

# **Chat**

## **Server APIs**

### **Product Documentation**



## Copyright Notice

©2013-2025 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## Server APIs

- Secure authentication with UserSig

## RESTful APIs

- RESTful API Overview

- RESTful API List

## Message Related

### Send Message

- Sending One-to-One Messages to One User
- Sending One-to-One Messages to Multiple Users
- Sending Ordinary Messages in a Group
- Sending System Messages in a Group
- Broadcast Message of Audio-Video Group
- Importing One-to-One Messages
- Importing Group Messages
- Sending Messages in a Official Account

### Historical Message

- Modifying Historical One-to-one Messages
- Modifying Historical Group Messages
- Pulling Historical One-to-one Messages
- Pulling Historical Group Messages
- Pulling Historical Official Account Messages

### Delete Message

- Deleting Messages Sent by a Specified User
- Clearing Historical Group Messages

### Withdraw Message

- Recalling One-to-One Messages
- Recalling Group Messages
- Recalling Official Account Messages

### Read Receipt

- Marking One-to-One Messages as Read
- Pulling Group Message Read Receipt Details
- Pulling Read Receipts for Group Messages

### Message Extension

- Pulling the Message Extension of a One-to-One Message
- Configuring Message Extension for a One-to-One Message

Pulling Message Extension of a Group Message

Configuring Message Extension for a Group Message

Pushing to All Users

API for Pushing to All Users

Pushing to All Users

Setting Application Attribute Names

Getting Application Attribute Names

Getting User Attributes

Setting User Attributes

Deleting User Attributes

Getting User Tags

Adding User Tags

Deleting User Tags

Deleting All Tags of a User

Session Related

Conversation List

Pulling a conversation list

Session Unread Count

Setting the Unread Message Count of a Member

Querying Unread One-to-One Message Count

Delete Session

Deleting a conversation

Session Grouping Tag

Creating Conversation Group Data

Updating Conversation Group Data

Deleting Conversation Group Data

Creating or Updating Conversation Mark Data

Searching for Conversation Group Marks

Pulling Conversation Group Mark Data

Group Related

Group Management

Getting All Groups in an App

Creating a Group

Disbanding a Group

Getting the Groups a User Has Joined

Group Information

Getting Group Profiles

Modifying the Profile of a Group



Importing a Group Profile

#### Group Member Management

Adding Group Members

Deleting Group Members

Banning Group Members

Unbanning Group Members

Bulk Muting and Unmuting

Getting the List of Muted Group Members

Changing Group Owner

Querying the Roles of Users in a Group

Importing Group Members

#### Group Member Information

Getting Group Member Profiles

Getting Specified Group Member Profiles

Modifying the Profile of a Group Member

#### Group Custom Attributes

Getting Group Custom Attributes

Modifying Group Custom Attributes

Clearing Group Custom Attributes

Resetting Group Custom Attributes

Deleting Group Custom Attributes

#### Live Group Management

Setting Live Streaming Group Robots

Deleting Live Streaming Group Robots

Setting/Deleting Live Streaming Group Administrators

Obtaining the List of Live Streaming Group Administrators

Checking Whether Users Are in a Live Streaming Group

Getting the Number of Online Users in an Audio-Video Group

Getting the List of Online Members in Audio-Video Group

Setting Audio-Video Group Member Marks

Getting the List of Banned Group Members

#### Community Management

Creating Topic

Deleting Topic

Getting Topic Profile

Modifying Topic Profile

Importing Topic Profiles

Permission Group Management

- Creating Permission Groups
- Terminating Permission Groups
- Modifying Permission Group Information
- Obtaining Permission Group Information
- Adding Topic Permissions
- Modifying Topic Permissions
- Deleting Topic Permissions
- Obtaining Topic Permissions
- Adding Members to a Permission Group
- Deleting Permission Group Members
- Obtaining Permission Group Member List
- Getting the List of Online Members in Community Group
- Group Counter
  - Getting Group Counters
  - Updating Group Counters
  - Deleting Group Counters
- User Management
  - Account Management
    - Importing a Single Account
    - Importing Multiple Accounts
    - Deleting Accounts
    - Querying Accounts
  - User Information
    - Setting Profiles
    - Pulling Profiles
  - User Status
    - Invalidating Account Login States
    - Querying Account Online Status
  - Friend Management
    - Adding Friends
    - Importing Friends
    - Updating Friends
    - Deleting Friends
    - Deleting All Friends
    - Verifying Friends
    - Pulling Friends
    - Pulling Specified Friends
  - Friend Lists

- Adding Lists

- Deleting Lists

- Pulling Lists

#### Blocklist

- Blocklisting Users

- Unblocklisting Users

- Pulling a Blacklist

- Verifying Users on a Blocklist

#### Following and Fans

- Following Users

- Unfollow Users

- Get The Fans, Follows, And Mutual Followers Count

- Checking Follow Relationship

- Get The Fans, Follows, And Mutual Followers Count

#### Global Mute Management

- Setting Global Mute

- Querying Global Mute

#### Operations Management

- Pulling Operations Data

- Downloading Recent Messages

- Getting Server IP Addresses

#### Chatbots

- Pulling Chatbot Accounts

- Creating Chatbot Accounts

- Deleting Chatbot Accounts

#### Official Channel Management

- Official Channel Related

- Creating Official Channel

- Terminating Official Channel

- Modifying Official Channel Information

- Obtaining Official Channel Information

- Obtaining Subscriber Information of Official Channel

#### Subscriber Relevant

- Adding Subscribers

- Deleting Subscribers

- Modifying Subscriber Information

- Obtaining Subscribed Official Channel

#### All/Tagged Push

- Pushing to All/Tagged Users
- Obtaining Application Attribute Names
- Setting Application Attribute Names
- Obtaining User Attributes
- Setting User Attributes
- Deleting User Attributes
- Obtaining User Tags
- Adding User Tags
- Deleting User Tags
- Deleting All User Tags
- Recalling Push

## Webhooks

- Webhook Overview
- Webhook Command List
- Operations Management Callbacks
  - API Overclocking Alarm Callbacks
- Online Status Webhooks
  - Status Change Webhooks
- Relationship Chain Webhooks
  - After a Profile Is Updated
  - Before a Friend Is Added
  - Before a Friend Request Is Responded
  - After a Friend Is Added
  - After a Friend Is Deleted
  - After a User Is Added to Blocklist
  - After a User Is Removed from Blocklist
- Following and Fans Related Webhooks
  - After a User Is Followed
  - After a User Is Unfollowed
- One-to-One Message Webhooks
  - Before a One-to-One Message Is Sent
  - After a One-to-One Message Is Sent
  - After a One-to-One message Is Marked as Read
  - After A One-to-One Message Is Recalled
- Group Webhooks
  - Group Creation Daily Net Increase Threshold Alarm Callback
  - Before a Group Is Created
  - After a Group Is Created

