业务本身不需要单独申请开通，但部分聊天室服务需要在控制台开通与配置，例如聊天室广播消息、聊天室消息云端存储、以及与聊天室相关的回调地址等。

聊天室服务配置主要在免费基础功能和 IM 服务管理页面。

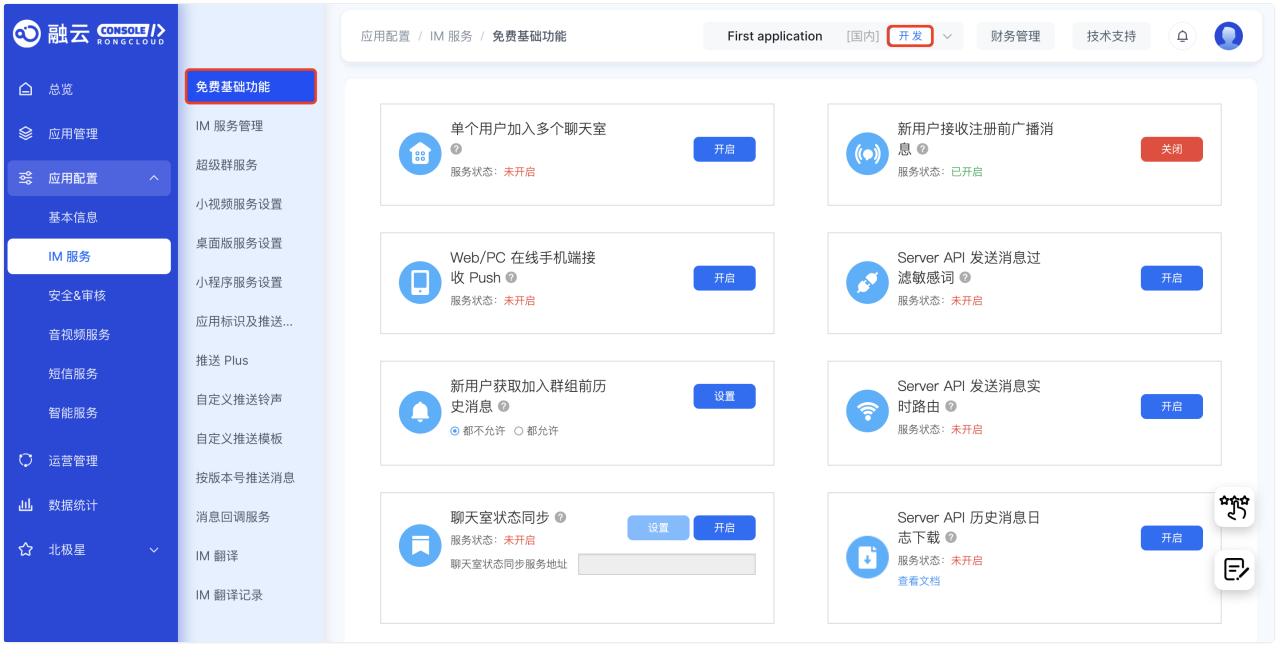
免费基础功能

以下是聊天室业务提供的免费基础功能：

- **单个用户加入多个聊天室**：默认一个用户只能加入一个聊天室中，开启后一个用户可以同时加入到多个聊天室中。
- **聊天室状态同步**：聊天室状态同步是即时通讯服务提供的回调服务，需同时提供可正常访问的回调地址。配置成功后，在应用下聊天室发生状态变化时，将实时同步到开发者的应用服务器地址，目前支持的同步状态包括：创建、销毁、成员加入、成员退出聊天室。详见「聊天室管理」下的聊天室状态同步。
- **加入聊天室获取指定消息设置**：默认加入聊天室时可最多获取全部消息类型的最近 50 条消息，开启后可设置指定消息类型获取。
- **聊天室销毁等待时间**：
  1. 可以支持配置不活跃聊天室销毁的等待时间，默认等待时间为1小时，即超过1小时不活跃即被销毁，客户可以按需调整这个时间（必须为 1-24 之间的整数），最长可设置24小时。
  2. 聊天室销毁时，会向聊天室成员发送聊天室销毁通知，以方便客户可以在聊天室销毁后，在终端自定义一些操作(依赖 5.1.1 及以后版本)。
  3. 聊天室增加 sessionid，在聊天室生存周期内保持不变，聊天室重建后重新生成。用于使用相同聊天室ID，多次开播时，客户端能区分出来。
- **聊天室属性自定义设置**：可在指定聊天室中设置自定义属性，用于语音直播聊天室场景的会场属性同步或狼人杀等卡牌类游戏场景中记录用户的角色和牌局状态等，详见「聊天室管理」下的设置聊天室属性。如果 App 业务服务端需要即时通讯服务提供聊天室属性变更数据同步，需要提供可正常访问的回调地址，配置成功后，自动开启即时通讯服务提供的回调服务，详见「聊天室管理」下的聊天室属性同步。

修改服务配置

访问控制台免费基础功能 页面，可调整聊天室业务相关的免费基础功能配置。



IM 旗舰版/尊享版功能

访问开发后台 IM 服务管理 页面，切换到**普通服务**标签下，可启用以下聊天室服务配置开关。

> 提示
>
> **开发环境下可以免费使用。生产环境下，IM 旗舰版或 IM 尊享版才能使用以下服务。**

- **聊天室广播消息**：向应用中的所有聊天室发送一条消息，单条消息最大 128k。详见「消息管理」下的发送全体聊天室广播消息。
- **聊天室全局禁言功能**：当不想让某一用户在所有聊天室中发言时，可将此用户添加到聊天室全局禁言中，被禁言用户可接收查看聊天室中用户聊天信息，但不能发送消息。详见「聊天室用户管理」下的全局禁言用户。
- **聊天室消息优先级服务**：在指定聊天室中设置指定类型的消息为低级别消息。当服务器负载高时低级别消息优先被丢弃，这样可以确保重要的消息不被丢弃。详见「聊天室消息优先级服务」下的添加低级别消息。
- **聊天室白名单服务**：开通后，可以使用以下功能对应的 Server API：
  - 聊天室用户白名单：可用于保护指定聊天室中的重要用户，支持按聊天室设置白名单用户。例如，App 业务中指定聊天室中的管理员、主播等重要角色的用户。
  - 聊天室消息白名单：可用于保护 App 下所有聊天室中的指定消息类型。例如 App 业务中自定义的红包消息。
- **聊天室保活服务**：当聊天室中 1 小时无人说话，同时没有人加入聊天室时，即时通讯服务端会自动把聊天室内所有成员踢出聊天室并销毁聊天室。保活的聊天室不会被自动销毁，可以调用 API 接口销毁聊天室。详见「聊天室管理」下的保活聊天室。
- **聊天室消息云端存储**：聊天消息保存在云端，用户进入聊天室后，可以查看聊天室中以前的消息，历史消息默认保存 2 个月。

创建聊天室

更新时间:2024-08-30

创建单个聊天室，支持配置销毁类型、是否全体禁言、全体禁言白名单，和聊天室自定义属性。如需设置销毁类型，请确保您已了解聊天室销毁机制。

该 API 替代了废弃接口创建聊天室（废弃）。

请求方法

POST： https://数据中心域名/chatroom/create_new.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。
destroyType	int	否	指定聊天室的销毁类型。`0`：默认值，表示不活跃时销毁。默认情况下，所有聊天室的自动销毁方式均为不活跃时销毁，一旦不活跃长达到 60 分钟即被销毁，可通过 `destroyTime` 延长该时间。`1`：固定时间销毁，设置为该类型后，聊天室默认在创建 60 分钟后自动销毁，可通过 `destroyTime` 设置更长的存活时间。您也可以在聊天室创建成功后再设置，详见设置聊天室销毁类型。
destroyTime	int	否	设置聊天室销毁时间。在 `destroyType=0` 时，表示聊天室应在不活跃达到该时长时自动销毁。在 `destroyType=1` 时，表示聊天室应在创建以后存活时间达到该时长后自动销毁。单位为分钟，最小值 60 分钟，最大 10080 分钟（7 天）。如果未设置，默认 60 分钟。
isBan	boolean	否	是否禁言聊天室全体成员，默认 `false`。您也可以在聊天室创建成功后再设置，详见设置聊天室全体禁言。
whiteUserIds	array	否	禁言白名单用户列表，支持批量设置，最多不超过 20 个。您也可以在聊天室创建成功后再设置，详见加入聊天室全体禁言白名单。
entryOwnerId	String	否	聊天室自定义属性的所属用户 ID。仅在开通聊天室自定义属性服务后可使用该字段，且必须与 `entryInfo` 字段一起使用。如果未开启服务，或者设置该字段时未同时传入 `entryInfo`，API 会返回创建失败。仅支持 1 个用户 ID。您也可以在聊天室创建成功后再设置，详见聊天室属性概述。
entryInfo	String	否	聊天室自定义属性 KV 对，JSON 结构。仅在开通聊天室自定义属性服务后可使用该字段，必须与 `entryOwnerId` 字段一起使用。支持批量设置 KV 对，一次最多 20 个。Key 为属性名，支持大小写英文字母、数字、部分特殊符号 + = - _ 的组合方式，大小写敏感。最大长度 128 字符。Value 为属性值，最大长度 4096 个字符。您也可以在聊天室创建成功后再设置，详见聊天室属性概述。

请求示例

```
POST /chatroom/create_new.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

chatroomId=gid1&destroyType=1&destroyTime=120
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码。 200 表示处理成功。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200}
```

查询聊天室信息

更新时间:2024-08-30

获取聊天室基础信息，包括聊天室 ID、创建时间、人数、自动销毁类型、是否全体禁言等。

该接口替代了废弃接口获取聊天室基础信息(废弃)。

请求方法

POST： https://数据中心域名/chatroom/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID

请求示例

```
POST /chatroom/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

chatroomId=gid1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码。**200** ：处理成功。**23410**:聊天室不存在
chatroomId	String	聊天室 ID
createTime	long	聊天室创建时间

返回值	返回类型	说明
memberCount	int	聊天室当前人数
destroyType	int	指定聊天室的销毁方式。0：默认值，表示不活跃时销毁。默认情况下，所有聊天室的自动销毁方式均为不活跃时销毁，一旦不活跃长达到 60 分钟即被销毁，可通过 destroyTime 延长该时间。1：固定时间销毁，设置为该类型后，聊天室默认在创建 60 分钟后自动销毁，可通过 destroyTime 设置更长的存活时间。
destroyTime	int	设置聊天室销毁等待时间。在 destroyType=0 时，表示聊天室应在不活跃达到该时长时自动销毁。在 destroyType=1 时，表示聊天室应在创建以后存活时间达到该时长后自动销毁。单位为分钟，最小值 60 分钟，最大 10080 分钟（7 天）。
ban	boolean	是否已开启聊天室全体禁言，默认 false。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200,"chatroomId":"chatroom001","createTime":1694777649942,"memberCount":0,"destroyType":1,"destr
e":120,"ban":true}
```

聊天室销毁机制

更新时间:2024-08-30

聊天室业务支持通过多种方式灵活控制聊天室的销毁条件与存活时长。

聊天室自动销毁机制

聊天室具有自动销毁机制。如果创建聊天室时未配置自动销毁类型、且后续未通过 API 设置自动销毁类型，则默认自动销毁类型为**不活跃后自动销毁**。聊天室会在不活跃达到 1 小时后踢出所有成员，并自动销毁（“不活跃”是指连续时间段内无成员进出且无新消息产生）。

您可以从控制台修改 App Key 级别默认设置，调整聊天室不活跃时长达到多少小时后应被自动销毁。自助配置最大支持配置为 24 小时。详见调整聊天室销毁等待时间。

一旦在调用 API 创建聊天室时配置了自动销毁类型，无论是否通过 API 设置销毁时间，均以 API 级别配置为准（包括默认值），App Key 级别的配置不生效。

设置自动销毁类型

您可以在创建聊天室时，可通过参数配置是否定时销毁该聊天室，或在不活跃时销毁该聊天室。

- 如果设置为不活跃后销毁，可通过 `destroyTime` 参数传入聊天室不活跃时长（分钟），控制连续不活跃（无人进出且无新消息产生）达到多久时应被自动销毁。
- 如果设置定时自动销毁，可通过 `destroyTime` 参数配置存活时长（分钟），存活时长自聊天室创建后开始计算。采用此销毁类型的聊天室最长存活 10800 分钟（7 天）。

在聊天室创建以后，您可以通过设置聊天室销毁类型 修改指定聊天室的自动销毁类型。

聊天室不被自动销毁的情况

- 绑定音视频房间: 聊天室与音视频房间绑定成功后，当聊天室达到预设的自动销毁条件时，服务端会先检测已绑定的音视频房间（RTCRoomId）是否仍存在。如果绑定的音视频房间仍存在，则则阻止聊天室自动销毁。如果绑定的音视频房间已销毁，则直接销毁聊天室。
- 保活聊天室：只要聊天室已被保活，就可确保聊天室不被自动销毁。保活状态下的聊天室只能通过调用销毁聊天室 API 接口销毁。

设置聊天室销毁类型

更新时间:2024-08-30

设置聊天室自动销毁类型。在设置前请确保您已了解聊天室销毁机制。

- 如果设置 `destroyType` 为不活跃后销毁，可通过 `destroyTime` 参数传入聊天室不活跃时长（分钟），控制连续不活跃（无人进出且无新消息产生）达到多久时应被自动销毁。
- 如果设置 `destroyType` 为固定时间销毁（定时自动销毁），可通过 `destroyTime` 参数配置存活时长（分钟），存活时长始终从聊天室创建后开始计算。采用此销毁类型的聊天室最长存活 10800 分钟（7 天）。

> 提示
>
> 如果聊天室为固定时间销毁类型，服务端仍会记录其处于不活跃状态的累计时长。如果聊天室已长时间处于不活跃状态，此时更改销毁类型为不活跃时销毁有可能导致聊天室立即销毁。

请求方法

POST： https://数据中心域名/chatroom/destroy/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID
destroyType	int	否	指定聊天室的销毁方式。`0`：默认值，表示不活跃时销毁。默认情况下，所有聊天室的自动销毁方式均为不活跃时销毁，一旦不活跃长达到 60 分钟即被销毁，可通过 `destroyTime` 延长该时间。`1`：固定时间销毁，设置为该类型后，聊天室默认在创建 60 分钟后自动销毁，可通过 `destroyTime` 设置更长的存活时间。
destroyTime	int	否	设置聊天室销毁时间。在 `destroyType=0` 时，表示聊天室应在不活跃达到该时长时自动销毁。在 `destroyType=1` 时，表示聊天室应在创建以后存活时间达到该时长后自动销毁。单位为分钟，最小值 60 分钟，最大 10080 分钟（7 天）。如果未设置，默认 60 分钟。

请求示例

```
POST /chatroom/destroy/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

chatroomId=gid1&destroyType=0&destroyTime=120
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码。200 表示处理成功。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200}
```

销毁聊天室

更新时间:2024-08-30

主动销毁指定的单个或多个聊天室。

请求方法

POST： https://数据中心域名/chatroom/destroy.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	要销毁的聊天室的 ID。每次可销毁多个聊天室。

请求示例

```
POST /chatroom/destroy.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=10001&chatroomId=10002
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

绑定音视频房间

更新时间:2024-08-30

聊天室与音视频房间绑定成功后，当聊天室达到预设的自动销毁条件时，服务端会先检测已绑定的音视频房间（RTCRoomId）是否仍存在：

- 如果绑定的音视频房间仍存在，则则阻止聊天室自动销毁。。
- 如果绑定的音视频房间已销毁，则直接销毁聊天室。

该接口仅创建从聊天室到音视频房间的单向绑定关系。因此在绑定音视频房间后，音视频房间的主动销毁或自动销毁，并不会直接触发聊天室房间的销毁。关于音视频房间的销毁机制，详见音视频房间销毁机制。

适用场景

聊天室具有自动销毁机制。在使用 RTC 业务的 App 中，可能会配合使用 IM SDK 的聊天室业务实现直播聊天、弹幕、属性记录等功能。这种情况下，可以考虑将聊天室与音视频房间绑定，确保聊天室不会在语聊、直播结束前销毁，以免丢失关键数据。

在单独使用聊天室业务情况的下，无需调用该接口。

> 提示
>
> **关于聊天室自动销毁逻辑的说明：**
>
> 聊天室具有自动销毁机制，默认情况下所有聊天室会在不活跃（连续时间段内无成员进出且无新消息）达到 1 小时后踢出所有成员并自动销毁，可延长该时间，也可配置为定时自动销毁。详见聊天室销毁机制。

请求方法

调用该接口必须传入聊天室 ID，因此必须在聊天室房间已创建成功之后调用。客户端调用加入聊天室接口时会自动完成创建与加入动作。

该接口不具备用户权限控制功能，建议由业务侧的房主或者主播角色用户加入聊天室房间成功后调用一次，其他用户无需调用。

POST： https://数据中心域名/chatroom/correlation/rtc.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文中可包含如下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID
rtcroomId	String	是	音视频房间 ID

请求示例

```
POST /chatroom/correlation/rtc.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

rtcroomId=jlk456j5&chatroomId=jlk456j6
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

创建聊天室（废弃）

更新时间:2024-08-30

通过 Server API 创建聊天室后，如 1 小时内该聊天室中没有发送过新消息或没有用户加入时，则该聊天室将自动销毁。

> 提示
>
> 加入聊天室一般为用户直接从客户端发起。

请求方法

POST： https://数据中心域名/chatroom/create.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroom[id]=name	String	是	新建的聊天室的 ID，支持大小写英文字母与数字的组合，最大长度 64 字节。name 为聊天室的名称。每次可创建多个聊天室，最多 100 个。

请求示例

```
POST /chatroom/create.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroom[10001]=name1&chatroom[10002]=name2&chatroom[10003]=name3
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室信息（废弃）

更新时间:2024-08-30

> 提示
>
> 该接口已废弃。请使用新接口获取聊天室基础信息。

查询聊天室基础信息，包括：聊天室 ID、名称、创建时间

请求方法

POST https://数据中心域名/chatroom/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	要查询的聊天室 ID，单次请求最多查询20个聊天室

请求示例

```
POST /chatroom/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=10001&chatroomId=10002
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。