- Before Applying to Join a Group
- Before Inviting a User to a Group
- After a User Joins a Group
- After a User Leaves a Group
- Before Group Message Is Sent
- After a Group Message Is Sent
- After a Group Is Full
- After a Group Is Disbanded
- After Group Profile Is Modified
- Callback After Recalling Group Messages
- Webhook for Online and Offline Status of Audio-Video Group Members
- Webhook for Exceptions When Group Messages Are Sent
- Before a Topic Is Created
- After a Topic Is Created
- After a Topic Is Deleted
- Topic Profile Change Webhook
- Callback After Group Member Profile Changed
- Callback After Group Attribute Changed
- Callback After Read Receipt
- Callback After the Group Owner Changed

Chatbot Callbacks

- Callback After Sending a One-to-One Message to a Chatbot
- Callback After Reminding a Chatbot in a Group

Webhooks related to the Official Channel

- Before a Official Channel Is Created
- After a Official Channel Is Created
- After Official Channel Profile Is Modified
- After Official Channel Is Destroyed
- Before Official Channel Is Subscribed
- After Official Channel Is Subscribed
- After a Official Channel Is Full
- After Official Channel Is Unsubscribed
- Before Official Channel Message Is Sent
- Callback After Sending an Official Channel Message
- Webhook After Recalling Official Channel Messages

# Server APIs

## Secure authentication with UserSig

Last updated : 2025-03-20 16:35:37

This document outlines two authentication methods for Chat services, with a focus on UserSig, a security signature by Tencent Cloud to safeguard against unauthorized access. For basic cloud service usage, provide SDKAppID, UserID, and UserSig during SDK initialization or login.

SDKAppID is used to identify your application.

UserID is used to identify your user.

UserSig is a security signature calculated based on the first two using the HMAC SHA256 encryption algorithm. As long as attackers cannot forge the UserSig, they cannot steal your cloud service traffic.

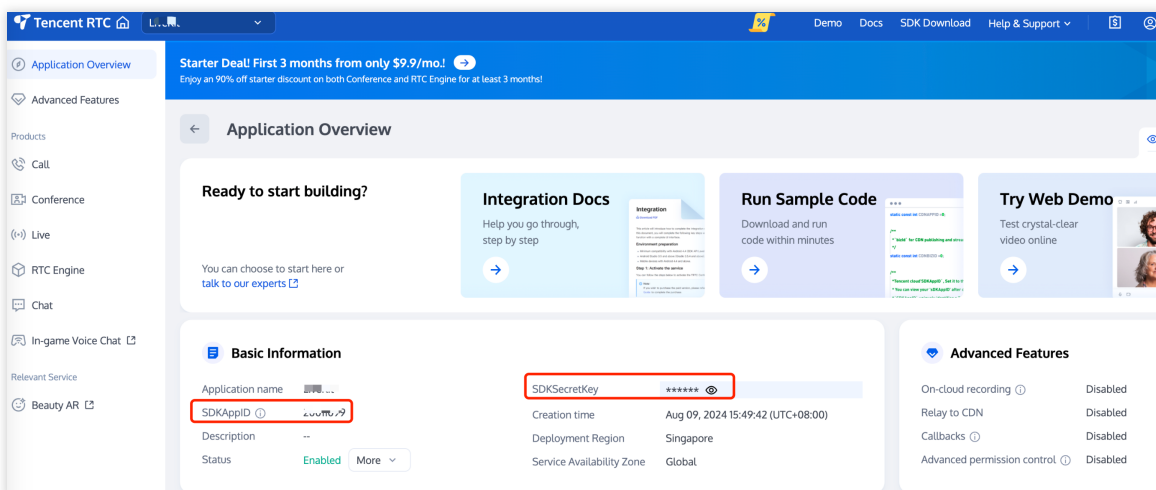
## Obtaining a Key

1. Log in to the [Chat console](#).

### Note:

If you do not have any app, [create an app](#) and then perform [step 2](#).

2. Click the target app card to go to its basic configuration page.
3. In the **Basic Information** section, click **Display key** to the right of **Key**.



4. Click **Copy** to copy and save the key information.

### Note:

Store the key information properly to prevent disclosure.

## Calculating UserSig on the Client

The `GenerateTestUserSig` open-source module provided in the sample code of the Chat SDK can help you quickly generate a UserSig. You only need to configure three member variables, including SDKAPPID (SDKAppID of the app), EXPIRETIME (UserSig expiration time), and SECRETKEY (key information), and then call the `genTestUserSig()` function to quickly obtain a UserSig.

To simplify this process, we provide the source code for computing a UserSig for the following languages and platforms. You can directly download and integrate the source code into your client.

Programing Language	Platform	GenerateTestUserSig Source Code
Java	Android	<a href="#">GenerateTestUserSig.java</a>
Objective-C	iOS	<a href="#">GenerateTestUserSig.h</a>
Objective-C	Mac	<a href="#">GenerateTestUserSig.h</a>
C++	Windows	<a href="#">GenerateTestUserSig.h</a>
Javascript	Web	<a href="#">GenerateTestUserSig.js</a>
Dart	Flutter	<a href="#">GenerateTestUserSig.dart</a>

#### Note:

In this method, the `SECRETKEY` is vulnerable to decompilation and reverse engineering. Once your `SECRETKEY` is disclosed, attackers can steal your Tencent Cloud traffic. Therefore, **this method is only suitable for locally running a demo project and feature debugging.**

The correct way to issue a UserSig is to integrate the UserSig computing code into your server and provide app-oriented APIs. When UserSig is needed, your app will send a request to the business server to obtain a dynamic UserSig. For more information, see [How to Calculate UserSig](#).

## Calculating UserSig on the Server

Generating a UserSig on the server provides maximum protection against the disclosure of the key used for calculating the UserSig. You only need to deploy the code for calculating the UserSig on your server and provide an app-oriented API. When a UserSig is needed, your app will send a request to the business server to obtain a dynamic UserSig.

To simplify this process, we provide the source code for calculating a UserSig for the following languages and platforms. You can directly download and integrate the source code into your server.

Programming Language	Key Function	Download URL
Java	HMAC-SHA256	<a href="#">genSig</a>

GO	HMAC-SHA256	<a href="#">GenSig</a>
PHP	HMAC-SHA256	<a href="#">genSig</a>
Nodejs	HMAC-SHA256	<a href="#">genSig</a>
Python	HMAC-SHA256	<a href="#">gen_sig</a>
C#	HMAC-SHA256	<a href="#">GenSig</a>
C++	HMAC-SHA256	<a href="#">gen_sig</a>

Key fields in a UserSig calculation function include the SDKAppID, UserID, and UserSig validity period, as described in the following table.

**Note:**

The following table uses the field names in the Java source code as an example. The field names may be different in other languages.

Field Name (Example)	Description
sdkappid	SDKAppID of the app. You can obtain the SDKAppID on the app card in the <a href="#">Chat console</a> .
userid	User ID (former name: <code>Identifier</code> ).
expire	UserSig validity period, in seconds.
userbuf	This field is set to <code>null</code> by default because APIs without UserBuf are used in Chat by default. APIs with UserBuf may be required in some TRTC use cases, for example, when entering a room. For more information, see <a href="#">Enabling Advanced Permission Control</a> .
key	Key. You can obtain a key on the app details page in the <a href="#">Chat console</a> . For more information, see <a href="#">Obtaining a Key</a> .

## Old Version of Algorithm

To simplify signature computing so that customers can conveniently and quickly use Tencent Cloud services, the signature algorithm of the Chat service has been upgraded from ECDSA-SHA256 to HMAC-SHA256 since July 19, 2019. This means that all SDKAppIDs created after July 19, 2019 will use the new HMAC-SHA256 algorithm. If your SDKAppID was created before July 19, 2019, we recommend that you upgrade the signature algorithm to [HMAC-SHA256](#). The upgrade will not affect your business. Alternatively, you can still use the signature algorithm of an earlier version. The URLs for downloading the source code for the ECDSA-SHA256 algorithm are as follows:

--	--	--



Programming Language	Signature Algorithm	Download Link
Java	ECDSA-SHA256	<a href="#">GitHub</a>
GO	ECDSA-SHA256	<a href="#">GitHub</a>
PHP	ECDSA-SHA256	<a href="#">GitHub</a>
Nodejs	ECDSA-SHA256	<a href="#">GitHub</a>
Python	ECDSA-SHA256	<a href="#">GitHub</a>
C#	ECDSA-SHA256	<a href="#">GitHub</a>
C++	ECDSA-SHA256	<a href="#">GitHub</a>

## Get UserSig from the console

Log in to Tencent-RTC Console, navigate to Development Tools > [UserSig Tools](#).

Under the UserSig Generation Tool, select the corresponding SDKAppID and UserID.

Click the Generate button to compute the corresponding UserSig.

Tencent RTC

Demo

Docs

SDK Download

Help & Support

min.hellow

Overview

Applications

Usage Statistics

Data Monitoring

Package Management

Relevant Services

Beauty AR

Development Tools

UserSig Tools

RTMP Address Generator

Starter Deal! First 3 months from only \$9.9/mo!

Enjoy an 90% off starter discount on both Conference and RTC Engine for at least 3 months!

UserSig Tools

Signature (UserSig) Generator

This tool can quickly generate a UserSig, which can be used to run through demos and to debug features.

Application (SDKAppID)

20011099-LiveKit

Username (UserID)

denny

SDKSecretKey

Generate

Generate result

Copy

Signature (UserSig) Verifier

This tool is used to verify the validity of the UserSig you use.

Application (SDKAppID)

Select an applicaiton

Username (UserID)

Set the user name

SDKSecretKey

Auto-generated after you select an application

UserSig

Please enter

Verify

©2013-2025 Tencent Cloud International Pte. Ltd.

Page 14 of 781

# RESTful APIs

## RESTful API Overview

Last updated : 2025-03-26 16:03:28

RESTful APIs are HTTP management APIs that provide the app backend with a management entry. For more information about RESTful APIs that Chat supports, see [RESTful API List](#).

In addition to RESTful APIs, the app console also supports simple data management and one-to-one/group messaging. Developers can manage, check, and test data in the console. Even though RESTful APIs are less user-friendly, they provide powerful management capabilities.

For security concerns, RESTful APIs are supported over HTTPS only.

## Prerequisites

Before you call a RESTful API, complete the following operations:

1. Create an app in the Chat console. For more information, see [Creating and Upgrading an Application](#).
2. Assign an admin account to the app. For more information, see the **Configuring Account Admins** section in [Basic Configuration](#).

### Caution

To avoid unnecessary call errors, use the admin account to call a RESTful API.

## Calling Method

### Request URL

The URL format of a RESTful API is as follows:

```
https://xxxxxx/$ver/$servicename/$command?  
sdkappid=$SDKAppID&identifier=$identifier&usersig=$usersig&random=99999999&cont  
enttype=json
```

The descriptions and values of parameters are as follows (parameter names and parameter values are case-sensitive):

Parameter	Description	Value
https	Request protocol	The request protocol is HTTPS, and the request method is POST.
xxxxxx	Dedicated domain name	China: console.tim.qq.com (backup: adminapi.my-

		<p>imcloud.com)</p> <p>Singapore: adminapisgp.im.qcloud.com (backup: adminapisgp.my-imcloud.com)</p> <p>Seoul: adminapikr.im.qcloud.com (backup: adminapikr.my-imcloud.com)</p> <p>Frankfurt: adminapiger.im.qcloud.com (backup: adminapiger.my-imcloud.com)</p> <p>Silicon Valley: adminapiusa.im.qcloud.com (backup: adminapiusa.my-imcloud.com)</p> <p>Jakarta: adminapiidn.im.qcloud.com (backup: adminapiidn.my-imcloud.com)</p>
ver	Protocol version number	Always <code>v4</code> .
servicename	Internal service name. Different values of <code>servicename</code> correspond to different service types.	<p>Example: For <code>v4/im_open_login_svc/account_import</code> , <code>im_open_login_svc</code> is the <code>servicename</code> . For more information, see <a href="#">RESTful API List</a>.</p>
command	Command word. This parameter is used with the <code>servicename</code> parameter to identify a specific service feature.	<p>Example: For <code>v4/im_open_login_svc/account_import</code> , <code>account_import</code> is the <code>command</code> . For more information, see <a href="#">RESTful API List</a>.</p>
sdkappid	App ID obtained in the Chat console	You can obtain the SDKAppID when applying for Chat SDK access.
identifier	Username, which must be the app admin account when a RESTful API is called	For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Password that corresponds to the user name.	For more information, see <a href="#">Generating UserSig</a> .
random	A parameter used to identify the current request	A random 32-bit unsigned integer ranging from 0 to 4,294,967,295
contenttype	Request format	Always <code>json</code> .

### Note

When the app server calls a RESTful API, `identifier` must be the app admin account.

The app can generate a `UserSig` for the admin account each time it calls a RESTful API or generate a fixed `UserSig` for reuse with period of validity.

## HTTP request body format

RESTful APIs only support the POST method, and its request body is in JSON format. For more information about the request body format, see the detailed description of each API.

Note that POST request bodies cannot be empty. Even when no information is required in a request body, the request body still needs to carry an empty JSON object ( `{ }` ).

## HTTP return code

The returned HTTP status code for RESTful APIs is always 200 unless a network error (such as error 502) occurs. The specific error code and error message are included in the HTTP response body.