返回值	返回类型	说明
chatRooms	Array of objects	聊天室信息数组。
chatRooms[].chrmId	String	聊天室 ID。
chatRooms[].name	String	聊天室名称。
chatRooms[].time	String	聊天室创建时间。精确到秒，格式为 YYYY-MM-DD HH:MM:SS，例如 2022-09-25 16:12:38。注意：time 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 time 为北京时间。如您的 App 业务使用海外数据中心，则 time 为 UTC 时间。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"chatRooms":[{
"chrmId":"10001","name":"name1","time":"2014-01-01 1:1:1"
},
{
"chrmId":"10002","name":"name2","time":"2014-01-01 1:1:2"
}]
}
```

保活聊天室

更新时间:2024-08-30

聊天室 1 小时内无人说话，同时没有人加入聊天室时，即时通讯服务端会自动把聊天室内所有成员踢出聊天室并销毁聊天室。聊天室保活功能，可以确保聊天室在此状态下不被自动销毁，只能通过调用 API 接口销毁聊天室。

开通服务

使用聊天室保活功能前，请确认已为当前 **App Key** 开通相关服务。开通后可设置 5 个聊天室为保活状态，如需要多个请联系商务。

如未开通服务，Server API 将返回 1009 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 1008）。

请求方法

POST： https://数据中心域名/chatroom/keepalive/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。

请求示例

```
POST /chatroom/keepalive/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消保活聊天室

更新时间:2024-08-30

从 App 的聊天室保活列表中移除指定聊天室。

请求方法

POST： https://数据中心域名/chatroom/keepalive/remove.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。

请求示例

```
POST /chatroom/keepalive/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询保活聊天室

更新时间:2024-08-30

查询已设置保活的聊天室。聊天室设置保活后，不会被自动销毁，只能通过调用 API 接口销毁聊天室。

请求方法

POST： https://数据中心域名/chatroom/keepalive/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

无

请求示例

```
POST /chatroom/keepalive/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
chatroomIds	String[]	保活聊天室数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"chatroomIds":["1000","1001"]}
```

聊天室状态同步

更新时间:2024-08-30

聊天室状态同步是即时通讯服务提供的回调服务，支持将应用下聊天室的状态变化实时同步到指定的应用服务器地址。目前支持同步以下状态：

- 创建聊天室
- 销毁聊天室
- 成员加入聊天室
- 成员退出聊天室

> 提示
>
> 不支持同一用户多端同时在线情况下的聊天室状态同步。

开通服务

使用**聊天室状态同步** 功能前，请确认已为当前 **App Key** 开通相关服务。详见聊天室服务配置。

开通服务时，请配置可正常访问的回调接收地址。如果您的网络有 IP 访问限制，请务必配置 IP 白名单，否则无法正常接收服务端回调。

回调方法

请求方法： POST

数据格式： `application/json`

即时通讯服务端会在 POST 请求 URL 中添加签名参数，您可通过签名验证调用者身份和数据有效性，详细参见 服务端回调签名。

正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	说明
chatRoomId	String	聊天室 ID。
userIds	String[]	用户 ID 数据。

参数	类型	说明

status	Int	操作状态取值如下：
		• `0`：操作者主动调用了聊天室事件类型（见 `type` 字段）对应的客户端或服务端接口。
		• `1`：聊天室自动退出机制被触发，导致用户退出聊天室（被踢出聊天室）。
		  • 聊天室成员在离线 30 秒后有新消息产生时，或离线后聊天室中产生 30 条消息时会被自动退出聊天室。
		  • 此状态的同步需要在聊天室中有新消息时才会进行触发。
		• `2`：用户被封禁。
		• `3`：聊天室自动销毁机制被触发。

type	Int	聊天室事件类型取值如下：`0` 为创建聊天室、`1` 加入聊天室、`2` 退出聊天室、`3` 销毁聊天室。
time	Int	发生时间。

回调代码示例

以下示例假设您在开通服务页面配置的回调接收地址为 `http://example.com/chatroom_status_sync.php`。

```
POST /chatroom_status_sync.php?
appKey=someappKey&timestamp=1408710653491&nonce=14314&signature=45beb7cc7307889a8e711219a47b7cf6a5b000e8
HTTP/1.1
Host: example.com
Content-Type: application/json

[
{
"chatRoomId":"destory_11",
"userIds":["gggg"],
"status":0,
"type":1,
"time":1574476797772
},
{
"chatRoomId":"destory_12",
"userIds":[],
"status":0,
"type":0,
"time":1574476797772
}
]
```

响应回调请求

> 提示
>
> - 只要有 HTTP `200 OK` 成功响应，服务端会认为状态已经同步。
> - 如果应答超时 5 秒，服务端会再尝试推送 2 次，如果仍然失败，将不再同步此条状态。
> - 如短时间内有大面积超时，将暂时停止请求您的服务器，1 分钟后会继续发送回调请求。异常断网情况

下的会延迟 5 分钟同步。

获取聊天室成员

更新时间:2024-08-30

获取指定聊天室中成员用户 ID，最多返回 500 个成员信息，支持按加入时间正序、倒序方式获取。

请求方法

POST https://数据中心域名/chatroom/user/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	要查询的聊天室 ID
count	String	是	要获取的聊天室成员信息数，最多返回 500 个成员信息
order	String	是	加入聊天室的先后顺序， 1 为加入时间正序， 2 为加入时间倒序

请求示例

```
POST /chatroom/user/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=10001&count=2&order=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	200：成功。
total	Int	当前聊天室中用户数。

返回值	返回类型	说明
users	String[]	聊天室成员数组，最多为 500 个。
id	String	用户 ID。
time	String	加入聊天室时间。精确到秒，格式为 YYYY-MM-DD HH:MM:SS，例如 2022-09-25 16:12:38。注意：time 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 time 为北京时间。如您的 App 业务使用海外数据中心，则 time 为 UTC 时间。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"total":500,
"users":[
{
"id":"uid1",
"time":"2015-09-10 16:38:26"
},
{
"id":"uid2",
"time":"2015-09-10 16:38:26"
}]
}
```

查询是否在聊天室中

更新时间:2024-08-30

查询指定用户是否在指定的聊天室中。

> 提示
>
> 用户是否在聊天室中可能会收到聊天室自动退出机制的影响。

了解聊天室离线成员自动退出机制

聊天室具有离线成员自动退出机制。用户离线后，如满足以下默认预设条件，即时通讯服务端会自动将该用户踢出聊天室：

- 从用户离线开始 30 秒内，聊天室中产生第 31 条消息时，触发自动踢出。
- 或用户已离线 30 秒后，聊天室有新消息产生时，触发自动踢出。

> 提示
>
> - 默认预设条件均要求聊天室中必须要有新消息产生，否则无法触发踢出动作。如果聊天室中没有消息产生，则无法将异常用户踢出聊天室。
> - 如需修改默认行为对新消息的依赖，请提交工单申请开通聊天室成员异常掉线实时踢出。开通该服务后，服务端会通过 SDK 行为（要求 Android/iOS IMLib SDK 版本 ≧ 5.1.6，Web IMLib 版本 ≧ 5.3.2）判断用户是否处于异常状态，最迟 5 分钟可以将异常用户踢出聊天室。
> - 如需保护特定用户，即不自动踢出指定用户（如某些应用场景下可能希望用户驻留聊天室），可使用 Server API 提供的聊天室用户白名单功能。

请求方法

POST https://数据中心域名/chatroom/user/exist.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	要查询的聊天室 ID

参数	类型	必传	说明
userId	String	是	要查询的用户 ID

请求示例

```
POST /chatroom/user/exist.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=10001&userId=5
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	200：成功。
isInChrm	Boolean	用户是否在聊天室中，true 表示在聊天室中，false 表示不在聊天室中。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"isInChrm":true}
```

批量查询是否在聊天室中

更新时间:2024-08-30

批量查询用户是否在指定聊天室中，每次查询最多不超过 1000 个。

> 提示
>
> 用户是否在聊天室中可能会收到聊天室自动退出机制的影响。

了解聊天室离线成员自动退出机制

聊天室具有离线成员自动退出机制。用户离线后，如满足以下默认预设条件，即时通讯服务端会自动将该用户踢出聊天室：

- 从用户离线开始 30 秒内，聊天室中产生第 31 条消息时，触发自动踢出。
- 或用户已离线 30 秒后，聊天室有新消息产生时，触发自动踢出。

> 提示
>
> - 默认预设条件均要求聊天室中必须要有新消息产生，否则无法触发踢出动作。如果聊天室中没有消息产生，则无法将异常用户踢出聊天室。
> - 如需修改默认行为对新消息的依赖，请提交工单申请开通聊天室成员异常掉线实时踢出。开通该服务后，服务端会通过 SDK 行为（要求 Android/iOS IMLib SDK 版本 ≧ 5.1.6，Web IMLib 版本 ≧ 5.3.2）判断用户是否处于异常状态，最迟 5 分钟可以将异常用户踢出聊天室。
> - 如需保护特定用户，即不自动踢出指定用户（如某些应用场景下可能希望用户驻留聊天室），可使用 Server API 提供的聊天室用户白名单功能。

请求方法

POST https://数据中心域名/chatroom/users/exist.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	要查询的聊天室 ID

参数	类型	必传	说明
userId	String	是	要查询的用户 ID，每次最多不超过 1000 个用户 ID

请求示例

```
POST /chatroom/users/exist.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=10001&userId=y41z2IXBW&userId=niCtlxnas
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	200：成功。
userid	String	聊天室中用户 ID。
isInChrm	Int	用户是否在聊天室中，1 表示在聊天室中，0 表示不在聊天室中。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"result":[
{"userid":"y41z2IXBW", "isInChrm":0},
{"userid":"niCtlxnas", "isInChrm":1}
]
}
```

禁言指定聊天室用户

更新时间:2024-08-30

聊天室业务支持禁言用户功能，可设置指定的一个或多个用户在指定聊天室中禁言。App 可使用该功能禁言 App 业务中的聊天室成员，被禁言用户可以接收查看聊天室中其他用户聊天信息，但不能发送消息。

用户退出聊天室不会使禁言状态失效。

> 提示
>
> 服务端（Server API）发送聊天室消息接口不受禁言状态的限制，被禁言用户可通过 Server API 往该聊天室中发送消息。

请求方法

POST： https://数据中心域名/chatroom/user/gag/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，可同时禁言多个用户，最多不超过 20 个。
chatroomId	String	是	聊天室 ID。
minute	String	是	禁言时长，以分钟为单位，最大值为 43200 分钟。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/user/gag/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2583&userId=2582&chatroomId=16&minute=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消禁言指定聊天室用户

更新时间:2024-08-30

取消指定的一个或多个用户的在指定聊天室中的禁言状态。

请求方法

POST： https://数据中心域名/chatroom/user/gag/rollback.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，可同时移除多个用户，最多不超过 20 个。
chatroomId	String	是	聊天室 ID。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/user/gag/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2583&userId=2582&chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室禁言用户列表

更新时间:2024-08-30

获取在指定聊天室中被禁言的用户列表。

请求方法

POST： https://数据中心域名/chatroom/user/gag/list.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。

请求示例

```
POST /chatroom/user/gag/list.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	String	被禁言用户数组。
time	String	解禁时间。精确到秒，格式为 `YYYY-MM-DD HH:MM:SS`，例如 `2022-09-25 16:12:38`。注意：`time` 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 `time` 为北京时间。如您的 App 业务使用海外数据中心，则 `time` 为 UTC 时间。
userId	String	被禁言用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"users":[{
"time":"2015-09-25 16:12:38",
"userId":"2582"
}]
}
```

设置聊天室全体禁言

更新时间:2024-08-30

聊天室业务支持将聊天室成员全体禁言。App 可以调用该 API，将指定的聊天室加入服务端的全体禁言聊天室列表。

- 将聊天室加入全体禁言列表后，聊天室全体禁言立即生效。该聊天室的所有成员均不能通过客户端 SDK 往该聊天室内发送消息。如需允许部分例外用户，可将用户加入聊天室禁言用户白名单。详见加入聊天室全体禁言白名单。
- 如果聊天室解散，全体禁言数据会被清除。

> 提示
>
> 服务端（Server API）发送聊天室聊消息接口不受聊天室全体成员禁言状态的限制，被禁言用户可通过 Server API 往该聊天室中发送消息。

请求方法

POST： https://数据中心域名/chatroom/ban/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	需要设置为禁言的聊天室 ID。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/ban/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消聊天室全体禁言

更新时间:2024-08-30

取消指定的一个或多个聊天室的全体成员禁言状态。

请求方法

POST： https://数据中心域名/chatroom/ban/rollback.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	被移除禁言的聊天室 ID
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/ban/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室全体禁言列表

更新时间:2024-08-30

查询 App Key 下已设置全体成员禁言的聊天室列表。

请求方法

POST： https://数据中心域名/chatroom/ban/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
size	Int	否	获取聊天室禁言列表的每页条数，不传时默认为 50 条，上限为 1000 条。
page	Int	否	当前页面数，不传时默认获取第 1 页。

请求示例

```
POST /chatroom/ban/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1592295337000
Signature: ef03d19bb860b4e90ce7bd5f50652a744bc2ce92
Content-Type: application/x-www-form-urlencoded

size=50&page=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
chatroomIds	String[]	被全体禁言的聊天室数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"chatroomIds":[
"123",
"234"
]
}
```

查询聊天室全体禁言状态

更新时间:2024-08-30

检查指定的单个聊天室是否已设置全部成员禁言。

请求方法

POST： https://数据中心域名/chatroom/ban/check.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	要查询的聊天室 ID

请求示例

```
POST /chatroom/ban/check.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1592295337000
Signature: ef03d19bb860b4e90ce7bd5f50652a744bc2ce92
Content-Type: application/x-www-form-urlencoded

chatroomId=123
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
status	Number	禁言状态，1 为全体禁言、0 为非全体禁言

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"status": 0
}
```

加入聊天室全体禁言白名单

更新时间:2024-08-30

添加一个或多个聊天室成员到指定聊天室的全体成员禁言白名单。

- 加入聊天室的禁言白名单后，在聊天室已被设置为全体禁言时，该成员仍可通过客户端 SDK 往该聊天室发送消息。
- 在白名单的用户如果主动退出聊天室或因封禁退出聊天室，再次加入时，白名单状态仍然有效。
- 聊天室销毁后再创建，原白名单状态自动失效。

禁言白名单服务与其他禁言功能关系说明：

- 聊天室单人禁言不影响白名单生效。即使用户已在聊天室中被设置了单人禁言，在用户添加到禁言白名单后，即可通过客户端 SDK 往聊天室中发送消息。
- 聊天室全局禁言影响白名单功能生效。如果出现将该用户添加到禁言白名单，该用户仍无法通过客户端 SDK 往聊天室中发送消息的情况，可检查该用户是否已被设置为 App Key 下所有聊天室中禁言。

> 提示
>
> 服务端（Server API）发送群聊消息接口不受聊天室全体成员禁言状态的限制，被禁言用户可通过 Server API 往该聊天室中发送消息。

请求方法

POST： https://数据中心域名/chatroom/user/ban/whitelist/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID
userId	String	是	需要添加到白名单中的用户 ID，白名单中用户上限为 20 个，支持批量添加，单次添加上限不超过 20 个。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/user/ban/whitelist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=LoDld8izA&userId=uu1&userId=uu2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移出聊天室全体禁言白名单

更新时间:2024-08-30

从指定聊天室的全体禁言白名单中移除一个或多个用户。

请求方法

POST： https://数据中心域名/chatroom/user/ban/whitelist/rollback.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID
userId	String	是	需要移除白名单的用户 ID，支持批量添加，单次添加上限不超过 20 个。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/user/ban/whitelist/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=LoDld8izA&userId=uu1&userId=uu2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室全体禁言白名单

更新时间:2024-08-30

获取指定单个聊天室的全体禁言白名单。

请求方法

POST： https://数据中心域名/chatroom/user/ban/whitelist/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID

请求示例

```
POST /chatroom/user/ban/whitelist/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1592295337000
Signature: ef03d19bb860b4e90ce7bd5f50652a744bc2ce92
Content-Type: application/x-www-form-urlencoded

chatroomId=LoDld8izA
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
userIds	String[]	聊天室中白名单用户数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code": 200,
"userIds": [
"uu2",
"uu1"
]
}
```

全局禁言用户

更新时间:2024-08-30

聊天室业务支持将指定用户在应用下的所有聊天室中禁言，支持设置禁言时长。

在禁言时间段内，被禁言用户在应用下的所有聊天室中都无法通过客户端 SDK 发送消息。

开通服务

使用聊天室全局禁言功能前，请确认已为当前 **App Key** 开通相关服务。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/chatroom/user/ban/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，可同时禁言多个用户，每次最多不超过 20 个。
minute	String	是	禁言时长，以分钟为单位，最大值为 43200 分钟。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：指定聊天室中所有成员。

请求示例

```
POST /chatroom/user/ban/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2582&userId=2583&minute=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

取消全局禁言用户

更新时间:2024-08-30

取消指定的一个或多个用户的聊天室全局禁言状态。

请求方法

POST： https://数据中心域名/chatroom/user/ban/remove.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，可同时移除多个用户，每次最多不超过 20 个。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：被解除全局禁言的用户。

请求示例

```
POST /chatroom/user/ban/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2582
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询全局禁言用户列表

更新时间:2024-08-30

查询 App Key 下已被设置聊天室全局禁言的用户列表，返回结果包含解禁时间。

请求方法

POST： https://数据中心域名/chatroom/user/ban/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

无

请求示例

```
POST /chatroom/user/ban/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
users	String	被禁言用户数组。
time	String	解禁时间。精确到秒，格式为 YYYY-MM-DD HH:MM:SS，例如 2022-09-25 16:12:38。注意：time 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 time 为北京时间。如您的 App 业务使用海外数据中心，则 time 为 UTC 时间。
userId	String	被禁言用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"users":[{
"time":"2015-09-25 16:12:38",
"userId":"2582"
}]
}
```

封禁聊天室用户

更新时间:2024-08-30

聊天室业务支持封禁用户功能，支持在指定聊天室中封禁一个或多个用户。

- 被封禁时，如果用户如在聊天室中，将立即被踢出该聊天室。
- 被封禁用户在封禁时间内不能加入此聊天室。

请求方法

POST： https://数据中心域名/chatroom/user/block/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，可同时封禁多个用户，最多不超过 20 个。
chatroomId	String	是	聊天室 ID。
minute	String	是	封禁时长，以分钟为单位，最大值为43200分钟。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：聊天室中所有成员，包括被封禁用户。

请求示例

```
POST /chatroom/user/block/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2583&userId=2582&chatroomId=16&minute=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

解除封禁聊天室用户

更新时间:2024-08-30

将指定的用户移出指定聊天室的封禁用户列表，即解除聊天室对用户的封禁状态。

请求方法

POST： https://数据中心域名/chatroom/user/block/rollback.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID，可同时移除多个用户，最多不超过 20 个。
chatroomId	String	是	聊天室 ID。
extra	String	否	通知携带的 JSON 格式的扩展信息，仅在 `needNotify` 为 `true` 时有效。
needNotify	boolean	否	是否通知成员。默认 `false` 不通知。如果为 `true`，客户端会触发相应回调方法（要求 Android/iOS IMLib ≧ 5.4.5；Web IMLib ≧ 5.7.9）。通知范围：被解除封禁的成员。

请求示例

```
POST /chatroom/user/block/rollback.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=2583&userId=2582&chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室封禁用户

更新时间:2024-08-30

获取被指定聊天室封禁的用户列表。被禁用户在封禁时间内不能再进入此聊天室中。

请求方法

POST： https://数据中心域名/chatroom/user/block/list.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。

请求示例

```
POST /chatroom/user/block/list.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	String	被封禁用户数组。
time	String	解禁时间。精确到秒，格式为 `YYYY-MM-DD HH:MM:SS`，例如 `2022-09-25 16:12:38`。注意：`time` 的值与应用所属数据中心有关。如您的 App 业务使用国内数据中心，则 `time` 为北京时间。如您的 App 业务使用海外数据中心，则 `time` 为 UTC 时间。
userId	String	被封禁用户 ID。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"users":[{
"time":"2015-09-25 16:12:38",
"userId":"2582"
}]
}
```

聊天室属性概述

更新时间:2024-08-30

聊天室属性（KV）管理接口用于在指定聊天室中设置自定义属性。在语音直播聊天室场景中，可利用此功能记录聊天室中各麦位的属性；或在狼人杀等卡牌类游戏场景中记录用户的角色和牌局状态等。客户端可以设置聊天室属性变化监听器获取实时属性变化。

开通服务

使用聊天室属性（KV） 功能前，请确认已为当前 App Key 开通相关服务。详见开通聊天室自定义属性。开通服务后，客户端与服务端均可以设置聊天室自定义属性（KV）。

服务限制

- 每个聊天室中，最多允许设置 100 个属性信息，以 `Key-Value` 的方式进行存储。
- 聊天室销毁后，聊天室中的自定义属性同时销毁。
- 聊天室 KV 也受聊天室消息抛弃策略影响。如要确保设置属性通知消息不被丢弃，可以使用聊天室白名单服务中的用户白名单进行保护。详见开通聊天室白名单服务。
- 调用客户端与服务端 API 均可以操作聊天室属性 KV，当前即时通讯服务未对总操作频率进行限制。为确保 App 业务正常运行，建议将单个聊天室房间操作 `Key-Value` 的总频率维持在100 对/每秒及以下（一秒内单次操作 100 对 KV）。

属性管理接口

开通服务后，可以使用以下接口：

- 设置聊天室属性: 在指定聊天室中设置一个自定义属性 KV 对。该接口要求同时传入操作用户 ID，作为该 KV 对的所属用户。
- 批量设置聊天室属性: 在指定聊天室中批量、强制设置自定义属性 KV 对，单次最多 20 个 KV 对。如果属性已存在，该接口会强制覆盖原有 KV 对的所属用户 ID 和 Value。
- 删除聊天室属性: 在指定聊天室中删除一个自定义属性 KV 对。该接口要求同时传入该 KV 对的所属用户 ID。
- 批量删除聊天室属性: 在指定聊天室中批量、强制删除自定义属性 KV 对，单次最多 20 个 KV 对。该接口会强制删除 KV 对，不要求传入待删除 KV 对的所属用户 ID。应用中管理员角色用户可使用该接口清除指定聊天室内的业务数据。

监控聊天室自定义属性变化

您可以使用以下方式监控聊天室属性的变化：

- 服务端回调：您可以在控制台自助配置接收聊天室属性的回调 URL。即时通讯服务端会将应用下的全部聊天室属性变化（设置，删除，全部删除等操作）同步到您指定的地址，方便 App 业务服务端了解聊天室属性变化。详见聊天室属性同步（KV）。
- 客户端回调：客户端 SDK 提供了聊天室属性变化的监听器，可监听聊天室属性同步、设置、删除等操作。

设置聊天室属性（KV）

更新时间:2024-08-30

在指定聊天室中设置一个自定义属性 KV 对。该接口要求同时传入操作用户 ID，作为该 KV 对的所属用户。

聊天室销毁后，聊天室中的自定义属性同时销毁。

开通服务

使用聊天室属性（KV） 功能前，请确认已为当前 App Key 开通相关服务。详见聊天室属性概述。

如未开通服务，Server API 将返回 1009 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 1008）。

请求方法

POST： https://数据中心域名/chatroom/entry/set.json

频率限制： 每秒钟限操作 100 对 KV，与批量设置接口 /chatroom/entry/batch/set.json 共享限频配额。如果仅使用设置单个属性的 API，则一秒可调用 100 次。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。
userId	String	是	操作用户 ID。通过 Server API 非聊天室中用户可以进行设置。
key	String	是	聊天室属性名称，Key 支持大小写英文字母、数字、部分特殊符号 + = - _ 的组合方式，大小写敏感。最大长度 128 字符。每个聊天室中，最多允许设置 **100** 个属性 Key-Value 对。
value	String	是	聊天室属性对应的值，最大长度 4096 个字符。
autoDelete	Int	否	属性的操作用户退出聊天室后，是否删除此 Key 值。为 1 时删除此 Key 值和对应的 Value，为 0 时用户退出后不删除，默认为 0。
objectName	String	否	聊天室属性变化通知消息的消息类型，一般为内置消息类型 RC:chrmKVNotiMsg，也可以是其他自定义消息类型。如果传入该字段，则在聊天室属性变化时发送一条消息。 **注意**：旧版客户端 SDK（Android/iOS < 4.0.2、Web < 3.0.6）依赖该通知消息从服务端同步聊天室属性变化。
content	String	否	聊天室属性变化通知消息的消息内容。JSON 结构，当 objectName 为 RC:chrmKVNotiMsg 时，content 必须包含 type、key、value 属性。RC:chrmKVNotiMsg 消息结构的详细说明请参见 聊天室属性通知消息。

请求示例

```
POST /chatroom/entry/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=kvchatroom2&userId=Lnq9MJsPY&key=huihui&value=555&autoDelete=0&objectName=RC%3AchrmKVNotiMsg&
nt=%7B%22key%22%3A%22keyli%22%2C%22value%22%3A%225%22%2C%22type%22%3A%221%22%7D&extra=111111
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

批量设置聊天室属性（KV）

更新时间:2024-08-30

在指定聊天室中批量设置自定义属性 KV 对，单次最多 20 个 KV 对。如果属性已存在，该接口会强制覆盖原有 KV 对的所属用户 ID 和 Value。

聊天室销毁后，聊天室中的自定义属性同时销毁。

开通服务

使用聊天室属性（KV） 功能前，请确认已为当前 App Key 开通相关服务。详见聊天室属性概述。

如未开通服务，Server API 将返回 1009 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 1008）。

请求方法

POST： https://数据中心域名/chatroom/entry/batch/set.json

频率限制： 每秒钟限操作 100 对 KV，与 /chatroom/entry/set.json 共享限频配额。如：一次设置 1 个属性，一秒可调用 100 次。一次设置 20 个属性，一秒可调用 5 次。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

请求参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID
autoDelete	Int	否	用户（entryOwnerId）退出聊天室后，是否删除此 Key 值。为 1 时删除此 Key 值和对应的 Value，为 0 时用户退出后不删除，默认为 0。
entryOwnerId	String	是	聊天室自定义属性的所属用户 ID
entryInfo	String	是	聊天室自定义属性 KV 对，JSON 结构，一次最多 20 个 KV。Key 为属性名，支持大小写英文字母、数字、部分特殊符号 + = - _ 的组合方式，大小写敏感。最大长度 128 字符。Value 为属性值，最大长度 4096 个字符。

请求示例

```
POST /chatroom/entry/batch/set.json HTTP/1.1
Host: api-cn.ronghub.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

chatroomId=gid1&autoDelete=true&entryOwnerId=test&entryInfo={"key1":"value1","key2":"value2"}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码。200 表示处理成功

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200}
```

删除聊天室属性（KV）

更新时间:2024-08-30

在指定聊天室中删除一个自定义属性 KV 对。该接口要求同时传入该 KV 对的所属用户 ID。

请求方法

POST： https://数据中心域名/chatroom/entry/remove.json

频率限制： 每秒钟限操作 100 对 KV，与批量删除接口 `/chatroom/entry/batch/remove.json` 共享限频配额。如果仅使用删除单个属性的 API，则一秒可调用 100 次。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。
userId	String	是	操作用户 ID。通过 Server API 非聊天室中用户可以进行设置。
key	String	是	聊天室属性名称，Key 支持大小写英文字母、数字、部分特殊符号 + = - _ 的组合方式，大小写敏感。最大长度 128 字。
objectName	String	否	通聊天室属性变化通知消息的消息类型，一般为内置消息类型 `RC:chrmKVNotiMsg`，也可以是其他自定义消息类型。如果传入该字段，则在聊天室属性变化时发送一条消息。 注意：从客户端 SDK 版本 4.0.2（Android / iOS） 与 3.0.6 （Web）开始，SDK 默认从服务端同步聊天室属性变化，不再依赖该通知消息。客户端可以设置聊天室属性变化监听器获取实时属性变化。
content	String	否	聊天室属性变化通知消息的消息内容。JSON 结构，当 objectName 为 `RC:chrmKVNotiMsg` 时，content 必须包含 type、key、value 属性。`RC:chrmKVNotiMsg` 消息结构的详细说明请参见 聊天室属性通知消息。

请求示例

```
POST /chatroom/entry/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=kvchatroom2&userId=jrT1igbKr
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

批量删除聊天室属性（KV）

更新时间:2024-08-30

在指定聊天室中批量删除自定义属性 KV 对，单次最多 20 个 KV 对。该接口会强制删除 KV 对，不要求传入待删除 KV 对的所属用户 ID。应用中管理员角色用户可使用该接口清除指定聊天室内的业务数据。

请求方法

POST： https://数据中心域名/chatroom/entry/batch/remove.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

请求参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID
entryOwnerId	String	是	自定义属性的所属用户 ID。
entryKeys	Array	是	需要删除的 key 的集合。

请求示例

```
POST /chatroom/entry/batch/remove.json HTTP/1.1
Host: api-cn.ronghub.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

chatroomId=gid1entryOwnerId=test&entryKeys=["key1","key2"]
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码。200 表示处理成功

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200}
```

查询聊天室属性（KV）

更新时间:2024-08-30

查询指定聊天室的自定义属性（KV）。

请求方法

POST： https://数据中心域名/chatroom/entry/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。
keys	String	否	批量获取指定聊天室中的 Key 值，最多上限为 100 个，不传时获取全部 key 值。

请求示例

```
POST /chatroom/entry/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=kvchatroom2&keys=Lnq9MJsPY&keys=Lnq88JsPY
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
keys	Array	属性数组。
keys[i].key	String	设置的属性名。
keys[i].value	String	属性对应的内容。
keys[i].userId	String	最后一次设置此 Key 的用户 ID。

返回值	返回类型	说明
keys[i].autoDelete	String	用户退出聊天室后是否删除此 Key，0 为不删除、1为删除。
keys[i].lastSetTime	String	最近一次设置 Key 的时间。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"keys":
[{"key":"key1","value":"value1","userId":"2121","autoDelete":"0","lastSetTime":"2121"},
{"key":"key2","value":"value2","userId":"4343","autoDelete":"1"," lastSetTime":"2121"}]
}
```

聊天室属性同步（KV）

更新时间:2024-08-30

「聊天室属性同步」是即时通讯服务端的一项服务端回调服务。

开通该回调服务后，应用下聊天室属性（KV）发生变化（设置，删除，全部删除等操作）时，将实时同步到开发者的应用服务器，便于对应用下的聊天室进行业务逻辑处理。

开通服务

使用**聊天室属性同步（KV）** 功能前，请确认已为当前 **App Key** 开通相关服务。详见聊天室服务配置。

开通服务时，请配置可正常访问的回调接收地址。开通服务时，请配置可正常访问的回调接收地址。如果您的网络有 IP 访问限制，请务必配置 IP 白名单，否则无法正常接收服务端回调。即时通讯服务端会将应用下的聊天室属性变化（设置，删除，全部删除等操作）同步到指定的回调地址。

回调方法

请求方法： POST

数据格式： `application/json`

即时通讯服务端会在 POST 请求 URL 中添加签名参数，您可通过签名验证调用者身份和数据有效性，详细参见 服务端回调签名。

正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

参数	类型	说明
chatroomId	String	聊天室 ID。
key	String	属性的 Key。
value	String	属性的 Value。
optType	Int	操作类型：1 设置、2 删除、3 删除全部属性（KV）。
userId	String	操作者的用户 ID。
status	Int	本字段暂不可用，请勿使用。
timestamp	Int	服务端记录的时间戳。
version	Int	属性的版本号（业务进行操作排序用）。

同步状态时需要服务提供应答 200，只要有 HTTP 应答码 200 ，服务端会认为状态已经同步。如果应答超时 5 秒，服务端会再尝试推送 2 次，如果仍然失败，将不再同步此条状态。如短时间内有大面积超时，将

暂停推送，1 分钟后会继续推送。异常断网情况下的会延迟 5 分钟同步。

回调请求示例

假设您在开通服务页面配置的接收地址：http://example.com/chatroom_kv_sync.php

```
POST /chatroom_kv_sync.php?
appKey=someappKey&timestamp=1408710653491&nonce=14314&signature=45beb7cc7307889a8e711219a47b7cf6a5b000e8
HTTP/1.1
Host: example.com
Content-Type: application/json

[
{
"chatroomId":"kvchatroom2",
"optType":1,
"userId":"1DBrZTGCI",
"key":"testKey",
"value":"testValue",
"status":"2",
"timestamp":1645437940739,
"version":1645437940738
},
{
"chatroomId":"kvchatroom3",
"optType":2,
"userId":"testUser",
"key":"testKey1",
"status":"2",
"value":"testValue",
"timestamp":1645437940740,
"version":1645437940740
}
]
```

添加低级别消息类型

更新时间:2024-08-30

为保证聊天室中重要消息正常下发，聊天室业务默认在消息量较大的情况下，可能会按消息发送的时间顺序丢弃超出消费上限的最新消息，确保服务器稳定。

App 可以通过聊天室业务提供的**聊天室低级别消息**功能，配置最多 20 个消息类型为低级别消息类型。当服务器负载高时，如果接收到低级别消息，则优先丢弃。例如，App 业务自定义了点赞消息，可配置为低级别消息类型，在聊天室消息量大时优先抛弃。

- 默认情况下，所有消息均为高级别消息。服务端默认单个聊天室中上行消息处理能力是每 200 毫秒 40 条。其中 20 条为高优先级消息专用配额。另 20 条的配额为高优先级消息和优先级消息共享配额。设置低级别消息类型后，服务端在高负载情况下会根据上述配额处理高级别与低级别消息，低级别消息优先丢弃，让出共享配额给高级别消息。
- 单个聊天室最多支持配置 20 个消息类型为低级别消息类型。默认均为高级别消息。

> 提示
>
> **聊天室低级别消息功能用于优先抛弃部分消息。如果需要保护重要消息，请使用聊天室用户白名单与聊天室消息白名单功能，详见聊天室白名单服务概述。**

开通服务

使用**聊天室低级别消息**功能前，请确认已为当前 **App Key** 开通相关服务。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/chatroom/message/priority/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
objectName	String	是	低优先级的消息类型，每次最多提交 5 个，设置的消息类型最多不超过 20 个。

请求示例

```
POST /chatroom/message/priority/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce:14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

objectName=RC:VcMsg&objectName=RC:ImgTextMsg&objectName=RC:ImgMsg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移除低级别消息类型

更新时间:2024-08-30

从聊天室低级别消息类型列表中移除一个或多个消息类型。

请求方法

POST： https://数据中心域名/chatroom/message/priority/remove.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
objectName	String	是	低优先级的消息类型，每次最多提交 5 个

请求示例

```
POST /chatroom/message/priority/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce:14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

objectName=RC:VcMsg&objectName=RC:ImgMsg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询低级别消息类型

更新时间:2024-08-30

查询聊天室低级别消息类型列表。

请求方法

POST： https://数据中心域名/chatroom/message/priority/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

无

请求示例

```
POST /chatroom/message/priority/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce:14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
objectNames	String[]	消息类型数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"objectNames":["RC:ImgMsg","RC:ImgTextMsg","RC:VcMsg"]
}
```

聊天室白名单服务概述

更新时间:2024-08-30

为保证聊天室中重要消息正常下发，聊天室业务默认在消息量较大的情况下，会按消息发送的时间顺序丢弃超出消费上限的最新消息，确保服务器稳定。

针对聊天室业务的以上特性，即时通讯服务提供**聊天室白名单服务**，可帮助 App 业务保护部分消息。开通服务后，可使用以下功能：

- 聊天室用户白名单：可用于保护指定聊天室中的重要用户，支持按聊天室设置白名单用户。例如，App 业务中指定聊天室中的管理员、主播等重要角色的用户。
- 聊天室消息白名单：可用于保护 App 下所有聊天室中的指定消息类型。例如 App 业务中自定义的红包消息。

加入聊天室用户白名单

更新时间:2024-08-30

聊天室业务提供聊天室用户白名单功能，可用于保护指定聊天室中的重要用户，例如，App 业务中具有管理员、主播等重要角色的用户。具体能力如下：

- **保护用户不被自动踢出聊天室**：聊天室业务具备离线用户自动踢出机制（在离线 30 秒后有新消息产生时，或离线后聊天室中产生 30 条消息时）。聊天室白名单用户离线后，不会被即时通讯服务端自动踢出该聊天室。
- **保护用户发送的聊天室消息**：聊天室业务具备消息丢弃机制。为保证聊天室中重要消息正常下发，聊天室业务默认在消息量较大的情况下，会按消息发送的时间顺序丢弃超出消费上限的最新消息，确保服务器稳定。聊天室白名单用户发送的消息在任何情况下会优先受到保护。
- App 可添加一个或多个用户到指定聊天室的用户白名单，单个聊天室最多支持 5 个白名单用户。

开通服务

使用聊天室用户白名单功能前，请确认已为当前 **App Key** 开通相关服务。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/chatroom/user/whitelist/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。
userId	String	是	聊天室中用户 ID，可提交多个。聊天室中白名单用户最多不超过 5 个。

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

请求示例

```
POST /chatroom/user/whitelist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16&userId=123&userId=456
```

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移出聊天室用户白名单

更新时间:2024-08-30

从指定聊天室的用户白名单中移除一个或多个用户。

请求方法

POST： https://数据中心域名/chatroom/user/whitelist/remove.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。
userId	String	是	聊天室白名单中用户 ID，可提交多个，最多不超过 5 个。

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

请求示例

```
POST /chatroom/user/whitelist/remove.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16&userId=123&userId=456
```

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室用户白名单

更新时间:2024-08-30

查询指定聊天室的用户白名单。

请求方法

POST： https://数据中心域名/chatroom/user/whitelist/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
chatroomId	String	是	聊天室 ID。

请求示例