## HTTP response body format

The RESTful API response body is in the JSON format and has the following elements:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
  // Other RESTful API response content
}
```

The response body must contain the `ActionStatus` , `ErrorInfo` , and `ErrorCode` fields. These three fields are described as follows:

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed. If the request fails, the cause of failure is displayed in the <code>ErrorInfo</code> field.
ErrorInfo	String	Failure causes
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For more information, see <a href="#">Error Codes</a> .

## Sample Call

The following example shows how to use the RESTful API to [get all groups in an app](#).

HTTPS request:

```
POST /v4/group_open_http_svc/get_appid_group_list?
usersig=xxx&identifier=admin&sdkappid=888888888&random=999999999&contenttype=json
HTTP/1.1
Host: console.tim.qq.com
Content-Length: 22
{
  "Limit": 2
}
```

#### HTTPS response:

```
HTTP/1.1 200 OK
Server: nginx/1.7.10
Date: Fri, 09 Oct 2015 02:59:55 GMT
Content-Length: 156
Connection: keep-alive
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: POST

{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "GroupIdList": [
    {
      "GroupId": "@TGS#1YTTZEAEG"
    },
    {
      "GroupId": "@TGS#1KVTZEA EZ"
    }
  ],
  "TotalCount": 58530
}
```

## RESTful API Common Error Codes

Error Code	Description
60002	HTTP parsing error. Check the URL format of the HTTP request.
60003	JSON parsing error. Check the JSON format of the HTTP request.
60004	Account or signature error in the request URL or JSON request body.

60005	Account or signature error in the request URL or JSON request body.
60006	Invalid SDKAppID. Check the validity of SDKAppID.
60007	RESTful API call frequency limit exceeded. Reduce your request frequency.
60008	Service request timeout or HTTP request format error. Check and try again.
60009	Request resource error. Check the request URL.
60010	The request requires app admin permissions.
60011	SDKAppID request frequency exceeded. Reduce your request frequency.
60012	SDKAppID is required for RESTful APIs. Check the SDKAppID parameter in the URL.
60013	JSON parsing error in the HTTP response body.
60014	Account switching timeout.
60015	Invalid account type in the request body. Make sure that the account is in string format.
60016	The SDKAppID is disabled.
60017	The request is disabled.
60018	Too many requests. Try again later.
60019	Too many requests. Try again later.
60020	Your Standard edition has expired and was disabled. Log in to the <a href="#">purchase page</a> and purchase it again. It will take effect in five minutes upon successful purchase.
60021	The source IP of the RESTful API call is invalid.

## FAQs

### The RESTful API request timed out and no response was received.

1. The timeout period specified for the RESTful API in the Chat backend is three seconds. You need to specify a timeout period that is longer than three seconds.
2. Run `telnet console.tim.qq.com 443` to check whether the service port is accessible.
3. Run `curl -I https://console.tim.qq.com` to check whether the status code is 200.
4. Check whether the machine uses a private or public DNS server. If the machine uses a private DNS server, make sure that the DNS server egress is in the same region as the ISP to which the egress IP of the machine belongs.

5. You are advised to use the "persistent connection+connection pool" mode.

**Note**

It is recommended that you use a RESTful API persistent connection to connect to the SDK. The reason is that it takes a long time to establish HTTPS non-persistent connections because each request causes TCP+TLS handshake overhead.

For scenario where a standard HTTP library is used: for HTTP 1.0, the request header `Connection: keep-alive` needs to be specified; for HTTP 1.1, persistent connections are supported by default; for scenarios where HTTPS requests are encapsulated based on TCP, TCP connections can be reused to send and receive requests.



# RESTful API List

Last updated : 2024-03-21 15:34:28

Feature	API
Imports a single account.	<a href="#">v4/im_open_login_svc/account_import</a>
Imports multiple accounts.	<a href="#">v4/im_open_login_svc/multiaccount_import</a>
Deletes accounts.	<a href="#">v4/im_open_login_svc/account_delete</a>
Queries accounts.	<a href="#">v4/im_open_login_svc/account_check</a>
Invalidating account login states	<a href="#">v4/im_open_login_svc/kick</a>
Queries the login status of an account.	<a href="#">v4/openim/query_online_status</a>

## One-to-One Message

Feature	API
Sends one-to-one messages to one user.	<a href="#">v4/openim/sendmsg</a>
Sends one-to-one messages to multiple users.	<a href="#">v4/openim/batchsendmsg</a>
Imports one-to-one messages.	<a href="#">v4/openim/importmsg</a>
Queries one-to-one messages.	<a href="#">v4/openim/admin_getroammsg</a>
Recalls one-to-one messages.	<a href="#">v4/openim/admin_msgwithdraw</a>
Marks one-to-one messages as read.	<a href="#">v4/openim/admin_set_msg_read</a>
Queries the unread one-to-one message counts.	<a href="#">v4/openim/get_c2c_unread_msg_num</a>
Modifies historical one-to-one messages	<a href="#">v4/openim/modify_c2c_msg</a>

## Pushing to All Users

Feature	API

Pushes to all users.	<a href="#">v4/all_member_push/im_push</a>
Sets app attribute names.	<a href="#">v4/all_member_push/im_set_attr_name</a>
Gets app attribute names.	<a href="#">v4/all_member_push/im_get_attr_name</a>
Gets user attributes.	<a href="#">v4/all_member_push/im_get_attr</a>
Sets user attributes.	<a href="#">v4/all_member_push/im_set_attr</a>
Deletes user attributes.	<a href="#">v4/all_member_push/im_remove_attr</a>
Gets user tags.	<a href="#">v4/all_member_push/im_get_tag</a>
Adds user tags.	<a href="#">v4/all_member_push/im_add_tag</a>
Deletes user tags.	<a href="#">v4/all_member_push/im_remove_tag</a>
Deletes all tags of a user.	<a href="#">v4/all_member_push/im_remove_all_tags</a>

## Profile Management

Feature	API
Configures profiles.	<a href="#">v4/profile/portrait_set</a>
Pulls profiles.	<a href="#">v4/profile/portrait_get</a>

## Relationship Chain Management

Feature	API
Adds friends.	<a href="#">v4/sns/friend_add</a>
Imports friends.	<a href="#">v4/sns/friend_import</a>
Updates friends.	<a href="#">v4/sns/friend_update</a>
Deletes friends.	<a href="#">v4/sns/friend_delete</a>
Deletes all friends.	<a href="#">v4/sns/friend_delete_all</a>
Verifies friends.	<a href="#">v4/sns/friend_check</a>

Pulls friends.	<a href="#">v4/sns/friend_get</a>
Pulls specified friends.	<a href="#">v4/sns/friend_get_list</a>
Blocklists users.	<a href="#">v4/sns/black_list_add</a>
Unblocklists users.	<a href="#">v4/sns/black_list_delete</a>