```
POST /chatroom/user/whitelist/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

chatroomId=16
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
users	String[]	白名单用户数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"users":["user1","user2"]}
```

加入聊天室消息白名单

更新时间:2024-08-30

聊天室业务提供聊天室消息白名单功能，可用于保护 App 聊天室业务中的重要消息，例如 App 业务中自定义的红包消息。

App 可以添加一种或多种消息类型到消息类型白名单中。在聊天室消息量较大的情况下，聊天室消息类型白名单的消息优先受到保护。

- 聊天室消息类型白名单最多支持 20 个消息类型。App 下所有聊天室共享一个消息类型白名单。
- 聊天室消息白名单与聊天室用户白名单可以同时使用。

开通服务

使用聊天室消息白名单功能前，请确认已为当前 **App Key** 开通相关服务。

如未开通服务，Server API 将返回 `1009` 错误。注意，在未开通服务时，如果连续请求导致 API 请求频率超过限制，Server API 会返回 HTTP 429 Too Many Requests 错误（错误码为 `1008`）。

请求方法

POST： https://数据中心域名/chatroom/whitelist/add.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
objectnames	String	是	消息标识，最多不超过 20 个，自定义消息类型，长度不超过 32 个字符

请求示例

```
POST /chatroom/whitelist/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb7j
Nonce: 1167631608
Timestamp: 1408710653491
Signature: c8ef4d25e5684e9fca820140eaf6a36abc4fbc93
Content-Type: application/x-www-form-urlencoded

objectnames=RC:VcMsg&objectnames=RC:ImgTextMsg&objectnames=RC:ImgMsg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移出聊天室消息白名单

更新时间:2024-08-30

从聊天室消息类型白名单中移除一种或多种消息类型。

请求方法

POST： https://数据中心域名/chatroom/whitelist/delete.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
objectnames	String	是	消息标识，最多不超过 20 个，自定义消息类型，长度不超过 32 个字符

请求示例

```
POST /chatroom/whitelist/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: c9kqb3rdkbb7j
Nonce: 1167631608
Timestamp: 1408710653491
Signature: c8ef4d25e5684e9fca820140eaf6a36abc4fbc93
Content-Type: application/x-www-form-urlencoded

objectnames=RC:VcMsg&objectnames=RC:ImgTextMsg&objectnames=RC:ImgMsg
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询聊天室消息白名单

更新时间:2024-08-30

查询聊天室消息类型白名单。

请求方法

POST： https://数据中心域名/chatroom/whitelist/query.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

请求方法

无

请求示例

```
POST /chatroom/whitelist/query.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce:14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
whitlistMsgType	String[]	消息类型数组。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"whitlistMsgType":["RC:ImgMsg","RC:ImgTextMsg","RC:VcMsg"]
}
```

内容审核能力概述

更新时间:2024-08-30

即时通讯支持对 IM 内容进行审核，提供基础审核能力、专业审核服务，以及自行对接审核服务的能力。

- **内置敏感词机制**：即时通讯（IM）服务内置开箱即用的敏感词机制。注意，敏感词机制仅是一种基础保护机制，且仅限于文本内容（默认最多 50 个敏感词），不可替代专业内容审核服务。
- **专业 IM 审核服务**：内容审核服务产品中提供 IM 审核，可为 IM 内容提供全面的保障与支持，支持审核文本、图片、语音片段、小视频，精准识别敏感信息。
- **消息回调服务**：消息回调可将消息先行发送至 App 指定的地址，由 App 返回结果决定是否继续发送至收件人。App 可以使用消息回调服务自行实现审核，或自行对接第三方审核服务。

如果消息被判定违规导致无法下发收件人，默认情况下消息发送者不会收到通知。如果需要接收通知，方式如下：

- **客户端**：支持通知消息发件人。如果 App 希望消息发送者接收消息已被拦截的通知，可提交工单开通**含敏感词消息屏蔽状态回调发送端**，并在客户端设置监听（要求 Android/iOS SDK 版本 ≧ 5.1.4，Web ≧ 5.0.2）。详见各客户端「敏感信息拦截回调」文档。
- **服务端**：支持对触发**专业 IM 审核服务**的消息提供审核结果回调。审核通过与未通过的消息都会在该服务中返回。

敏感词机制

敏感词机制是一种基础保护机制，仅支持对文本消息内容中的敏感词进行识别与过滤。对命中敏感词的消息，您可以选择进行屏蔽该消息（不会下发给接收方），或按指定规则替换消息中的敏感词后再进行下发。

目前支持的敏感词过滤语言包括：中文、英文、日语、德语、俄语、韩语、阿拉伯语。

> 提示
>
> 如需审核文本（支持语义检测）、图片、语音片段、小视频，建议使用即时通讯服务提供的 内容审核服务产品。

您可以通过控制台或 API 方式管理 App Key 下开发环境或生产环境的敏感词：

- 控制台 **IM 服务**下的**敏感词设置**页面，支持添加、删除、导出等操作。详见配置敏感词服务。
- 服务端 API，支持添加、删除、获取敏感词等操作。详见消息敏感词。

默认行为

- 默认最多设置 50 个敏感词。
- 默认仅对从客户端 SDK 发送的消息生效。
- 默认仅支持识别官方内置的文本消息类型（消息标识为 `RC:TxtMsg`）中的敏感词。支持单聊、群聊、聊天室、超级群会

话。超级群中文本消息修改后的内容默认也会敏感词识别、拦截或过滤。

- 触发敏感词默认没有通知。

敏感词过滤规则

- **简体、繁体智能过滤**：设置中文简体敏感词后，对应繁体敏感词也会自动识别过滤。
- **智能忽略标点符号**：支持智能忽略字符串中的标点符号，匹配敏感词。例如：设置敏感词为 “反动”，如果消息内容中包含 “反，动”、也会被过滤。这种情况下 “反v动”、“反3动” 也会被过滤。
- **智能识别英文词**：支持对英文单词进行智能识别过滤，提升匹配的准确性。英文支持全角、半角，大、小写自动匹配。例如：设置敏感词 “AV”，只过滤完整单词 “AV”、"av"，不会对单词 “Java ” 中包含的 “av” 进行过滤。
- **数字敏感词**：支持匹配连续数字字符串。支持智能匹配全角数字。例如：设置敏感词 “123”，可过滤 “123”、“1 2 3”、“123 汉字”、“１ ２ ３”（全角），但无法过滤 “123467” 中的 “123” 片段。

IM 内容审核服务

如果您希望全面审核 IM 内容，可以使用内容审核服务产品，该产品提供 **IM 审核服务**与**音视频审核服务**。即时通讯服务服务端回调提供审核结果。

IM 审核针对即时通讯业务，具体可提供以下能力：

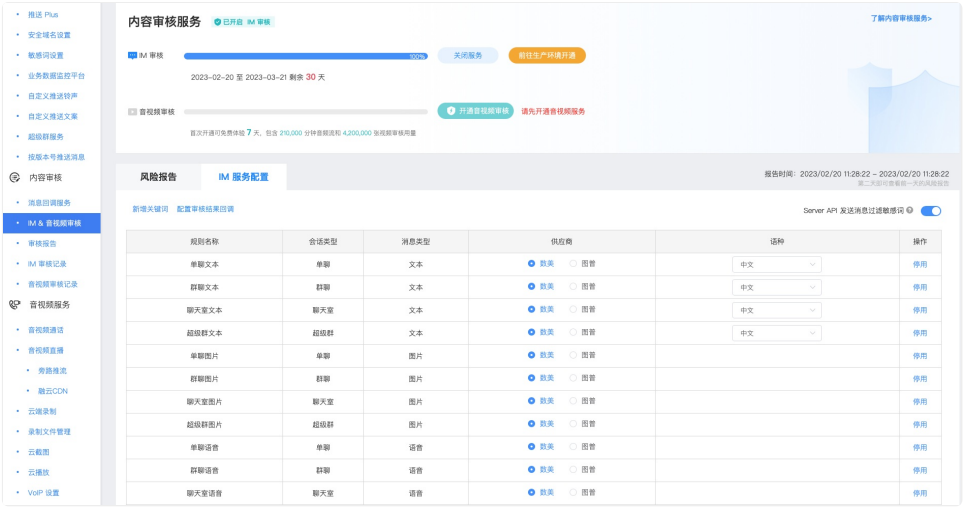
- 审核文本内容
- 审核图片
- 审核语音片段
- 审核小视频
- 审核自定义消息类型（需要提交工单申请）
- 审核超级群业务中的消息修改
- 从控制台查看审核报告
- 从控制台查询 IM 审核记录
- 审核结果回调

IM 内容审核计费

内容审核服务为付费服务，开发环境可免费体验，生产环境下需预存才能使用服务。具体计费说明详见资费标准 · IM 审核。

IM 内容审核服务配置

在控制台的 IM & 音视频审核 页面开通 **IM 审核服务**后，可以配置审核服务厂商、审核关键词、会话与消息类型等。



如需接收审核结果，请务必配置接收审核结果回调的地址。详见审核结果回调。

消息回调服务

如果您希望对接自己的审核系统或其他第三方内容审核服务，可以使用消息回调服务。

消息回调服务（原模版路由）提供一种消息过滤机制。您可以根据发送用户 ID、接收用户 ID、消息类型、会话类型等参数，将相应的消息同步到您指定的服务器。超级群业务中，修改消息内容、更新消息扩展也支持通过消息回调同步到您指定的服务器。

消息同步到您指定的服务器后，可以使用您自己的审核系统执行内容审核，也可以对接其他第三方审核系统。即时通讯服务端会根据您应用服务器返回的响应结果，决定是否将消息下发、是否替换消息中的内容，以及如何进行内容替换。

您可以通过控制台的消息回调服务页面管理 App Key 下开发环境或生产环境的消息回调服务状态和路由规则。

关于如何创建路由规则，以及回调参数的具体说明，请参见消息回调服务文档。

消息回调服务计费

费用以官方价格说明页面及计费说明文档为准。

添加消息敏感词

更新时间:2024-08-30

文本消息支持敏感词过滤、替换功能，敏感词替换支持单聊、群聊、聊天室、超级群会话，默认最多设置 50 个敏感词，设置 30 分钟后生效。

目前支持的敏感词过滤语言包括：中文、英文、日语、德语、俄语、韩语、阿拉伯语。

详见内容审核能力概述中的**敏感词过滤规则**。

开通服务

即时通讯服务内置敏感词机制，无需开通。注意，敏感词机制默认仅对客户端发送的消息生效，不对通过即时通讯服务端 API 发送的消息生效。如有需要，请开启 **Server API 发送消息过滤敏感词**。详见配置敏感词服务配置 。

请求方法

POST： https://数据中心域名/sensitiveword/add.json

调用频率：每秒钟限 100 次

签名规则：所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
word	String	是	敏感词，最长不超过 32 个字符，格式为汉字、数字、字母。
replaceWord	String	否	替换后的词，最长不超过 32 个字符。如未设置，当消息中含有敏感词时，消息将被屏蔽，用户不会收到消息。如设置了，当消息中含有敏感词时，将被替换为指定的词进行发送。

请求示例

```
POST /sensitiveword/add.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

word=money&replaceWord=****
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

移除消息敏感词

更新时间:2024-08-30

从敏感词列表中移除单个指定敏感词。

请求方法

POST： https://数据中心域名/sensitiveword/delete.json

调用频率：每秒钟限 100 次

签名规则：所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
word	String	是	敏感词，最长不超过 32 个字符。

请求示例

```
POST /sensitiveword/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

word=money
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

批量移除消息敏感词

更新时间:2024-08-30

从敏感词列表中批量移除指定敏感词。

请求方法

POST： https://数据中心域名/sensitiveword/batch/delete.json

调用频率：每秒钟限 100 次

签名规则：所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
words	String[]	是	敏感词数组，一次最多移除 50 个敏感词。

请求示例

```
POST /sensitiveword/batch/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

words=money&words=aaa&words=bbb
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询消息敏感词

更新时间:2024-08-30

获取敏感词列表，支持按敏感词类型查询屏蔽敏感词、替换敏感词或全部敏感词。

请求方法

POST： https://数据中心域名/sensitiveword/list.json

调用频率：每秒钟限 100 次

签名规则：所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
type	String	否	查询敏感词的类型。0 为查询替换敏感词。1 为查询屏蔽敏感词。2 为查询全部敏感词。默认为 1。

请求示例

```
POST /sensitiveword/list.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

type=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
words	Array of objects	敏感词列表。
words[i].word	String	敏感词内容。
words[i].replaceWord	String	替换敏感词的内容，为空时对应 Word 敏感词类型为屏蔽敏感词。

返回值	返回类型	说明
words[i].type	String	敏感词的类型。`0` 为替换敏感词。`1` 为屏蔽敏感词。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"words":[
{
"word":"money",
"replaceWord":"***",
"type":"0"
}
]
}
```

消息回调服务

更新时间:2024-08-30

消息回调服务（原模版路由）支持对 App 中发送的消息进行拦截、内容过滤、内容替换，适用于对接您自己或其他第三方内容审核服务的应用场景。

目前已支持单聊、群组、聊天室、超级群业务。

消息回调服务概述

消息回调服务会根据您在控制台配置的路由规则拦截发送中的消息。被拦截的消息副本会先行发送至您指定的应用服务器（App 后端），而不是直接发送给目标用户。

App 后端可通过对回调请求的响应，实现对消息的以下处理：

- 决定是否需要继续下发消息（如超时未回复，即时通讯服务端默认自动下发消息）
- 决定是否需要继续下发消息扩展（仅针对超级群会话类型，如超时未回复，即时通讯服务端默认自动下发）
- 决定是否需要继续下发消息修改（仅针对超级群会话类型，如超时未回复，即时通讯服务端默认自动下发）
- 使用响应正文中指定字段，直接修改消息内容
- 使用响应正文中指定字段，直接修改消息的推送内容（展示在推送通知栏的内容）
- 使用响应正文中指定字段，直接修改消息扩展 KV。

即时通讯服务端会根据 App 后端返回的响应结果，决定是否将消息下发、是否替换消息中的内容，以及如何进行内容替换。

开通服务

前往控制台的消息回调服务 页面开通服务。生产环境下需要预存费用才能开通。

如果您的网络有 IP 访问限制，请务必配置 IP 白名单，否则无法正常接收服务端回调。

消息回调默认不会对 Server API 接口发送的消息生效。如果需要将 IM Server API 发送的消息按照您配置的路由规则过滤并发送到应用服务器，需要在控制台的免费基础功能 页面启用 **Server API 发送消息过滤敏感词**。

> 提示
>
> - 2021.5.10 号之后开通消息回调服务，默认支持在响应正文中使用 `replaceContent` 字段消息内容替换。2021.5.10 号之前开通的客户需要此功能，必须提交工单 申请。
> - 2022.8.18 号之后开通消息回调服务，默认支持响应正文中使用 `replaceDisablePush`、`replacePushExt`、`replaceExtraContent` 字段对消息内容进行替换。2022.8.18 号之前开通的客户需要此功能，必须提交工单 申请。

创建路由规则

使用消息回调服务需要在控制台创建消息回调服务规则（即路由规则）。

服务开通后，可在消息回调服务 页面创建路由规则。一条路由规则包含以下字段：

- **规则名称**：填写规则名称。
- **会话类型**：选择规则适用的会话类型类型，支持单聊、群聊、聊天室、超级群。
- **消息标识**：填写消息类型的唯一标识，一条规则仅支持填写一个标识。例如内置的文字消息类型的标识是 `RC:TxtMsg`。详见消息类型概述。支持自定义消息类型的标识。
- **发送 ID**：填写发送者的用户 ID，也可以指定规则匹配用户 ID。配置成功后，本条规则仅针对匹配的用户生效。
- **接收 ID**：填写会话 ID，也可以指定规则匹配会话 ID。配置成功后，本条规则仅针对匹配的会话生效。单聊会话 ID 为接收用户 ID。群组、超级群的会话 ID 为群 ID。聊天室会话 ID 为聊天室房间 ID。
- **回调地址**：填写接收回调的地址，请保证公网可访问。

注意事项说明

- **单群聊消息扩展支持消息回调**
  - 在发消息时即携带的扩展数据，可通过回调参数 `extraContent` 获取。
  - 2022.08.11 之后，支持将超级群业务中的消息扩展变更发送到应用服务器。超级群消息进行扩展后，服务端会生成一条类型为 `RC:MsgExMsg` 的消息。新消息仅用于消息路由同步，不会同步给客户端 SDK。新消息 ID 由服务端生成，`originalMsgUID` 字段中会携带原消息的 `messageId`，`content` 字段中为当次变更的扩展数据（结构参见消息扩展功能消息）。如需对消息扩展启用消息回调服务，您需要在控制台添加针对 `RC:MsgExMsg` 类型的路由规则。
- **超级群消息扩展支持消息回调**：
  - 在发消息时即携带的扩展数据，可通过回调参数 `extraContent` 获取。
  - 2022.08.11 之后，支持将超级群业务中对消息扩展的后续变更操作也同步到应用服务器。变更超级群消息扩展后，服务端会生成一条类型为 `RC:MsgExMsg` 的消息。新消息仅用于消息路由同步，不会同步给客户端 SDK。新消息 ID 由服务端生成，`originalMsgUID` 字段中会携带原消息的 `messageId`，`content` 字段中为当次变更的扩展数据（结构参见消

息扩展功能消息）。如需对消息扩展启用消息回调服务，您需要在控制台添加针对 `RC:MsgExMsg` 类型的路由规则。

- **超级群消息内容修改支持消息回调**：超级群消息发送之后支持对其进行修改（不可修改消息类型）。2022.08.11 之后，支持将超级群业务中 App 用户修改的消息内容发送到应用服务器。用户修改消息后，服务端会生成一条新消息，同时保留原消息内容与原消息 ID。新消息仅用于同步，消息类型同原消息相同，因此无需额外创建路由规则。新消息 ID 由服务端生成，不会同步给客户端 SDK。新消息的 `originalMsgUID` 字段中会携带原消息的 `messageId`。

- **媒体消息中的文件需要自行获取**：消息为图片、视频类消息时，如果需要获取图片、视频等文件信息，可通过消息中携带的地址进行下载，文件存储有效期为 6 个月。

- **与全量消息路由服务可以同时使用**：消息回调服务与全量消息路由服务可以同时使用，消息回调中被拒绝下发的消息，默认不执行全量消息路由同步操作。如需修改默认行为，请提交工单 申请开通**含敏感词消息路由功能**。

回调方法

请求方法： POST

数据格式： `application/x-www-form-urlencoded`

即时通讯服务端会在 POST 请求 URL 中添加签名参数，您可通过签名验证调用者身份和数据有效性，详细参见 服务端回调签名。

回调正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，包含以下 HTTP 表单参数：

名称	类型	说明
appKey	String	应用 App Key。
fromUserId	String	发送用户 ID。
targetId	String	目标会话 ID，根据会话类型可能为单聊 ID、群聊 ID、聊天室 ID、超级群 ID。
toUserIds	String	群成员 ID 列表（以英文逗号分隔）。发送群聊定向消息时，接收消息的群成员用户 ID 列表。非群定向消息时为空。
msgType	String	配置好回调的消息类型（包括自定义消息），详见消息类型概述。
content	Object	JSON 结构的消息内容。如果 `msgType` 为内置消息类型，消息内容结构参考用户内容类消息格式或其他内置消息类型的消息内容格式。如果 `msgType` 是您自定义的消息类型，请参考该自定义消息的消息结构。 该字段支持通过回调响应进行修改。
pushContent	String	发送消息时设置的推送通知栏显示内容。离线推送通知中显示该内容。 该字段支持通过回调响应进行修改。
disablePush	Boolean	是否为静默消息，默认为 `false`，设为 `true` 时终端用户离线情况下不会收到通知提醒。 该字段支持通过回调响应进行修改。
pushExt	String	配置消息的推送通知，如推送通知的标题等。详见服务端文档发送消息中对 `pushExt` 的说明。`disablePush` 属性为 true 时此属性无效。 该字段支持通过回调响应进行修改。
expansion	Boolean	是否为可扩展消息，默认为 `false`，设为 `true` 时终端在收到该条消息后，可对该条消息设置扩展信息。 移动端 SDK 4.0.3 版本、Web 端 3.0.7 版本支持此功能

名称	类型	说明
extraContent	Object	消息扩展的内容，JSON 结构的 Key、Value 对，如：`{"type":"3"}`。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符`+ = _` 的组合方式，不支持汉字。Value 最大 4096 个字符。该字段仅在 `expansion` 为 `true` 时生效。 该字段支持通过回调响应进行修改。
channelType	String	会话类型。支持的会话类型包括：`PERSON`（二人会话）、`PERSONS`（讨论组会话）、`GROUP`（群组会话）、`TEMPGROUP`（聊天室会话）、`ULTRAGROUP`（超级群会话）。
msgTimeStamp	String	服务端收到客户端发送消息时的服务器时间（1970年到现在的毫秒数）。
messageId	String	消息唯一标识。
originalMsgUID	String	原始消息 ID，仅针对超级群会话有效。在修改消息、扩展消息、删除消息时，此字段携带有效值。可通过此字段查询原始消息内容。修改超级群消息后，如果需要在服务端撤回消息，必须使用该 ID。
os	String	消息来源，包括：iOS、Android、Websocket、MiniProgram（小程序）、PC、Server。
busChannel	String	会话频道 ID。如果消息为超级群频道中的消息，您可通过 `busChannel` 获取频道 ID。未使用时该内容为空。
clientIp	String	用户当前的 IP 地址及端口，格式：`192.168.6.124:80`。

回调请求示例

例如您在开通服务页面配置的接收地址：`http://example.com/receive_message.php`

```
POST /receive_message.php?
timestamp=1408710653491&nonce=14314&signature=45beb7cc7307889a8e711219a47b7cf6a5b000e8 HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded

appKey=123&content=
{"content":"123"}&fromUserId=fid123&targetId=tid123&msgType=RC:TxtMsg&messageId=596E-P5PG-4FS2-
7OJK&msgTimeStamp=1408710653491&channelType=ULTRAGROUP&os=Server&busChannel=basketball
```

响应回调请求

消息发送到应用服务器后，应用服务器需要返回 HTTP 应答码 200，同时指定 `pass` 属性值。即时通讯服务端将根据 `pass` 状态决定是否下发消息。

响应的 HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

返回参数名称	类型	说明

pass	Int	• **2021.5.10 号前开通服务返回状态说明：**
		1 表示正常下发此条消息；其他表示拒绝。
		
		• **2021.5.10 号后开通服务返回状态说明：**
		  • `0` 表示拒绝，不下发此条消息。
		  • `1` 表示正常下发此条消息，如满足其他消息回调条件继续执行。
		  • `2` 表示正常下发此条消息，如满足其他消息回调条件不继续执行；
		
		注意：2021.5.10 号前开通服务的客户如需使用最新返回状态定义，可提交工单申请。

返回参数名称	类型	说明
replaceContent	String	非必传，JSON 结构，需要替换的消息内容，例如需要替换文本消息内容传入 {\"content\":\"替换后的内容\"} ；针对自定义消息支持标准 JSON 结构的内容替换，例如需要替换自定义消息 msg 中 content 的内容，传入 \"msg\": {\"content\":\"替换后内容\"} 即可，自定义消息最多支持替换到第六层级的结构内容，如不传则不进行替换。
replacePushContent	String	非必传，替换后的 pushContent 字段内容。如果此字段不填或填写了空字符串，则默认使用原值。
replaceDisablePush	Boolean	是否为静默消息，默认为 false，设为 true 时终端用户离线情况下不会收到通知提醒。
replacePushExt	String	非必传，替换后的 pushExt 字段内容。详见服务端文档发送消息中对 pushExt 的说明。如果此字段不填或填写了空字符串，则默认使用原值。
replaceExtraContent	String	非必传，有值则替换 extraContent 的内容。 数据格式如：{\"key1\":{\"v\":\"value1\"},\"key2\":{\"v\":\"value2\"}}。详见服务端文档发送消息中对 extraContent 的说明。 数据集大小限制如下：Key 支持大小写英文字母、数字、特殊字符 + = - _ 的组合方式，不支持汉字，最大 32 个字符，Value 最大 4096 个字符。
extra	String	消息不下发时（即 pass 为 0 时）用于通知消息发送方的通知数据，例如无法下发消息的具体原因。该字段携带的数据将通过客户端的「敏感消息拦截回调」返回发送方，放入被拦截消息详细信息的 extra 字段。目前支持单聊、群聊、 聊天室、超级群会话类型。长度不可超过 1024 个字符。客户端 SDK 版本要求：Web IMLib SDK ≧ 5.3.3，Android IM SDK 版本 ≧ 5.2.5（或 5.2.3.2 及以上的 5.2.3.x 系列的维护版本）。

重要

- 响应正文中是否可使用的 replaceContent、replaceDisablePush、replacePushExt、replaceExtraContent 字段与开通服务日期相关。请仔细阅读开通服务。
- 字段中如数据大小超出限制、数据格式错误会导致消息下发失败。建议 replacePushExt 控制在 1.5 KB 以内。推送相关字段总体长度不可超过 3.8 KB，如超限制可能导致部分第三方厂商通道推送消息失败。推送相关字段具体指 replacePushContent、 replacePushExt，或未经替换字段的原值。
- 如果应答超时 5 秒，服务端会再尝试推送 3 次，如果仍然失败，即时通讯服务默认将下发此条消息。
- 如短时间内有大面积超时，将暂时停止请求您的服务器，90 秒后会继续发送回调请求。异常断网情况下的会延迟 5 分钟同步。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "pass": 1 }
```

关于替换消息内容的提示

替换消息内容对发送端、接收端的本地与服务端的历史消息记录有不同的影响。请谨慎使用此功能。

提示

- 单聊和群聊消息替换后，发送方的本地和服务端历史消息中存储的是替换前的内容，而接收方本地和服务端存储的是经过替换后的内容。
- 聊天室消息替换后，发送方本地为替换前的内容，接收方本地是经过替换后的内容；发送方和接收方服务端存储的都为替换后的内容。
- 超级群消息替换后，发送方本地为替换前的内容，服务端历史消息中存储的是替换后的内容（因此换端登录后拉取历史消息时为替换后的内容），接收方本地和服务端为替换后的内容。

审核结果回调

更新时间:2024-08-30

在控制台的 IM & 音视频审核 页面开通 **IM** 审核服务后，可使用审核结果回调，实时接收审核结果详情。

> 提示
>
> 该回调服务仅返回 IM & 音视频审核 的审核结果。

配置审核结果回调地址

请在控制台的IM & 音视频审核 页面配置回调地址。配置后 30 分钟内生效。请确保公网可访问您提供的回调接收地址。

回调方法

请求方法： POST

数据格式： `application/json`

即时通讯服务端会在 POST 请求头部添加签名参数，您可通过签名验证调用者身份，并确保数据有效性，详见 服务端回调签名。

回调正文参数

该回调服务的 HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

名称	类型	说明
result	Int	返回审核是否通过。`10000` 为审核通过；`10001` 为审核不通过。审核结果详情在 `resultDetail` 字段中返回。
content	String	返回审核的消息内容，详见下方 `content` 结构说明。
msgUID	String	消息 ID。
serviceProvider	String	审核渠道商标识。
resultDetail	String	审核结果详情，直接返回审核服务商的审核结果 JSON 结构响应体，详见下方 `resultDetail` 说明。

- `content` 结构说明：

名称	类型	说明
appKey	String	appkey
fromUserId	String	发送者 ID

名称	类型	说明
targetId	String	接收者 ID
toUserIds	String	群成员 ID 列表（以英文逗号分隔）。发送群聊定向消息时，接收消息的群成员用户 ID 列表。非群定向消息时为空。
conversationType	String	会话类型。支持的会话类型包括：PERSON（二人会话）、GROUP（群组会话）、TEMPGROUP（聊天室会话）、ULTRAGROUP（超级群会话）。
objectName	String	消息类型。一般包括：RC:TxtMsg（文本消息）、RC:ImgMsg（图片消息）、RC:VcMsg（语音消息）、RC:HQVCMsg（高清语音消息）、RC:SightMsg（小视频消息）。
message	String	消息体内容，详细查看消息结构文档。
extraContent	Object	消息扩展的内容，JSON 结构的 Key、Value 对，如：{"type":"3"}。Key 最大 32 个字符，支持大小写英文字母、数字、 特殊字符+ = - _ 的组合方式，不支持汉字。Value 最大 4096 个字符。
clientOs	String	客户端类型，包括：iOS、Android、Websocket、MiniProgram（小程序）、PC、Server（通过 Server API 发送，需要开通 Server API 发送消息进行消息路由功能）。
messageTime	Int	服务端收到客户端发送消息时的服务器时间，精确到毫秒
messageId	String	消息 ID

- resultDetail 说明：直接返回审核服务商的审核结果响应体。
  - 如果您在控制台选择的审核服务商为数美，resultDetail 中为数美审核的返回结果，详细结构说明请参见以下数美文档。注意，文本和图片审核返回的是单条同步请求的响应体，音频和视频审核返回的是单条异步的响应体。数美返回的审核结果响应体中的 riskLabel1 中携带了审核不通过的具体原因。
    - 文本消息审核结果 参见该数美文档中的「返回结果」。文本审核返回的是单条同步请求的响应体。
    - 图片消息审核结果 参见该数美文档中的「同步返回结果」。图片审核返回的是单条同步请求的响应体。
    - 语音消息审核结果 参见该数美文档中的「回调结果」。音频审核返回的是单条异步的响应体。
    - 小视频消息审核结果 参见该数美文档中的「异步回调结果」。视频审核返回的是单条异步的响应体。

回调请求示例

以下示例您在开通服务页面配置的接收地址为 http://example.com/review_result.php

```
POST review_result.php HTTP/1.1
Host: example.com
RC-App-Key: uwd1c0sxdlx2
RC-Timestamp: 1408710653491
RC-Nonce: 14314
RC-Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/json

{
"result":10000,
"content":“{****}”,//审核的消息结构 JSON
"serviceProvider":"ShuMei",
"msgUID":"596E-P5PG-4FS2-7OJK",
"resultDetail":"{****}";//审核结果详细 JSON 结构
}
```

响应回调请求

> 提示
>
> - 只要有 HTTP `200 OK` 成功响应，服务端会认为状态已经同步。
> - 如果应答超时 5 秒，服务端会再尝试推送 3 次，如果仍然失败，即时通讯服务默认将下发此条消息。
> - 如短时间内有大面积超时，将暂时停止请求您的服务器，90 秒后会继续发送回调请求。异常断网情况下的会延迟 5 分钟同步。

免打扰功能概述

更新时间:2024-08-30

即时通讯业务支持多维度、多级别精细化的免打扰设置。

IM 的免打扰设置维度

即时通讯业务支持对免打扰功能进行多维度地控制。开发者可从 App Key、特定的 IM 细分业务（目前仅超级群）、用户级别进行免打扰功能配置。

> 提示
>
> - Android/iOS 客户端 SDK 从 5.2.2 开始提供对以下维度免打扰设置的完整支持。
> - Web 客户端 SDK 从 5.3.0 开始提供对以下维度免打扰设置的完整支持。

免打扰配置的维度	适用场景	说明	服务端 API
App Key 级设置	单聊、群聊、系统会话、超级群	以 App Key 为单位，设置整个应用的默认免打扰级别。默认未设置，等同于全部消息都接收通知。该级别的配置暂未在控制台开放，如有需要，请提交工单。	服务端不提供该 API。
超级群默认设置（全体群成员）	仅限超级群业务	可为指定的超级群的设置默认的免打扰级别，对全体群成员生效。	详见「超级群管理」下的设置群/频道默认免打扰
超级群频道默认设置（全体群成员）	仅限超级群业务	可为指定的超级群频道设置默认的免打扰级别，对全体群成员生效。	详见「超级群管理」下的设置群/频道默认免打扰
会话类型设置（用户级）	单聊、群聊、系统会话、超级群	允许用户设置指定类型会话的免打扰级别。	详见设置会话类型免打扰。
会话的设置（用户级）	单聊、群聊、系统会话、超级群	允许用户设置指定会话的免打扰级别。	详见设置会话免打扰。
超级群会话频道的设置（用户级）	仅限超级群业务	允许用户设置指定超级群频道的免打扰级别。	详见设置会话免打扰。
应用全局设置（用户级）	单聊、群聊、系统会话、超级群	允许用户设置指定时段内全局免打扰级别。	服务端不提供该 API。

免打扰设置的优先级

- 针对单聊、群聊、系统会话，即时通讯服务端会遵照以下顺序搜索免打扰配置。优先级从左至右依次降低，以优先级最高的配置为准判断是否需要触发推送：

  全局免打扰设置（用户级） > 指定会话的免打扰设置（用户级） > 指定会话类型的免打扰设置（用户级） > App 级的免打扰设置

- 针对超级群会话，即时通讯服务端会遵照以下顺序搜索免打扰配置。优先级从左至右依次降低，以优先级最高的配置为准判断是否需要触发推送：

  全局免打扰设置（用户级）> 指定超级群频道的免打扰设置（用户级）> 指定会话的免打扰设置（用户级）> 指定会话类型的免打扰设置（用户级）> 指定超级群频道的默认免打扰设置（超级群全员）> 指定超级群的免打扰设置（超级群全员）> App 级的免打扰设置

设置指定会话免打扰

更新时间:2024-08-30

为指定用户（requestId）设置指定会话（targetId）的免打扰逻辑。支持的会话类型包括：单聊、群聊、超级群、系统会话。设置后 SDK 可实时获取到最新的会话免打扰状态信息。

> 提示
>
> - 当前设置为用户级别设置。
> - 超级群业务还提供针对指定超级群、指定群频道全员生效的默认免打扰配置。详见「超级群管理」下的设置群/频道默认免打扰。

请求方法

POST： https://数据中心域名/conversation/notification/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
conversationType	String	是	会话类型。支持的会话类型包括：1（二人会话）、3（群组会话）、6（系统会话）、10（超级群会话）。
requestId	String	是	设置消息免打扰的用户 ID。
targetId	String	是	目标 ID，根据不同的会话类型（ConversationType），可能是用户 ID、群组 ID、超级群 ID 等。
isMuted	Int	是	消息免打扰设置状态，0 表示关闭，1 表示开启。 该字段已废弃，推荐使用 unpushLevel。isMuted 与 unpushLevel 只需要传一个。如果都传，使用 unpushLevel。
busChannel	String	否	超级群的会话频道 ID。 • 如果传入频道 ID，则针对该指定频道设置消息免打扰级别。 • 注意：2022.09.01 之前开通超级群业务的客户，如果不指定频道 ID，则默认传 "" 空字符串，即仅针对指定超级群会话（targetId）中不属于任何频道的消息设置免打扰状态级别。如需修改请提交工单。

参数	类型	必传	说明
unpushLevel	Int	是	• `-1`：全部消息通知 • `0`：未设置（用户未设置情况下，默认以群 或者 APP级别的默认设置为准，如未设置则全部消息都通知） • `1`：仅针对 @ 消息进行通知 • `2`：仅针对 @ 指定用户进行通知 如：@张三 则张三可以收到推送，@所有人 时不会收到推送。 • `4`：仅针对 @ 群全员进行通知，只接收 @所有人 的推送信息。 • `5`：不接收通知 注意：IMKit 5.2.1 及之前版本不支持 -1、2、4、5，具体表现为 -1、2、4 无法弹出本地通知，5 不生效。推荐 IMKit 用户升级到 5.2.2 及之后版本。

请求示例

```
POST /conversation/notification/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

conversationType=1&requestId=b5NwvIrW8&targetId=UAhIaLkR0&unpushLevel=0
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询指定会话免打扰

更新时间:2024-08-30

查询指定用户（requestId）为指定会话（targetId）的设置的免打扰状态。

> 提示
>
> 该设置为用户级别设置。对应的设置接口详见设置指定会话免打扰。

请求方法

POST： https://数据中心域名/conversation/notification/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
conversationType	String	是	会话类型。支持的会话类型包括：1（二人会话）、3（群组会话）、6（系统会话）、10（超级群会话）。如需查询超级群指定频道的免打扰设置，请传入频道 ID（busChannel）。
requestId	String	是	设置消息免打扰的用户 ID。
targetId	String	是	目标 ID，根据不同的会话类型（ConversationType），可能是用户 ID、群组 ID。
busChannel	String	否	超级群的会话频道 ID。 • 如果传入频道 ID，则查询该频道的消息免打扰级别。 • 注意：如果不指定频道 ID，则查询指定超级群会话（targetId）中不属于任何频道的消息的免打扰状态级别。

请求示例

```
POST /conversation/notification/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

conversationType=2&requestId=b5NwvIrW8&targetId=2MqOJa1Un
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
isMuted	Int	消息免打扰设置状态。 • `-1`：全部消息通知 • `0`：未设置（用户未设置情况下，默认以群 或者 APP级别的默认设置为准，如未设置则全部消息都通知） • `1`：仅针对 @ 消息进行通知 • `2`：仅针对 @ 指定用户进行通知 • `4`：仅针对 @ 群全员进行通知。 • `5`：不接收通知

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"isMuted":0}
```

设置指定会话类型免打扰

更新时间:2024-08-30

为指定用户（requestId）设置针对指定会话类型的免打扰逻辑。支持的会话类型包括：单聊、群聊、超级群、系统会话。设置后 SDK 可实时获取到最新的会话免打扰状态信息。