Pulls a blocklist.	<a href="#">v4/sns/black_list_get</a>
Checks whether specified users are on a user's blocklist and/or vice versa.	<a href="#">v4/sns/black_list_check</a>
Adds lists.	<a href="#">v4/sns/group_add</a>
Deletes lists.	<a href="#">v4/sns/group_delete</a>
Pulls lists.	<a href="#">v4/sns/group_get</a>

## Recent Contacts

Feature	API
Pulls a conversation list.	<a href="#">v4/recentcontact/get_list</a>
Deletes a conversation.	<a href="#">v4/recentcontact/delete</a>
Creates conversation group data.	<a href="#">v4/recentcontact/create_contact_group</a>
Deletes conversation group data.	<a href="#">v4/recentcontact/del_contact_group</a>
Updates conversation group data.	<a href="#">v4/recentcontact/update_contact_group</a>
Searches for conversation group mark data.	<a href="#">v4/recentcontact/search_contact_group</a>
Creates or updates conversation mark data.	<a href="#">v4/recentcontact/mark_contact</a>
Pulls conversation group mark data.	<a href="#">v4/recentcontact/get_contact_group</a>

## Group Management

Feature	API
Gets all groups in an app.	<a href="#">v4/group_open_http_svc/get_appid_group_list</a>

Creates a group.	<a href="#">v4/group_open_http_svc/create_group</a>
Gets group profiles.	<a href="#">v4/group_open_http_svc/get_group_info</a>
Gets group member profiles.	<a href="#">v4/group_open_http_svc/get_group_member_info</a>
Modifies the profile of a group.	<a href="#">v4/group_open_http_svc/modify_group_base_info</a>
Adds group members.	<a href="#">v4/group_open_http_svc/add_group_member</a>
Deletes group members.	<a href="#">v4/group_open_http_svc/delete_group_member</a>
Modifies the profile of a group member.	<a href="#">v4/group_open_http_svc/modify_group_member_info</a>
Disbands a group.	<a href="#">v4/group_open_http_svc/destroy_group</a>
Gets the groups a user has joined.	<a href="#">v4/group_open_http_svc/get_joined_group_list</a>
Queries the roles of users in a group.	<a href="#">v4/group_open_http_svc/get_role_in_group</a>
Mutes and unmutes group members.	<a href="#">v4/group_open_http_svc/forbid_send_msg</a>
Gets the list of muted group members.	<a href="#">v4/group_open_http_svc/get_group_shutted_uin</a>
Sends ordinary messages in a group.	<a href="#">v4/group_open_http_svc/send_group_msg</a>
Sends system messages in a group.	<a href="#">v4/group_open_http_svc/send_group_system_notification</a>
Recalls group messages.	<a href="#">v4/group_open_http_svc/group_msg_recall</a>
Changes the group owner.	<a href="#">v4/group_open_http_svc/change_group_owner</a>
Imports a group profile.	<a href="#">v4/group_open_http_svc/import_group</a>
Imports group messages.	<a href="#">v4/group_open_http_svc/import_group_msg</a>
Imports group members.	<a href="#">v4/group_open_http_svc/import_group_member</a>
Sets the unread message count of a member.	<a href="#">v4/group_open_http_svc/set_unread_msg_num</a>
Deletes messages sent by a specified user.	<a href="#">v4/group_open_http_svc/delete_group_msg_by_sender</a>
Gets group message history.	<a href="#">v4/group_open_http_svc/group_msg_get_simple</a>
Gets the number of online users in an audio-video group.	<a href="#">v4/group_open_http_svc/get_online_member_num</a>
Gets custom attributes of a group.	<a href="#">v4/group_open_attr_http_svc/get_group_attr</a>
Gets the list of banned group members.	<a href="#">v4/group_open_http_svc/get_group_ban_member</a>

Bans group members.	<a href="#">v4/group_open_http_svc/ban_group_member</a>
Unbans group members.	<a href="#">v4/group_open_http_svc/unban_group_member</a>
Modifies custom attributes of a group.	<a href="#">v4/group_open_http_svc/modify_group_attr</a>
Clears custom attributes of a group.	<a href="#">v4/group_open_http_svc/clear_group_attr</a>
Resets custom attributes of a group.	<a href="#">v4/group_open_http_svc/set_group_attr</a>
Modifies historical group chat messages.	<a href="#">v4/openim/modify_group_msg</a>
Delivers broadcast messages to all audio-video groups.	<a href="#">v4/group_open_http_svc/send_broadcast_msg</a>
Gets the group counter.	<a href="#">v4/group_open_http_svc/get_group_counter</a>
Updates the group counter.	<a href="#">v4/group_open_http_svc/update_group_counter</a>
Deletes the group counter.	<a href="#">v4/group_open_http_svc/delete_group_counter</a>

## Global Mute Management

Feature	API
Sets global mute.	<a href="#">v4/openconfigsvr/setnospeaking</a>
Queries global mute.	<a href="#">v4/openconfigsvr/getnospeaking</a>

## Operations Management

Feature	API
Pulls operations data.	<a href="#">v4/openconfigsvr/getappinfo</a>
Downloads recent messages.	<a href="#">v4/open_msg_svc/get_history</a>
Gets server IP addresses.	<a href="#">v4/ConfigSvc/GetIPList</a>

# Message Related

## Send Message

### Sending One-to-One Messages to One User

Last updated : 2024-12-30 10:48:29

## Feature Overview

When the admin sends a message to an account, the sender displayed to the recipient is the admin.

When the admin specifies an account to send a message to another account, the sender displayed to the recipient is not the admin, but the account specified by the admin.

This API does not check whether the sender and the recipients are friends or blocklisted by either party or whether the senders are muted.

This API does not check whether the recipient has muted notifications for messages from the sender by default. If needed, enter `WithMuteNotifications` in the `SendMsgControl` parameter.

For one-to-one messages, the `MsgSeq` field is specified by the user when the message is sent. Its value can be repeated. It is not generated by the backend and not globally unique. For group chat messages, the value of the `MsgSeq` field is generated by the backend, and each group maintains its respective `MsgSeq` which increases strictly from 1. For one-to-one messages, the historical messages of the same conversation are sorted by timestamp first, and then the historical messages of the same second are sorted by `MsgSeq`.

### Caution

When calling this API to send a one-to-one message, you must specify whether to synchronize the message to the sender, which is the admin account or the account specified by the admin. Synchronization can be implemented via online terminals and roaming servers. This API provides the `SyncOtherMachine` parameter to determine whether to synchronize the message. For more information, see **Sample requests** below.

## Sample request URL

```
https://xxxxxx/v4/openim/sendmsg?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim/sendmsg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Here, we use sending a text message as an example. To send messages of other types, set `MsgBody` to the corresponding message type. For more information, see [Message Formats](#).