> 提示
>
> - 当前设置为用户级别设置。用户针对“会话类型”设置的免打扰逻辑优先级次于为“指定会话”设置的免打扰逻辑，详见设置指定会话免打扰。
> - 超级群业务提供针对指定超级群、指定群频道的默认免打扰逻辑配置。非用户级别配置，优先级较低。详见"超级群管理“下的设置群/频道默认免打扰。

请求方法

POST：https://数据中心域名/conversation/type/notification/set.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
conversationType	String	是	会话类型。支持的会话类型包括：1（二人会话）、3（群组会话）、6（系统会话）、10（超级群会话）。
requestId	String	是	设置消息免打扰的用户 ID。

参数	类型	必传	说明
unpushLevel	Int	是	• `-1`：全部消息通知 • `0`：未设置（用户未设置情况下，默认以群或者 APP 级别的默认设置为准，如未设置则全部消息都通知） • `1`：仅针对 @ 消息进行通知 • `2`：仅针对 @ 指定用户进行通知 如：@张三 则张三可以收到推送，@所有人 时不会收到推送。 • `4`：仅针对 @ 群全员进行通知，只接收 @所有人 的推送信息。 • `5`：不接收通知 注意：IMKit 5.2.1 及之前版本不支持按会话类型设置的免打扰逻辑。推荐 IMKit 用户升级到 5.2.2 及之后版本。

请求示例

```
POST /conversation/type/notification/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

conversationType=1&requestId=b5NwvIrW8&unpushLevel=2
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

查询指定会话类型免打扰

更新时间:2024-08-30

查询指定用户（requestId)为指定会话类型设置的免打扰配置。

> 提示
>
> 该设置为用户级别设置，优先级较高。对应的设置接口详见设置指定会话类型免打扰。

请求方法

POST：https://数据中心域名/conversation/type/notification/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
conversationType	String	是	会话类型。支持的会话类型包括：1（二人会话）、3（群组会话）、6（系统会话）、10（超级群会话）。
requestId	String	是	设置消息免打扰的用户 ID。

请求示例

```
POST /conversation/type/notification/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

conversationType=2&requestId=b5NwvIrW8
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
isMuted	Int	• `-1`： 全部消息通知 • `0`： 未设置（用户未设置情况下，默认以群 或者 APP级别的默认设置为准，如未设置则全部消息都通知） • `1`： 仅针对 @ 消息进行通知 • `2`： 仅针对 @ 指定用户进行通知 • `4`： 仅针对 @ 群全员进行通知。 • `5`： 不接收通知

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"isMuted":2}
```

查询用户免打扰时段

更新时间:2024-08-30

查询指定单个用户免打扰时段设置。

方法说明

POST： https://数据中心域名/user/blockPushPeriod/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户ID

请求示例

```
POST /user/blockPushPeriod/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: awd1c0sxdlx1
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7bc7307889a8e711219a47b7cf6a5b000e9
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=userId
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码， 200 为处理成功
data	Object	返回值。
data.startTime	String	开始时间，精确到秒。格式为 HH:MM:SS，示例：22:00:00。注意：startTime 与应用所属数据中心有关。如您的 App 业务使用国内数据中心，该时间为北京时间。如您的 App 业务使用海外数据中心，该时间为 UTC 时间。

返回值	返回类型	说明
data.period	Number	免打扰时间窗口大小，单位为分钟。范围为 [0-1439] 的整数。`0` 表示未设置。
data.unPushLevel	Number	免打扰级别。`1`：仅 @消息进行通知，普通消息不进行通知。`0`：所有消息都进行通知。`5`：所有消息都不进行通知。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200,"data":{"startTime":"23:59:59" ,"period":120, "unPushLevel":1 }}
```

设置用户免打扰时段

更新时间:2024-08-30

设置用户免打扰时段与免打扰级别。

- 该接口会设置一个免打扰时间窗口。在再次设置或删除用户免打扰时间段之前，当次设置的免打扰时间窗口会每日重复生效。例如，App 用户希望设置永久全天免打扰，可设置 `startTime` 为 `00:00:00`，`period` 为 `1439`。
- 单个用户仅支持设置一个时间段，重复设置会覆盖该用户之前设置的时间窗口。

方法说明

POST： https://数据中心域名/user/blockPushPeriod/set.json

调用频率：每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID
startTime	String	是	开始时间，精确到秒。格式为 `HH:MM:SS`，示例：22:00:00。注意：`startTime` 与应用所属数据中心有关。如您的 App 业务使用国内数据中心，请使用北京时间。如您的 App 业务使用海外数据中心，请使用 UTC 时间。
period	Number	是	免打扰时间窗口大小，单位为分钟。支持范围为 [0-1439] 的整数。`0` 表示未设置。
level	Number	否	免打扰级别。`1`：仅 @消息进行通知，普通消息不进行通知。`0`：所有消息都进行通知。`5`：所有消息都不进行通知。 默认 `1`。

请求示例

```
POST /user/blockPushPeriod/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0soslx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7as7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=userId&startTime=23:59:59&period=120&level=1
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

删除用户免打扰时段

更新时间:2024-08-30

删除指定单个用户的免打扰时段设置。

方法说明

POST： https://数据中心域名/user/blockPushPeriod/delete.json

调用频率：每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID

请求示例

```
POST /user/blockPushPeriod/delete.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdsdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7487889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX(最大长度36)

userId=userId
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	int	返回码， 200 为处理成功

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Request-ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXX

{"code":200}
```

统计推送数据

更新时间:2024-08-30

即时通讯服务提供推送统计数据功能，您可以在控制台推送成功率统计 页面查看推送服务的统计数据。

> 提示
>
> 推送数据统计功能仅针对已上线应用的生产环境。

推送数据统计包括：

- **推送总量**：表示实际需要推送的通知条数，包括第三方厂商推送和即时通讯自建推送服务（RongPush）。
- **推送成功总量**：表示成功推送到第三方厂商推送服务、或即时通讯自建推送服务（RongPush）的通知数，推送成功并不代表实际已推送到了目标设备上。
- **推送失败总量**：表示推送到第三方厂商推送服务、或即时通讯自建推送服务（RongPush）失败的通知数。
- **推送到达总量（需配置）**：表示手机设备实际已经收到的通知数，部分系统手机需要终端主动上报达到数据。
- **推送点击总量（需配置）**：表示终端用户点击通知总数，部分系统手机需要终端主动上报点击数据。

因第三方设计差异，即时通讯服务无法直接获取部分第三方推送厂商的推送到达和推送点击数据，需要您在第三方推送平台进行配置或者客户端自行上报。

采集推送到达数据

推送到达是指通知已发送到第三方厂商推送通道，或即时通讯自建推送服务（RongPush）后，推送通知成功下发到目标设备。

华为、魅族推送到达数据依赖手机厂商推送通道提供的「送达回执」服务。Google FCM 仅支持采集透传消息方式的推送到达数据。Apple APNs 必须在客户端手动上报。

详细支持情况参见下表：

推送平台	提供推送到达数据	是否需要配置
华为	厂商支持	需要，参见配置华为推送回执
魅族	厂商支持	需要，参见配置魅族推送回执
小米	厂商支持	无需配置
Vivo	厂商支持	无需配置
OPPO（包括 Realme、OnePlus）	厂商支持	无需配置
Google FCM	厂商不支持，由客户端 SDK 提供	详见采集 FCM 推送到达数据
Apple APNs	厂商不支持，由客户端 SDK 提供（要求 SDK ≧ 5.1.4）	需要，需要 SDK 主动上报到达事件。参见上报推送数据

采集推送点击数据

推送点击数据是指表示终端用户点击通知的总数。

即时通讯服务可直接获取大部分厂商的推送通道的点击数据。如果即时通讯客户端 SDK 版本低于 5.2.3，华为推送通道下发的推送通知点击数据需要您在客户端手动上报。Apple APNs 必须在客户端手动上报。

详细支持情况参见下表：

推送平台	推送点击事件上报	是否需要配置
RongPush	支持	不需要，客户端 SDK 默认实现上报逻辑
小米	支持	不需要，客户端 SDK 默认实现上报逻辑
华为	支持（要求 SDK ≧ 5.1.4）	从 IMLib 5.2.3 版本开始，客户端 SDK 默认实现上报逻辑（低于该版本则需要调用客户端 SDK 手动上报点击事件）
Vivo	支持	不需要，客户端 SDK 默认实现上报逻辑
魅族	支持	不需要，客户端 SDK 默认实现上报逻辑
OPPO	支持	不需要，客户端 SDK 默认实现上报逻辑
Google FCM	支持	不需要，客户端 SDK 默认实现上报逻辑
Apple APNs	支持（要求 SDK ≧ 5.1.4）	需要，需要 SDK 主动上报点击事件。参见上报推送数据

获取推送日志

更新时间:2024-08-30

即时通讯服务端可以提供 App Key 下的推送历史记录。推送历史记录以日志文件方式提供，并已经过压缩。您可以使用服务端 API 获取指定 App 的推送日志。

服务端 API 接口仅返回推送日志文件下载地址，获取地址后请您自行下载。

开通服务

如有需求，请提交工单，申请开通推送日志下载服务。

计费规则：「推送日志下载」为增值服务。IM 尊享版可免费使用该服务，其他 IM 套餐版本按服务端生成的推送日志数量档位计费。每千万条为一个档位，国内 150 元/千万条，国外 300 元/千万条。月度推送日志条数若不满足 1 千万条，按最低档收费。具体以计费说明文档为准。

推送日志说明

> 提示
>
> 如果是海外数据中心的应用，请根据您的 SDK 版本完成必要配置，确保推送相关统计数据可上报到正确的数据中心，详见知识库文档海外数据中心使用指南。

推送日志已支持统计单聊、群聊、系统会话的离线消息推送、以及服务端 `/push.json` 和 `/push/user.json` 接口发出的推送。超级群离线推送日志记录暂不完善，暂不建议使用。

推送到达日志采集说明

- 对华为、魅族设备的推送到达率的数据采集，依赖华为和魅族的「送达回执」功能。您需要在厂商的平台上完成相应配置。开发者文档中已提供的简要说明：
  - 参见配置华为推送回执
  - 参见配置魅族推送回执
- 对 Google FCM 推送的到达率统计仅支持**透传消息方式**，暂不支持**通知消息方式**，并且日志数据采集依赖应用程序主动上报。详见 Android 客户端 SDK 文档 采集 FCM 推送到达数据。
- 对 Apple APNs 推送的到达率统计依赖应用程序主动上报，详见 iOS 客户端 SDK 文档 上报推送数据。

推送点击日志采集说明

- 对华为推送的点击率统计数据采集可能依赖应用程序主动上报，具体与客户端 SDK 有关：
  - SDK < 5.1.4，不支持统计推送点击数据。
  - 5.1.4 ≦ SDK < 5.2.3，依赖应用程序主动上报。详见 Android 客户端 SDK 文档 采集华为推送点击数据。

- SDK ≧ 5.2.3，SDK 已实现自动上报，无需应用程序处理。

- 对 Apple APNs 推送的点击率统计数据采集依赖应用程序主动上报，详见 iOS 客户端 SDK 文档 上报推送数据 。

数据时效性

获取数据有一定延迟（T+0 模式）。

- 首先，每自然小时内的数据文件需要在该时段结束后才会生成。例如，12 至 13 点之间日志数据在 13 点以后才能生成。
- 其次，因数据统计、文件压缩打包等耗时，一般需要再等待 3 个小时，才可提供下载地址。例如，12 至 13 点之间日志数据在 16 点后可获取到下载地址。
- 另外，开通服务后，服务端即可保存当前自然小时内的推送日志数据。例如您在 12:00 至 13:00 之间任意时间开通服务，则即时通讯服务端可提供从 12:00 开始的推送日志。

即时通讯服务端以自然小时为节点，打包推送日志数据，因此推送日志数据包仅支持按小时获取。

日志保存时限

服务端保存的推送日志文件仅保留 7 天。7 天后服务端自动删除该日志文件。

日志格式

名称	类型	说明
messageId	String	推送消息 ID。
receiverId	String	接收者用户 ID。
systemType	String	接收者的设备操作系统类型：iOS、Android
pushStatus	String	推送状态，可能为成功（`0`）、失败（`1`）、到达（`2`）、已被点击（`3`）。
sendTime	String	消息发送时间，精确到毫秒，海外为 UTC 时间。
time	String	推送状态变更的时间戳，根据 `pushStatus` 取值，该字段可能为推送成功、失败、到达、点击的时间。精确到毫秒，海外为 UTC 时间
pushType	String	推送类型。可能为 `MI`、`HW`、`HONOR`、`MEIZU`、`OPPO`、`VIVO`、`FCM`、`APNS`、`RONG`（即时通讯自建推送）。
pushChannelId	String	渠道 ID，只有 `pushStatus` 推送状态为成功（`0`）时有效，目前仅支持小米、华为、OPPO。
pId	String	推送关联 ID，仅支持 FCM、APNS，可用于关联推送成功、到达、点击记录，追踪 FCM 和 APNs 的推送状态变化。该字段要求客户端 SDK 版本 ≧ 5.1.7。
requestId	String	推送请求 ID，是即时通讯服务端调用第三方推送 API 时的请求唯一标识。如需追踪小米、魅族、华为、VIVO、OPPO 推送的状态变化，可与 `pushToken` 一起使用，关联推送成功、到达、点击记录，
pushToken	String	推送 Token。追踪小米、魅族、华为、VIVO、OPPO 推送的状态变化，可与 `requestId` 一起使用，关联推送成功、到达、点击记录。
deviceId	String	推送的目标设备 ID。如需追踪华为、VIVO、OPPO 的推送的状态，可与 `messageId` 一起使用，关联推送成功、点击记录。
senderId	String	消息发送者 ID。如果直接发送推送（非由消息触发），则没有消息发送者 ID。
channelType	String	消息所属的会话类型，可能为单聊会话（`1`）、群组会话（`3`）、系统通知（`6`）、超级群会话（`10`）。
channelId	String	此字段已废弃。 目前此字段仅用于兼容旧版本，不建议使用。请使用 `targetId`作为替代。

名称	类型	说明
targetId	String	会话 ID。根据 `channelType` 取值不同，可能为群聊 ID、超级群 ID。如果 `channelType` 为系统会话，单聊会话，因会话 ID 即接收者的用户 ID（`receiverId`），此字段为空。
groupChannelId	String	超级群群场景中的频道 ID，在非超级群场景中此字段为空。
thirdRespBody	String	（第三方平台返回字段）第三方推送平台的推送状态回调，依赖各厂商提供的推送回执服务。受第三方平台自身限制，回调数据有差异：小米、魅族的回调包含推送到达、点击数据。华为、VIVO、OPPO 包含推送到达数据。在华为推送回调数据中，如果是由即时通讯服务触发的推送，`bitag` 参数值会带有 `rc` 前缀，结构为 `rc_{appkey}_{messageid}_0`，例如：`rc_fafweefwf_596E-P5PG-4FS2-7OJK_0`。
pushErrorCode	String	即时通讯服务推送到第三方厂商时失败错误码。
pushErrorMsg	String	对应 pushErrorCode 的推送失败描述。
thirdApiRespInfo	String	（第三方平台返回字段）第三方推送 API 返回 Response，请参考表格下方第三方推送平台文档。
thirdApiHttpStatusCode	String	（第三方平台返回字段）第三方推送 API 返回的 HTTP Status Code 码，请参考表格下方第三方推送平台文档。
thirdApiBusCode	String	（第三方平台返回字段）第三方推送 API 返回的业务 Code 码，请参考表格下方第三方推送平台文档。

第三方推送平台官方文档：

- 华为推送服务端 API 文档、华为推送下行消息回执
- 小米推送服务端 API 文档、小米推送消息回执、小米推送错误码
- 魅族推送服务端 API 文档
- VIVO 推送服务端 API 文档、VIVO 推送消息回执
- OPPO 推送服务端 API 文档（广播推送）、OPPO 推送服务端 API 文档（单点推送）、OPPO 推送消息回执

请求方法

POST：https://数据中心域名/message/push/detail/history.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必填	说明
date	String	是	指定时间，格式为 `YYYYMMDDHH`，用于获取指定整小时的推送日志数据。例如，使用国内数据中心的应用，如需获取 2023 年 1 月 1 日凌晨 1 点至 2 点的数据，传入 `2023010101`。注意：如您的 App 业务使用新加坡数据中心，`date` 的值使用北京时间。

请求示例

```
POST /message/push/detail/history.json HTTP/1.1
Host: api-cn.ronghub.com
App-Key: uwd1c0sxdlx2
Timestamp: 1408710653491
Nonce: 14314
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

date=2023010101
```