### The admin sends a message to another account.

#### Caution

If you do not want to synchronize the message to `From_Account`, set `SyncOtherMachine` to `2`. To synchronize the message to `From_Account`, set `SyncOtherMachine` to `1`.

```
{
  "SyncOtherMachine": 2, // Do not synchronize the message to the sender.
  "To_Account": "lumotuwe2",
  "MsgSeq": 93847636,
  "MsgRandom": 1287657,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ]
}
```

```

    }
  ],
  "CloudCustomData": "your cloud custom data",
  "SupportMessageExtension": 0
}

```

The admin sends a message to another account and forbids callbacks for the message.

### Caution

If you do not want to synchronize the message to `From_Account`, set `SyncOtherMachine` to `2`. To synchronize the message to `From_Account`, set `SyncOtherMachine` to `1`.

```

{
  "SyncOtherMachine": 2, // Do not synchronize the message to the sender.
  "To_Account": "lumotuwe2",
  "MsgSeq": 93847636,
  "MsgRandom": 1287657,
  "ForbidCallbackControl": [
    "ForbidBeforeSendMsgCallback",
    "ForbidAfterSendMsgCallback"], // Callback forbidding control option
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ],
  "CloudCustomData": "your cloud custom data"
}

```

The admin specifies an account to send a message to another account and set the information of offline push, without synchronizing the message to `From_Account`.

### Caution

If you do not want to synchronize the message to `From_Account`, set `SyncOtherMachine` to `2`.

```

{
  "SyncOtherMachine": 2,
  "From_Account": "lumotuwe1",
  "To_Account": "lumotuwe2",
  "MsgSeq": 93847636,
  "MsgRandom": 1287657,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",

```



```

        "MsgContent": {
            "Text": "hi, beauty"
        }
    },
    "CloudCustomData": "your cloud custom data",
    "OfflinePushInfo": {
        "PushFlag": 0,
        "Desc": "Content to push offline",
        "Ext": "Passthrough content",
        "AndroidInfo": {
            "Sound": "android.mp3"
        },
        "ApnsInfo": {
            "Sound": "apns.mp3",
            "BadgeMode": 1, // If this field is left as default or is set to
`0`, the message is counted. If this field is set to `1`, the message is not
counted, that is, the badge counter in the upper-right corner does not
increase.
            "Title": "apns title", // APNs title
            "SubTitle": "apns subtitle", // APNs subtitle
            "Image": "www.image.com" // Image URL
        }
    }
}

```

The admin specifies an account to send a message to another account, and specifies that the message will only be sent to online users, and no historical chat records will be stored (set `OnlineOnlyFlag` as 1).

### Caution

To synchronize the message to `From_Account`, set `SyncOtherMachine` to 1.

```

{
    "SyncOtherMachine": 1, // Synchronize the message to the sender.
    "From_Account": "lumotuwei1",
    "To_Account": "lumotuwei2",
    "OnlineOnlyFlag": 1,
    "MsgSeq": 93847636,
    "MsgRandom": 1287657,
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ]
}

```

```

    ],
    "CloudCustomData": "your cloud custom data"
  }

```

## Request fields

Field	Type	Required	Description
SyncOtherMachine	Integer	No	<p>0 : synchronize the message to the <code>From_Account</code> terminal and roaming server.</p> <p>2 : do not synchronize the message to <code>From_Account</code></p> <p>3 : do not synchronize the message to <code>To_Account</code>.</p> <p>If this field is not specified, the message will be synchronize <code>From_Account</code> roaming server.</p>
From_Account	String	No	<code>UserID</code> of the sender (used to specify the message sender)
To_Account	String	Yes	<code>UserID</code> of the recipient
OnlineOnlyFlag	Integer	No	<p>If this field is set to 0, the message will only be sent to the recipient online and not retained offline.</p> <p>1 means that messages will not be stored in the historical records, meaning if the recipient is online, they will receive the message; if the recipient is not online, they will not receive the message. This is suitable for implementing some real-time features, such as the "typing" feature.</p>
MsgSeq	Integer	No	Sequence number (32-bit unsigned integer) of the message. The backend will use this field to remove duplicate messages at the same time as messages in the same second. For details, see <b>Feature Description</b> . If this field is left empty, the backend will generate a random number.
MsgRandom	Integer	Yes	Message random number (32-bit unsigned integer), used by the backend for deduplication of messages within the same second. Please ensure this field is filled with a random number.
ForbidCallbackControl	Array	No	<p>Message callback forbidding field, which is valid only for this message. <code>ForbidBeforeSendMsgCallback</code> forbids callback before sending the message.</p> <p><code>ForbidAfterSendMsgCallback</code> forbids the callback after sending the message.</p>
SendMsgControl	Array	No	<p>Message sending control options is a String array, which is valid only for this message.</p> <p>"NoUnread" means this message is not counted as unread</p>

			<p>"NoLastMsg" means this message does not update the conversation List.</p> <p>"WithMuteNotifications" means for this message, the Do Not Disturb Option set by the recipient for the sender takes effect (does not take effect by default).</p> <p>"NoMsgCheck" means that when cloud-based review is enabled, this message will not be submitted for review.</p> <p>Example:</p> <p>"SendMsgControl":</p> <p>["NoUnread", "NoLastMsg", "WithMuteNotifications", "NoMsgCheck"]</p>
MsgBody	Array	Yes	<p>Message body. For details on formats, please see <a href="#">Message Formats</a>. (Note: a message can contain multiple message elements, in which case <code>MsgBody</code> is an array.)</p>
MsgType	String	Yes	<p>TIM message object type. Valid values:</p> <ul style="list-style-type: none"> <li><code>TIMTextElem</code> (text message)</li> <li><code>TIMLocationElem</code> (location message)</li> <li><code>TIMFaceElem</code> (emoji message)</li> <li><code>TIMCustomElem</code> (custom message)</li> <li><code>TIMSoundElem</code> (voice message)</li> <li><code>TIMImageElem</code> (image message)</li> <li><code>TIMFileElem</code> (file message)</li> <li><code>TIMVideoFileElem</code> (video message)</li> </ul>
MsgContent	Object	Yes	<p>Different message object types ( <code>MsgType</code> ) have different formats ( <code>MsgContent</code> ). For details, see <a href="#">Message Format</a>.</p>
CloudCustomData	String	No	<p>Custom message data. It is saved in the cloud and will be sent to the peer end. Such data can be pulled after the app is uninstalled and reinstalled.</p>
SupportMessageExtension	Integer	No	<p>Whether the message supports message extension. <code>0</code> : No. <code>1</code> : Yes</p>
OfflinePushInfo	Object	No	<p>The information to be pushed offline. For more information, see <a href="#">Message Formats</a>.</p>
IsNeedReadReceipt	Integer	No	<p>Whether the message requires a read receipt. <code>0</code> : No. <code>1</code> : Yes. The default value is <code>0</code>.</p>

## Sample response

Response to a successful request

```
{
```

```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0,
"MsgTime": 1572870301,
"MsgKey": "89541_2574206_1572870301",
"MsgId": "144115233406643804-1727580296-4026038328"
}
```

#### Response to a failed request

```
{
  "ActionStatus": "FAIL",
  "ErrorInfo": "Fail to Parse json data of body, Please check it",
  "ErrorCode": 90001
}
```

### Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MsgTime	Integer	Message timestamp in the UNIX format
MsgKey	String	Unique identifier of the message. This field is required to recall a message. The value is a string of no more than 50 characters.
MsgId	String	Unique identifier of the message on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
20001	Invalid request.
20002	<code>UserSig</code> or <code>A2</code> has expired.

20003	The <code>UserID</code> of the sender or recipient is invalid or does not exist. Make sure that the <code>UserID</code> has been imported into Chat.
20004	Network exception. Try again.
20005	Internal server error. Try again.
20006	The callback before sending a one-to-one message was triggered, and the app backend returned a response to forbid delivering the message.
90001	Failed to parse the JSON request. Make sure the format is valid.
90002	The <code>MsgBody</code> in the JSON request does not meet message format requirements or <code>MsgBody</code> is not an array. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90003	The JSON request does not contain the <code>To_Account</code> field or the <code>To_Account</code> field is not a string.
90005	The JSON request does not contain the <code>MsgRandom</code> field or the <code>MsgRandom</code> field is not an integer.
90007	The <code>MsgBody</code> field in the JSON request is not an array. Change it to an array.
90009	The request requires app admin permissions.
90010	The JSON request does not meet message format requirements. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90012	The account specified in <code>To_Account</code> does not exist or is not registered. Make sure the account has been imported to Chat and is correct.
90026	The offline retention time of the message is incorrect. Messages cannot be retained offline for more than 7 days.
90031	The <code>SyncOtherMachine</code> field in the JSON request is not an integer.
90044	The <code>MsgLifeTime</code> field in the JSON request is not an integer.
91000	Internal service error. Try again.
90992	Internal service error. Try again. If this error code is returned for all requests and third-party callback is enabled, make sure the app server returns the callback results to the Chat backend normally.
93000	The JSON packet exceeds the maximum size of 12 KB.
90048	The requested account does not exist.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

Sending One-to-One Messages to Multiple Users ([v4/openim/batchsendmsg](#))

Querying One-to-One messages ([v4/openim/admin\\_getroammsg](#))

Recalling One-to-One Messages ([v4/openim/admin\\_msgwithdraw](#))

## Possible Webhooks

[Before a One-to-One Message Is Sent](#)

[After a One-to-One Message Is Sent](#)

# Sending One-to-One Messages to Multiple Users

Last updated : 2024-12-19 15:00:28

## Feature Overview

This API is used to send one-to-one messages to multiple users (up to 500) at a time.

Compared with the API for sending one-to-one messages to one user, this API is more suitable for time-sensitive messages, such as marketing messages and system notifications.

If you don't want to count messages as unread or store them, you can set `MsgLifeTime` to `1`. This will make message delivery faster.

When the admin specifies an account to send a message to multiple target accounts, the sender displayed to the recipients is not the admin, but the account specified by the admin.

This API does not trigger callback requests.

This API does not check whether the sender and the recipients are friends or blocklisted by either party or whether the senders are muted.

This API does not check whether the recipient has muted notifications for messages from the sender by default. If needed, enter `WithMuteNotifications` in the `SendMsgControl` parameter.

For one-to-one messages, the `MsgSeq` field is specified by the user when the message is sent. Its value can be repeated. It is not generated by the backend and not globally unique. For group chat messages, the value of the `MsgSeq` field is generated by the backend, and each group maintains its respective `MsgSeq` which increases strictly from 1. For one-to-one messages, the historical messages of the same conversation are sorted by timestamp first, and then the historical messages of the same second are sorted by `MsgSeq`.

### Note :

When calling this API to batch send a message, you must specify whether to synchronize the message to the sender, which is the admin account or the account specified by the admin. Synchronization can be implemented via online terminals and roaming servers. This API provides the `SyncOtherMachine` parameter to determine whether to synchronize the message. For more information, please see **Sample request** below.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/openim/batchsendmsg?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim/batchsendmsg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

12,000 messages/minute. If sent to 500 users at one time, it will be counted as 500 messages.

## Sample request

Here, we use sending a text message as an example. To send messages of other types, set `MsgBody` to the corresponding message type. For more information, see [Message Formats](#).

### The admin sends a message to multiple target accounts.

#### Note :

If you do not want to synchronize the message to `From_Account`, set `SyncOtherMachine` to `2`.  
To synchronize the message to `From_Account`, set `SyncOtherMachine` to `1`.

```
{
```



```

"SyncOtherMachine": 2, // Do not synchronize the message to the sender.
"To_Account": [ // A list of target accounts
    "bonnie",
    "rong"
],
"MsgSeq": 28360, // Sequence number of the message.
"MsgRandom": 19901224, // Random number of the message
"MsgBody": [ // Message body
    {
        "MsgType": "TIMTextElem", // Message type. `TIMTextElem` indicates
text messages.
        "MsgContent": {
            "Text": "hi, beauty" // Message text
        }
    }
],
"CloudCustomData": "your cloud custom data"
}

```

**The admin specifies an account to send a message to multiple target accounts and sets the information of offline push.**

`From_Account` is the sender specified by the admin. The sender displayed to the recipients is not the admin, but the account specified by the admin. In the following JSON request, dave sends a message to bonnie and rong. When bonnie and rong receive the message, the message sender displayed to them is dave.

#### Note :

If you do not want to synchronize the message to `From_Account`, set `SyncOtherMachine` to `2`.