返回参数

返回值	返回类型	说明
code	Int	返回码，200 为正常。
url	String	历史记录下载地址。如果暂无消息记录数据，则 `url` 值为空。
date	String	历史记录时间。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"url":"http://xx.com/1/c6720eea-452b-4f93-8159-7af3046611f1.gz",
"date":"2023010101"
}
```

推送 Plus 概述

更新时间:2024-08-30

推送 Plus 是为高频、并发推送需求提供的增值推送服务。支持推送文本、富媒体、自定义消息等，适用于社交互动、交易状态同步、系统升级、帐号唤醒拉活、活动通知等推送运营服务场景，并支持推送数据统计。

推送 Plus 独享推送通道，具备更高的并发处理能力，可毫秒级触达目标用户。推送 Plus 数据统计提供推送成功、失败、到达、点击数据统计能力。开发者可通过 API 或控制台获取相关数据。

推送 Plus 需要付费开通后才能使用。欢迎访问官网推送 Plus 产品介绍页，或直接前往控制台开通推送 Plus。

> 该功能开发环境下可免费使用。生产环境下，需要在控制台开通推送 Plus 服务后才能使用。

注意事项

- 手机厂商推送通道是从系统层维护的长连接通道，比即时通讯服务自建推送通道 RongPush 更省电，到达率更高。希望使用推送 Plus 的客户，应尽量集成第三方厂商推送通道。
- 华为推送通道上报推送点击数据，要求客户端使用 5.1.4 或更高版本的 SDK。从 5.2.3 版本开始无需客户端手动上报。
- Google FCM 支持统计推送点击数据和 透传消息方式 的推送到达数据，5.3.0 版本前的客户端需手动上报推送到达数据。
- APNs 推送通道无法直接获取推送到达、点击数据。iOS 客户端需要主动上报推送到达数据与推送点击数据，且必须使用 5.1.4 或更高版本的 SDK。
- 如需按用户标签推送，请确保已通过设置用户标签、批量设置用户标签 接口给 App Key 下用户设置了标签。

基本流程

1. 在控制台开通推送 Plus。
2. 因第三方推送服务设计差异，客户端 SDK 无法直接获取全部推送平台的推送到达、点击通知数据。需要您前往厂商推送平台进行设置或通过客户端 SDK 主动上报：
   - 详见 Android 端 IM SDK 5.X 文档上报推送数据
   - 详见 iOS 端 IM SDK 5.X 文档上报推送数据
3. 使用推送 Plus 专用推送接口，发送全量用户不落地通知。
4. 使用控制台与服务端 API 获取推送成功、到达、点击数据。

推送 Plus 专用推送接口

推送 Plus 服务支持高并发发送不落地通知。开通推送 Plus 后，请使用专用的服务端 API 接口发送全量用户不落地通知。

> 提示

从即时通讯服务端发送不落地通知时，无论用户是否正在使用 App，都会向用户发送通知。通知仅展示在通知栏。区别于落地通知，不落地通知不携带聊天会话消息，即用户登录 App 后不会在聊天页面看到该内容，不会存储到本地数据库。
即时通讯服务将 30 天内连接过 IM 服务的设备作为推送的目标。30 天内未打开过应用的设备，被视为应用已被卸载。

请求方法

POST ：https://数据中心域名/push/custom.json

频率限制： 每天（自然日）限发送 100 次，每小时最多发送 20 次。如需要调整发送频率，可联系销售咨询，电话 13161856839。

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/json`，包含具有以下结构的 JSON 对象：

提示

`audience` 字段使用须知：

- 如果 `audience` 中指定了 `is_to_all` 为 `true`，则忽略其他推送条件（`tag`、`tag_or`、`packageName`）。
- `audience` 中 `tag`、`tag_or` 共存时，两个数组之间为逻辑与（AND）关系。
- `audience` 中包含 `packageName` 时，`packageName` 与 `tag` 或 `tag_or` 为逻辑与（AND）关系。例如三者共存时关系为 `tag` AND `tag_or` AND `packageName`。
- `audience` 中包含 `tagItems` 且 `tagItems` 包含有效数据时，服务端不会使用 `tag`、`tag_or` 字段传入的值。

参数	类型	必传	说明
platform	String[]	是	目标平台（操作系统），可以为 `ios`、`android` 其中一个或全部。全部填写时给给 Android、iOS 两个平台推送消息。
audience	Object	是	推送条件。支持按用户 ID 推送，按用户标签推送（`tag`、`tag_or`、`tagItems`）、按应用包名推送（`packageName`）和按指定平台全部推送（`is_to_all`）。**注意**：如果推送条件中 `is_to_all` 为 `true`，则忽略其他推送条件。
audience.userid	String[]	否	用户 ID 数组。如果 `userid` 有值，则 `platform`、`tag`、`tagItems` 均无效。如果 `is_to_all` 为 `true`，则当前参数无效。
audience.tag	String[]	否	用户标签数组，标签之间为 AND 关系。数组中最多包含 20 个标签。
audience.tag_or	String[]	否	用户标签数组，标签之间为 OR 关系。数组中最多包含 20 个标签。
audience.packageName	String	否	应用包名。与 `tag` 或 `tag_or` 为逻辑与（AND）关系。

参数	类型	必传	说明
audience.is_to_all	Boolean	是	是否按指定平台（操作系统）全部推送。`true` 表示全部推送，此时其他推送条件字段均无效。`false` 表示按其他推送条件进行推送。
audience.tagItems	Array of Objects	是	在按用户标签推送场景下，可通过 tagItems 实现复杂与或非逻辑。在 `tagItems` 包含有效内容的情况下，`tag`、`tag_or` 字段无效。详见下方 **tagItems 结构说明**。
notification	Object	是	按平台（操作系统）指定推送内容。
notification.title	String	否	通知栏显示标题，最长不超过 50 个字符。
notification.alert	String	否	默认推送通知内容。
notification.ios	Object	否	设置 iOS 平台下的推送及附加信息。详见下方 `notification.ios` **结构说明**。
notification.android	Object	否	设置 Android 平台下的推送及附加信息。详见下方 `notification.android` **结构说明**。

- `audience.tagItems` **结构说明**

参数	类型	必传	说明
tags	Array of strings	是	用户标签数组。
isNot	Boolean	是	是否对 tags 数组的运算结果进行非运算。默认为 `false`。
tagsOperator	String	是	tags 数组内标签之间的运算符。
itemsOperator	String	是	tagItems 数组内当前 Object 与上一个 Object 之间的运算符。**注意**：首个 Object 内的 `itemsOperator` 未被使用，为无效字段。

tagItems 运算优先级

`tagItems` 节点之间由 `itemsOperator` 控制关系。`tagItems` 各个节点之间存在计算优先级关系。例如，`tagItems` 内存在四个节点，节点的 `itemsOperator` 分别为 and、or、and、or。第一个运算符 and 无效可忽略。那么这四个节点的计算逻辑为：(((1 or 2) and 3) or 4)。

- `notification.ios` **结构说明**

参数	类型	必传	说明
title	String	否	通知栏显示的推送标题，仅针对 iOS 平台，支持 iOS 8.2 及以上版本，参数在 `ios` 节点下设置，详细可参考“设置 iOS 推送标题请求示例”，此属性优先级高于 notification 下的 title。
contentAvailable	Int	否	针对 iOS 平台，静默推送是 iOS7 之后推出的一种推送方式。 允许应用在收到通知后在后台运行一段代码，且能够马上执行。详情请查看知识库文档。1 表示为开启，0 表示为关闭，默认为 0

参数	类型	必传	说明
badge	int	否	应用角标，仅针对 iOS 平台；不填时，表示不改变角标数；为 0 或负数时，表示 App 角标上的数字清零；否则传相应数字表示把角标数改为指定的数字，最大不超过 9999，参数在 `ios` 节点下设置，详细可参考“设置 iOS 角标数 HTTP 请求示例”。
thread-id	String	否	iOS 平台通知栏分组 ID，相同的 thread-id 推送分一组，单组超过 5 条推送会折叠展示
apns-collapse-id	String	否	iOS 平台，从 iOS10 开始支持，设置后设备收到有相同 ID 的消息，会合并成一条
category	String	否	iOS 富文本推送的类型开发者自己定义，自己在 App 端进行解析判断，与 richMediaUri 一起使用，当设置 category 后，推送时默认携带 mutable-content 进行推送，属性值为 1。
richMediaUri	String	否	iOS 富文本推送内容的 URL，与 category 一起使用。
interruption-level	String	否	适用于 iOS 15 及之后的系统。取值为 `passive`，`active`（默认），`time-sensitive`，或 `critical`，取值说明详见对应的 APNs 的 interruption-level 字段。在 iOS 15 及以上版本中，系统的“定时推送摘要”、“专注模式”都可能导致重要的推送通知（例如余额变化）无法及时被用户感知的情况，可考虑设置该字段。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

- `notification.android` 结构说明

参数	类型	必传	说明
honor.importance	String	否	荣耀通知栏消息优先级，取值： • NORMAL（服务与通讯类消息） • LOW（咨询营销类消息）。若资讯营销类消息发送时带图片，图片不会展示。
honor.image	String	否	荣耀推送自定义通知栏消息右侧的大图标 URL，若不设置，则不展示通知栏右侧图标。 • URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。 • 图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp。 超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.channelId	String	否	华为推送通知渠道的 ID。详见自定义通知渠道。

参数	类型	必传	说明
hw.importance	String	否	华为推送通知栏消息优先级，取值 NORMAL、LOW，默认为 NORMAL 重要消息。
hw.image	String	否	华为推送自定义的通知栏消息右侧大图标 URL，如果不设置，则不展示通知栏右侧图标。URL 使用的协议必须是 HTTPS 协议，取值样例：`https://example.com/image.png`。图标文件须小于 512KB，图标建议规格大小：40dp x 40dp，弧角大小为 8dp，超出建议规格大小的图标会存在图片压缩或显示不全的情况。
hw.category	String	否	华为推送通道的消息自分类标识，category 取值必须为大写字母，例如 `IM`。App 根据华为要求完成自分类权益申请 或 申请特殊权限 后可传入该字段有效。详见华为推送官方文档消息分类标准。该字段优先级高于控制台为 App Key 下的应用标识配置的华为推送 Category。
mi.channelId	String	否	小米推送通知渠道的 ID。详见小米推送消息分类新规。
mi.large_icon_uri	String	否	（由于小米官方已停止支持该能力，该字段已失效）消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。国内版仅 MIUI12 以上版本支持，以下版本均不支持；国际版支持。图片要求：大小120 * 120px，格式为 png 或者 jpg 格式。
oppo.channelId	String	否	oppo 推送通知渠道的 ID。详见推送私信通道申请。
vivo.classification	String	否	VIVO 推送服务的消息类别。可选值 `0`（运营消息，默认值） 和 `1`（系统消息）。该参数对应 VIVO 推送服务的 `classification` 字段，见 VIVO 推送消息分类说明。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送通道类型。
vivo.category	String	否	VIVO 推送服务的消息二级分类。例如 `IM`（即时消息）。该参数对应 VIVO 推送服务的 `category` 字段。详细的 category 取值请参见 VIVO 推送消息分类说明。如果指定 `category` ，必须同时传入与当前二级分类匹配的 `classification` 字段的值（系统消息场景或运营消息场景）。请注意遵照 VIVO 官方要求，确保二级分类（`category`）取值属于 VIVO 系统消息场景或运营消息场景下允许发送的内容。该字段优先级高于控制台为 App Key 下的应用标识配置的 VIVO 推送 Category。
fcm.channelId	String	否	Google FCM 推送通知渠道的 ID。应用程序必须先创建一个具有此频道 ID 的频道，然后才能收到具有此频道 ID 的任何通知。更多信息请参见 Android 官方文档。
fcm.collapse_key	String	否	Google FCM 推送中可以折叠的一组消息的标识符，以便在可以恢复传递时仅发送最后一条消息。

参数	类型	必传	说明
fcm.imageUrl	String	否	Google FCM 推送自定义的通知栏消息右侧图标 URL，如果不设置，则不展示通知栏右侧图标。 • 图片的大小上限为 1MB。 • 要求控制台 FCM 推送配置为证书与通知消息方式。
extras	JSONObject	否	附加信息，如果开发者自己需要，可以自己在 App 端进行解析。

请求示例

- 示例 1：按用户标签推送（同时使用 `audience.tag` 与 `audience.tag_or`）

```
POST /push/custom.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{
"platform":["ios","android"],
"audience":{
"tag":["女","年轻"],
"tag_or":["北京","上海"],
"is_to_all":false
},
"notification":{
"title":"标题",
"alert":"this is a push",
"ios":
{
"thread-id":"223",
"apns-collapse-id":"111",
"extras": {"id": "1","name": "2"}
},
"android": {
"hw":{
"channelId":"NotificationKanong",
"importance": "NORMAL",
"image":"https://example.com/image.png"
},
"mi":{
"channelId":"rongcloud_kanong",
"large_icon_uri":"https://example.com/image.png"
},
"oppo":{
"channelId":"rc_notification_id"
},
"vivo":{
"classification":"0"
},
"extras": {"id": "1","name": "2"}
}
}
}
```

- 示例 2：按用户标签推送（使用 audience.tagItems）

```
POST /push/custom.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1585127132438
Signature: 890b422b75c1c5cb706e4f7921df1d94e69c17f4
Content-Type: application/json

{
"platform":["ios","android"],
"audience":{
"tagItems":[
```

```
{
"tags":[
"guangdong",
"hunan"
],
"isNot":false,
"tagsOperator":"OR",
"itemsOperator":"OR"
},
{
"tags":[
"20200408"
],
"isNot":true,
"tagsOperator":"OR",
"itemsOperator":"AND"
},
{
"tags":[
"male",
"female"
],
"isNot":false,
"tagsOperator":"OR",
"itemsOperator":"OR"
}
],
"userid":[
"123",
"456"
],
"is_to_all":false
}
"notification":{
"title":"标题",
"alert":"this is a push",
"ios":
{
"thread-id":"223",
"apns-collapse-id":"111",
"extras": {"id": "1","name": "2"}
},
"android": {
"hw":{
"channelId":"NotificationKanong",
"importance": "NORMAL",
"image":"https://example.com/image.png"
},
"mi":{
"channelId":"rongcloud_kanong",
"large_icon_uri":"https://example.com/image.png"
},
"oppo":{
"channelId":"rc_notification_id"
},
"vivo":{
"classification":"0"
},
"extras": {"id": "1","name": "2"}
}
}
}
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，200 为正常。
id	String	推送唯一标识。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"id":"BRBL-IQ6N-80CO-D5U5"}
```

获取推送聚合统计数据

> 提示
>
> 生产环境下，需要在控制台开通推送 Plus 版本后才能使用。

查询某天应用全量推送数据统计，最多可查询 30 天内的推送数据，当天数据次日支持查询。

推送聚合统计数据也可以在控制台-数据统计中查看。

请求方法

GET：http://api.developer.rongcloud.cn/stat/getDayPushData?date={date}

签名规则：所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 1 次

路径参数

参数	类型	必传	说明
date	String	否	查询日期，非必填，默认查询昨天的推送统计数据，最多支持查询 30 天内的数据情况，如：20210811

请求示例

每次请求 API 接口时，均需要提供 4 个 HTTP Request Header。详见 API 请求签名。

```
GET http://api.developer.rongcloud.cn/stat/getDayPushData?date=20210802 HTTP/1.1
App-Key: e0x9wycfev2lq
Nonce: 66583
Timestamp: 1629275060000
Signature: af590f4963f24e2d862b5f82d2170320e39c5477
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，2000 为正常。
data	String	按推送平台，返回的推送统计数据，支持平台包括：IM、HW、OPPO、VIVO、MEIZU、FCM、APNs、RongPush（即时通讯自建推送）
allcnt	Int	单日推送总量
succcnt	Int	单日推送成功总量
arrcnt	Int	单日推送成功到达设备总量，华为、魅族平台的推送到达数，需要到厂商平台设置后，才能支持。FCM 暂不支持推送到达数据统计
opencnt	Int	单日推送通知栏点击总量，受厂商平台限制 OPPO 平台不支持推送点击数据统计。FCM 暂不支持推送点击数据统计
failcnt	Int	单日推送失败总量

返回结果示例

```
{
"code": 2000,
"msg": "success",
"data": {
"MI": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"HW": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"OPPO": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 0
},
"VIVO": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"MEIZU": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"FCM": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 0,
"opencnt": 0,
"failcnt": 241
},
"RongPush": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"APNs": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
}
}
}
```

获取单次推送统计数据

> 提示
>
> 生产环境下，需要在控制台开通推送 Plus 版本后才能使用。
>
> 针对每一次推送，可统计到推送 3 天内的推送到达、点击数据变化情况，如 1 号推送一条消息，2、3 号仍然可以查询到 1 号推送的数据，当日推送数据次日支持查询。

根据推送 ID 查询不落地推送统计，支持通过 /push/custom.json 接口的推送统计查询。

单次推送统计数据也可以在控制台-数据统计中查看。

请求方法

GET：http://api.developer.rongcloud.cn/stat/getPushIdData?push_id={push_id}

签名规则：所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 1 次

路径参数

参数	类型	必传	说明
push_id	String	是	推送唯一标识，调用 /push/custom.json 接口后返回

请求示例

每次请求 API 接口时，均需要提供 4 个 HTTP Request Header。详见 API 请求签名。

```
GET http://api.developer.rongcloud.cn/stat/getPushIdData?push_id=BR17-PG4T-G3OO-D5U5 HTTP/1.1
App-Key: e0x9wycfev2lq
Nonce: 49166
Timestamp: 1629275088000
Signature: 566e194c1cc2e1ac7d2b30c8b531bc54baf26d6e
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Int	返回码，2000 为正常。
data	String	按推送平台，返回的推送统计数据，平台包括：IM、HW、OPPO、VIVO、MEIZU、FCM、APNs、RongPush（即时通讯自建推送）

返回值	返回类型	说明
allcnt	Int	单日推送总量
succcnt	Int	单日推送成功总量
arrcnt	Int	单日推送成功到达设备总量，华为、魅族平台的推送到达数，需要到厂商平台设置后，才能支持。FCM 暂不支持推送到达数据统计
opencnt	Int	单日推送通知栏点击总量，受厂商平台限制 OPPO 平台不支持推送点击数据统计。FCM 暂不支持推送点击数据统计
failcnt	Int	单日推送失败总量

返回结果示例

```
{
"code": 2000,
"msg": "success",
"data": {
"MI": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"HW": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"OPPO": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 0
},
"VIVO": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"MEIZU": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"FCM": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 0,
"opencnt": 0,
"failcnt": 241
},
"RongPush": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
},
"APNs": {
"allcnt": 1400,
"succcnt": 840,
"arrcnt": 740,
"opencnt": 140,
"failcnt": 241
}
}
}
```

按应用版本定义推送内容

即时通讯服务端支持按客户端 SDK 上报的 App 版本定义消息推送通知中的内容。开通推送 **Plus**后可使用该功能。

> 提示
>
> - 您需要调用客户端 SDK 接口主动上报 App 版本。要求 Android / iOS 客户端 SDK 版本大于等于 5.2.2。
> - 当前仅支持根据 App 版本订制自定义消息类型的推送通知，不支持即时通讯服务预定义的消息类型。

通过 SDK 将 App 版本信息上报至即时通讯服务后，您可以订制不同版本的 App 所接收的消息推送通知的内容（pushContent）。例如，给低版本 App 的用户订制消息推送通知时，可在推送通知中提示“需要升级 App 才能解析新消息类型”。该功能也可用于按 App 版本、平台、消息类型（仅支持自定义消息类型）订制个性化的推送通知内容。

创建规则

您需要在控制台按版本号推送消息页面创建按 App 版本号推送规则。一条规则仅适用于一个客户端操作系统（Android/iOS），仅包含一个自定义消息类型的标识，可包含多个 App 的版本号。

规则具体包含以下字段：

- **规则名称**：填写规则名称。
- **消息标识**：填写消息类型的唯一标识，一条规则仅支持填写一个标识。仅支持自定义消息类型的标识。不支持内置消息类型（消息类型概述）。
- **Push 消息内容**：自定义的推送内容（pushContent）。
- **平台**：Android 或 iOS。

- **涉及版本号**：填写由客户端 SDK 上报的 App 版本号。

开启、关闭服务在设置完成后30分钟后生效。

上报 App 版本号

该功能依赖由客户端手动上报的应用程序版本号（要求 **SDK 版本 ≧ 5.2.2**）。在 SDK 完成初始化之后，您需要调用 SDK 接口主动上报 App 版本。即时通讯服务端获取规则与客户端 App 版本号后，一旦有符合规则的离线消息触发推送，服务端将替换远程推送通知中的内容为规则中指定的 **Push 消息内容**。

- Android：

```
RongIMClient.getInstance().setAppVer(appVersion);
```

- iOS

```
[[RCIMClient sharedRCIMClient] setAppVer:app_Version];
```

注意 App 版本号不可为空，长度小于 20。例如 `1.1.0`。

自定义多语言推送模板

更新时间:2024-08-30

即时通讯服务支持通过推送模板功能实现多语言推送。服务端会根据 App 用户通过客户端上报的推送语言，从指定推送模板中匹配对应语言的推送内容进行远程推送。

> 提示
>
> - 客户端 SDK 从 5.0.0 版本开始，支持在发消息时指定使用一个推送模板。
> - App 用户需要通过客户端 SDK 设置接收推送的语言，否则服务端会默认使用 App Key 级别的 **Push 语言**设置为其匹配推送文案。

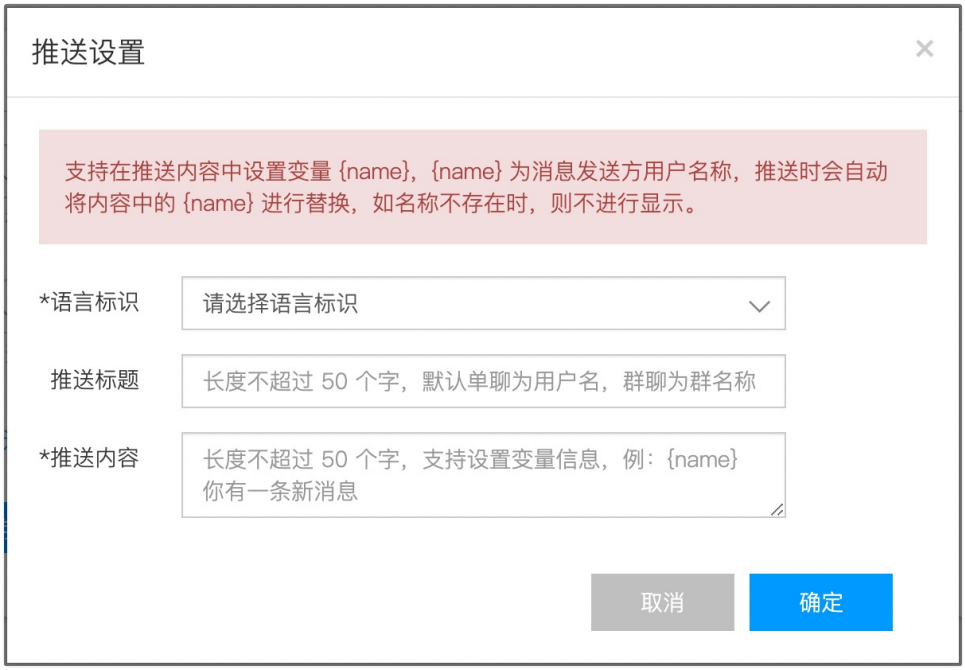
创建多语言推送模板

在使用推送模板功能前，必须在控制台创建多语言推送模板。

1. 访问控制台 自定义推送文案 页面，创建推送模板，最多可创建 100 个。

   - **推送模板 ID**：推送模板的唯一标识。发送消息时，若使用该推送模板，需填入此模板 ID。
   - **推送模板名称**：设置推送模板的名称。
   - **推送内容设置**：每个推送模板中支持设置多个语言的推送内容，每种语言对应一条推送标题和一条推送内容。

2. 添加推送内容。您可以按**语言标识**逐个设置对应的推送标题与推送内容。例如，模板 `asia` 中可同时添加 zh_CN、zh_TW、zh_HK、ja_JP、ko_KR 等多种语言的对应文案。



- **语言标识**：从下拉菜单中选择一个语言标识，如 en_US，以添加与该语言匹配的文案。
- **推送标题**：设置对应语言的通知栏标题。可选参数，默认单聊为用户名，群聊、超级群为群名称。请确保未在消息的推送通知 `pushExt` 中设置推送标题，否则推送模板中设置的标题不生效。
- **推送内容**：设置对应语言的通知栏显示内容。请在发送消息时 `pushContent` 传入空值，否则推送模板中设置的内容不生效。

创建模板后 30 分钟生效。

使用自定义推送模板

IM Server API 支持在发送消息时，为消息启用指定的多语言推送模板。设置后，如该消息触发推送，服务端会根据指定推送模板中的多语言语言内容进行推送。

如果接收方客户端设置了推送语言，服务端将从推送模板中匹配用户上报的推送语言进行远程推送。如果用户未设置，或无法匹配用户推送语言，服务端根据 App 在即时通讯服务端的默认推送语言内容进行推送。

客户端 SDK 在设置接收推送的语言时，语言标识需要控制台的语言标识一致，才能匹配成功。例如，客户端设置推送语言为美式英语 en-US，而推送模板中美式英语的表示法为 en_US，则无法成功匹配。

> 提示
>
> - 关于如何指定推送模版 ID（`templateId`），详见各 API 的参数说明。
> - 关于客户端如何设置接收推送的语言，详见各平台客户端推送开发指南中的用户级推送配置。
> - 您可以修改 App Key 在即时通讯服务的默认推送语言配置。如有需要，请提交工单申请更改 App Key 级别的 **Push 语言**。

设置用户级推送备注名

更新时间:2024-08-30

为指定接收者用户设置推送通知中发送者的显示名称，单次可设置最多 100 个备注名。

默认情况下，即时通讯服务端发送的推送通知会携带发送者注册用户（/user/register.json）或修改用户信息（/user/refresh.json）时传入的用户名。在社交等场景下，假设接收者与发送者互为好友，接收者可能希望在推送通知中将发送者名称显示为个性化名称（如好友昵称）。通过设置用户级推送备注名功能，可以将接收者用户为其他用户设置的个性化名称配置到即时通讯的推送服务中。配置成功后，推送通知中将显示自定义的推送备注名。

请求方法

POST： https://数据中心域名/user/remarks/set.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
remarks	String	是	设置的目标用户推送备注名 JSON 字符串。详见 remarks 结构说明。

- remarks 结构说明：

remarks 参数	类型	必传	说明
id	String	是	目标用户 ID。单次最多设置 100 个。
remark	String	是	收到目标用户推送时显示的备注名。

请求示例

```
POST /user/remarks/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=uid1&remarks=[{"id":"user11","remark":"remark1"},{"id":"user12","remark":"remark2"}]
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

删除用户级推送备注名

更新时间:2024-08-30

删除用户为指定用户（targetId）设置的推送备注名，单次可删除一个备注。

删除成功后，即时通讯服务端发送的推送通知会携带发送者注册用户（/user/register.json）或修改用户信息（/user/refresh.json）传入的用户名。

> 提示
>
> 如果用户设置了群成员推送备注名，则删除用户级推送备注名后，群消息推送通知中会显示为群成员推送备注名。详见设置群成员推送备注名。

请求方法

POST： https://数据中心域名/user/remarks/del.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	操作者用户ID。
targetId	String	是	需要删除推送备注名的用户 ID。

请求示例

```
POST /user/remarks/del.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

targetId=aaee0319&userId=qq
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200
}
```

查询用户级推送备注名

更新时间:2024-08-30

获取用户设置的推送备注名，支持分页获取。

请求方法

POST： https://数据中心域名/user/remarks/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户 ID。
page	Integer	否	页数，默认为第一页。
size	Integer	否	每页条数，默认每页 50 条。

请求示例

```
POST /user/remarks/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=7
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
total	Int	用户的备注名总数。

返回值	返回类型	说明
users	Array	JSON 对象数组，包含用户 ID（`id`）和对应的备注名（`remark`）。单次最多返回 50 个用户备注名。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"total":2,
"users":[
{
"id":"10001",
"remark":"name1"
},
{
"id":"10002",
"remark":"name2"
}]
}
```

设置群成员推送备注名

更新时间:2024-08-30

设置群消息推送通知中发送者的显示名称。

默认情况下，即时通讯服务端为群消息发送的推送通知会携带发送者注册用户（/user/register.json）或修改用户信息（/user/refresh.json）传入的用户名。

在社交等场景下，可能希望在群消息的推送通知中将发送者名称显示为群组内的昵称或其他个性化名称。通过设置指定群成员推送备注功能，可以在即时通讯的推送服务中设置指定用户（userId）在指定群组（groupId）的群消息推送通知中显示的个性化名称（推送备注名）。

配置成功后，其他群成员在收到该用户的推送通知时，发送者名称显示为发送者的群成员推送备注名。

> 提示
>
> 如果推送通知的接收者已为发送者用户设置了用户推送备注名，则群消息推送通知中会优先使用用户推送备注名。详见设置用户推送备注名。

请求方法

POST： https://数据中心域名/group/remarks/set.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	群成员用户 ID。
groupId	String	是	群 ID。
remark	String	是	群成员推送备注。

请求示例

```
POST /group/remarks/set.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=a&groupId=123&remark=备注
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200}
```

删除群成员推送备注名

更新时间:2024-08-30

删除指定用户（userId）在指定群组（groupId）的群消息推送通知中显示的推送备注名。

删除成功后，即时通讯服务端发送的推送通知会携带发送者注册用户（/user/register.json）或修改用户信息（/user/refresh.json）时传入的用户名。

> 提示
>
> 如果推送通知的接收者为发送者用户设置了用户级推送备注名，则删除群成员推送备注名后，群消息推送通知中会使用用户级推送备注名。详见设置用户级推送备注名。

请求方法

POST： https://数据中心域名/group/remarks/del.json

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

频率限制： 每秒钟限 100 次

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	群成员用户 ID。
groupId	String	是	群 ID。

请求示例

```
POST /group/remarks/del.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

groupId=aaee0319&userId=qq
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200
}
```

查询群成员推送备注名

更新时间:2024-08-30

获取指定用户（userId）在指定群组（groupId）的群消息推送通知中显示的推送备注名。

请求方法

POST： https://数据中心域名/group/remarks/get.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 application/x-www-form-urlencoded，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	群成员用户 ID。
groupId	String	是	群 ID。

请求示例

```
POST /group/remarks/get.json HTTP/1.1
Host: api.rong-api.com
App-Key: uwd1c0sxdlx2
Nonce: 14314
Timestamp: 1408710653491
Signature: 45beb7cc7307889a8e711219a47b7cf6a5b000e8
Content-Type: application/x-www-form-urlencoded

userId=uid1&groupId=100
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
remark	String	备注名称。

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
"code":200,
"remark":"name1"
}
```

获取 JWT Token

更新时间:2024-08-30

本文描述如何从即时通讯服务端获取翻译插件鉴权专用的 JWT（JSON Web Token）。JWT 是一种基于 JSON 的轻量级开放标准，用于在网络应用之间安全地传输声明信息。JWT 由三个主要部分组成：头部（Header）、负载（Payload）和签名（Signature）。Payload 包含了 JWT 的主体信息，通过解析可获得 JWT 的有效期（两个小时）、UserId 等信息。

翻译插件鉴权专用的 JWT Token 不同于 IM 用户连接 IM 服务的 Token，请注意区分。

> 提示
>
> - 该接口暂仅适用于使用新加坡数据中心的应用。详见海外数据中心。
> - 调用该接口必须使用专用的 API 域名： https://ai.sg-light-edge.com

适用场景

即时通讯业务客户端 SDK 提供了翻译插件，可为 IMLib 与 IMKit SDK 快速接入外部翻译服务。客户端从 App 后端获取有效的 JWT，才能向即时通讯服务端请求翻译结果。

即时通讯服务端提供了获取 JWT 的 API 接口。该接口必须由 App 后端调用，成功获取 JWT 后由 App 后端返回给客户端。

获取和刷新 JWT 流程图

请求方法

POST： https://ai.sg-light-edge.com/jwt/getToken.json

频率限制： 每秒钟限 100 次

签名规则： 所有服务端 API 请求均需要进行规则校验，详见 API 请求签名。

正文参数

HTTP 请求正文数据格式为 `application/x-www-form-urlencoded`，支持以下 HTTP 表单参数：

参数	类型	必传	说明
userId	String	是	用户Id

请求示例

```
POST /jwt/getToken.json HTTP/1.1
Host: ai.sg-light-edge.com
App-Key: your-own-app-key
Nonce: 14314
Timestamp: 1408710653000
Signature: 30be0bbca9c9b2e27578701e9fda2358a814c88f
Content-Type: application/x-www-form-urlencoded
Content-Length: 78

userId=jlk456j5
```

返回结果

HTTP 响应正文包含具有以下结构的 JSON 对象：

返回值	返回类型	说明
code	Number	返回码，200 为正常。
token	String	返回的JWT

返回结果示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"code":200,"token":""}
```

Server SDK

更新时间:2024-08-30

为了帮助广大开发者节约开发资源和时间，我们在 GitHub 上提供了一组开源的 Server SDK。开发者可选择适用的语言版本。

GitHub 项目如下：

- server-sdk-java (GitHub) · (Gitee)
- server-sdk-php (GitHub) · (Gitee)
- server-sdk-go (GitHub) · (Gitee)

状态码 HTTP 状态码

更新时间:2024-08-30

状态码	描述	详细解释
200	成功	成功
400	请求参数错误	该请求是无效的，详细的错误信息会说明原因
401	未授权	验证失败，详细的错误信息会说明原因
403	服务器拒绝请求	被拒绝调用，详细的错误信息会说明原因
404	未找到	服务器找不到请求的地址
405	请求方法不支持	群容量超出上限，禁止请求此方法
429	请求频率超限	超出了调用频率限制，详细的错误信息会说明原因
430	未开通服务	一般为未开通服务导致，详细的错误信息会说明原因
500	服务器内部错误	服务器内部出错了，请联系我们尽快解决问题
504	网关超时	服务器在运行，本次请求响应超时，请稍后重试

业务返回码

状态码	描述	详细解释
404	未找到	服务器找不到请求的地址（HTTP 状态码 404）
1000	服务内部错误	服务器端内部逻辑错误,请稍后重试（HTTP 状态码 500）
1001	App Secret 错误	App Key 与 App Secret 不匹配（HTTP 状态码 401）

状态码	描述	详细解释
1002	参数错误	参数错误，详细的描述信息会说明（HTTP 状态码 400）
1003	无 POST 数据	没有 POST 任何数据（HTTP 状态码 400）
1004	验证签名错误	验证签名错误（HTTP 状态码 401）
1005	参数长度超限	参数长度超限，详细的描述信息会说明（HTTP 状态码 400）
1006	App 被锁定或删除	App 被锁定或删除（HTTP 状态码 401）
1007	被限制调用	该方法被限制调用，详细的描述信息会说明（HTTP 状态码 401）
1008	调用频率超限	调用频率超限。注意，即使未开通 API 相关服务，如果连续请求导致 API 调用频率超限，也会触发该错误。（HTTP 状态码 429）
1009	服务未开通	未开通该服务，请到开发者管理后台开通或提交工单申请。（HTTP 状态码 430）
1010	服务未开通	未开通白名单服务，请到开发者管理后台开通或提交工单申请。（HTTP 状态码 430）
1011	服务未开通	未开通黑名单服务，请到开发者管理后台开通或提交工单申请。（HTTP 状态码 430）
1015	删除的数据不存在	要删除的保活聊天室 ID 不存在。（HTTP 状态码 200）
1016	设置保活聊天室个数超限	设置的保活聊天室个数超限。（HTTP 状态码 403）
1017	实时音视频 SDK 版本不支持	音视频 SDK 版本不支持，请使用 2.9.0 及之后的实时音视频 SDK。
1018	实时音视频服务未开启	未开启服务，请到开发者管理后台开通实时音视频服务。（HTTP 状态码 403）
1050	内部服务超时	内部服务响应超时（HTTP 状态码 504）
2007	注册用户数量超限	开发环境下注册用户上限为 100 个，生产环境未开通 IM 付费版本时可免费注册 100个，超过后需要升级到付费版本，私有部署客户请联系商务；单独使用音视频服务没有注册用户数限制，如您单独使用音视频服务触发了该错误，请检查您的音视频服务是否因欠费或其它原因处于关停状态。
20107	单聊黑名单人数超限	调用「添加用户到黑名单」接口，用户 ID 的黑名单人数已达到上限（3000 ），返回该错误码。
20108	单聊白名单人数超限	调用「添加用户到白名单」接口，用户 ID 的白名单人数已达到上限（3000 ），返回该错误码。

状态码	描述	详细解释
23407	聊天室禁言白名单人数超限	单次 API 请求中聊天室中禁言白名单用户已达到上限（20 个），返回该错误码。请分次添加禁言白名单用户。
23423	聊天室属性超限	单个聊天室最多设置 100 个属性 KV。如果超过个数上限，当前请求设置的属性全部设置失败，返回该错误码和当前已有的属性数量。
23426	聊天室属性功能未开通	聊天室自定义属性未开通时，返回该错误码。
23410	聊天室不存在	聊天室不存在。
23432	聊天室销毁时间不合法	设置聊天室销毁时间（分钟）非有效值时，返回该错误码。有效值为 [60 - 10080] 的整数。
23433	聊天室销毁类型不合法	设置聊天室销毁类型非有效值时，返回该错误码。有效值请参见文档中定义的类型对应值。
23434	聊天室属性无对应用户 ID	如果设置聊天室自定义属性，必须同时传用户 ID，否则返回该错误码。
23435	聊天室已存在	创建聊天室时聊天室已存在，返回该错误码。
23436	聊天室存活时长无效	为定时销毁的聊天室设置存活时长时，应保证传入的存活时长大于聊天室自创建后的已存活时长，否则返回该错误码。
24353	重复注销	调用注销接口，传入已完成注销的用户 ID 时，返回该错误码。
24354	重复激活	调用重新激活用户 ID 的接口，传入已激活的用户 ID 时，返回该错误码。
24356	用户注销中	调用注销接口，传入正在注销流程中的用户 ID 时，返回该错误码。
24410	超级群不存在	未找到超级群，请先进行创建
24412	用户加入超级群数量超限	用户加入超级群数量超限，每个用户最多可加入 100 个超级群。
24413	频道数超出上限	该超级群下创建的频道数超限，每个超级群下默认可创建 50 个频道。
24414	超级群频道 ID 不存在	请确认该超级群下存在该频道。
24416	发送者不在超级群私有频道成员列表中	请确认该超级群下该私有频道的成员列表中是否包含该用户 ID，或者发送消息的频道 ID 是否填写正确。
24418	用户组数量超出限制	请检查该超级群下用户组数量。默认最多 50 个。

状态码	描述	详细解释
24419	重复的用户组 ID	请检查入参中的用户组 ID 是否重复。
24420	用户组不存在	请确认该超级群下是否存在该用户组，入参中的超级群 ID、用户组 ID 等是否填写正确。
24421	用户组成员数量超出限制	请检查该超级群用户组中的成员数量。默认最多 100 个用户。
24422	频道绑定的用户组数量超出限制	请检查该超级群频道绑定的用户组数量。默认单个频道最多支持与 10 个用户组绑定。