To synchronize the message to `From_Account`, set `SyncOtherMachine` to `1`.

```

{
    "SyncOtherMachine": 1, // Synchronize the message to the sender.
    "From_Account": "dave",
    "To_Account": [
        "bonnie",
        "rong"
    ],
    "MsgSeq": 28360, // Sequence number of the message.
    "MsgRandom": 19901224, // Random number of the message
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ]
}

```

```

    ],
    "CloudCustomData": "your cloud custom data",
    "OfflinePushInfo": {
        "PushFlag": 0,
        "Desc": "Content to push offline",
        "Ext": "Passthrough content",
        "AndroidInfo": {
            "Sound": "android.mp3"
        },
        "ApnsInfo": {
            "Sound": "apns.mp3",
            "BadgeMode": 1, // If this field is left as default or is set to
`0`, the message is counted. If this field is set to `1`, the message is not
counted, that is, the badge counter in the upper-right corner does not
increase.
            "Title": "apns title", // APNs title
            "SubTitle": "apns subtitle", // APNs subtitle
            "Image": "www.image.com" // Image URL
        }
    }
}

```

## Request fields

Field	Type	Required	Description
SyncOtherMachine	Integer	No	<p>1 : synchronize the message to the online terminal and roaming server of <code>From_Account</code> .</p> <p>2 : do not synchronize the message to <code>From_Account</code> . If this field is not specified, the message is synchronized to the roaming server of <code>From_Account</code> .</p>
From_Account	String	No	Sender account specified by the admin. To set the information of <code>From_Account</code> , the value of this field cannot be left empty.
To_Account	Array	Yes	<code>UserID</code> of the message recipient
MsgSeq	Integer	No	Sequence number (32-bit unsigned integer) of the message. The backend will use this field to remove duplicate messages and sort messages in the same second. For details, see <b>Feature Description</b> . If this field is left empty, the backend will enter a random number.
MsgRandom	Integer	Yes	Random number (32-bit unsigned integer) of the message. It is used by the backend for removing duplicate message within a second. Make sure a random number is entered.

MsgBody	Array	Yes	TIM message. For more information, see <a href="#">Message Formats</a> .
OnlineOnlyFlag	Integer	No	<p>If this field is set to <code>0</code>, the message will be stored in the historical chat records.</p> <p><code>1</code> means that messages will not be stored in the historical chat records, meaning if the recipient is online, they will receive this message; if offline, they won't. Suitable for implementing some real time status features.</p>
MsgType	String	Yes	<p>TIM message object type. Valid values:</p> <ul style="list-style-type: none"> <li><code>TIMTextElem</code> (text message)</li> <li><code>TIMLocationElem</code> (location message)</li> <li><code>TIMFaceElem</code> (emoji message)</li> <li><code>TIMCustomElem</code> (custom message)</li> <li><code>TIMSoundElem</code> (voice message)</li> <li><code>TIMImageElem</code> (image message)</li> <li><code>TIMFileElem</code> (file message)</li> <li><code>TIMVideoFileElem</code> (video message)</li> </ul>
MsgContent	Object	Yes	TIM message object. For more information, see <a href="#">Message Formats</a> .
CloudCustomData	String	No	Custom message data. It is saved in the cloud and will be sent to the receiver. Such data can be pulled after the app is uninstalled and reinstalled.
SendMsgControl	Array	No	<p>Message sending control options is an array of strings, which is valid only for this request.</p> <p>"NoUnread" means this message is not counted as unread messages.</p> <p>"NoLastMsg" indicates that this message does not update the conversation list.</p> <p>"WithMuteNotifications" means that the recipient's Do Not Disturb settings for the sender are effective for this message (it does not take effect by default).</p> <p>"NoMsgCheck" means that, when cloud-based review is enabled, this message will not be submitted for review.</p> <p>Example:</p> <p>"SendMsgControl":</p> <p><code>["NoUnread", "NoLastMsg", "WithMuteNotifications", "NoMsgCheck"</code></p>
OfflinePushInfo	Object	No	The information to be pushed offline. For more information, see <a href="#">Message Formats</a> .
IsNeedReadReceipt	Integer	No	<p>Whether the message requires a read receipt.</p> <p><code>0</code> : No.</p> <p><code>1</code> : Yes.</p>

The default value is 0 .

Sample response

Response when the message was sent to all the target accounts

```
{
  "ErrorInfo": "",
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "MsgKey": "128493_903762_1572870301",
  "MsgId": "144015217074523023-1734398005-1067690400"
}
```

Response when the message was not sent to some target accounts

```
{
  "ActionStatus": "SomeError",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "MsgKey": "4852_28135_1579678877",
  "MsgId": "144015217074523023-1734398005-1067690400",
  "ErrorList": [ // List of the accounts to which the message was not sent
    {
      "To_Account": "rong", // Account to which the message was not sent
      "ErrorCode": 70107 // Error code. `70107` indicates that the account
does not exist.
    }
  ]
}
```

Response when the message was sent to none of the target accounts

```
{
  "ActionStatus": "FAIL",
  "ErrorInfo": "invalid To_Account",
  "ErrorCode": 90012
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. OK : successful; FAIL : failed
ErrorCode	Integer	Error code returned for the request.

		<p>If the message was sent to any account, the value is <code>0</code> .</p> <p>If the message was sent to none of the target accounts, the value is not <code>0</code> .</p>
ErrorInfo	String	Error information
ErrorList	Array	List of the target accounts to which the message was not sent or that do not exist. If the message was sent to all the target accounts, the value of this field is empty.
ErrorList.To_Account	String	Target account to which the message was not sent.
ErrorList.ErrorCode	Integer	Error code indicating that the message was not sent. If the error code is 70107, the account does not exist.
MsgKey	String	Unique identifier of the message. This field is required to recall a message. The value is a string of no more than 50 characters.
MsgId	String	Unique identifier of the message on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
70107	The requested account does not exist.
70169	Server timeout. Try again later.
90001	Failed to parse the JSON request. Make sure the format is valid.
90002	The <code>MsgBody</code> in the JSON request does not meet message format requirements or <code>MsgBody</code> is not an array. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90004	Invalid <code>MsgSeq</code> field in the JSON request.
90007	The <code>MsgBody</code> field in the JSON request is not an array. Change it to an array.
90008	The JSON request does not contain the <code>From_Account</code> field or the account specified in

	<code>From_Account</code> does not exist.
90009	The request requires app admin permissions.
90010	The JSON request does not meet message format requirements. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90011	The number of target accounts exceeds 500. Delete some <code>To_Account</code> .
90012	The account specified in <code>To_Account</code> does not exist or is not registered. Make sure the account has been imported to Chat and is correct.
90026	The offline retention time of the message is incorrect. Messages cannot be retained offline for more than 7 days.
90048	The requested account does not exist.
90992	Internal service error. Try again. If this error code is returned for all requests and webhook is enabled, make sure the app server returns the results to the Chat backend normally.
91000	Internal service error. Try again.
93000	The JSON packet exceeds the maximum size of 12 KB.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Sending One-to-One Messages to One User](#) (v4/openim/sendmsg)

[Sending Ordinary Messages in a Group](#) (v4/group\_open\_http\_svc/send\_group\_msg)

[Sending System Messages in a Group](#) (v4/group\_open\_http\_svc/send\_group\_system\_notification)

[Message Formats](#)

# Sending Ordinary Messages in a Group

Last updated : 2025-01-14 11:59:12

## Feature Overview

This API is used by the app admin to send ordinary messages in a group.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/send_group_msg?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/send_group_msg</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

The app admin sends ordinary group messages, and the sender is the app admin.

```
{
  "GroupId": "@TGS#2C5SZEAEF",
  "Random": 8912345, // A random number. If the random numbers of two messages
  are the same within five minutes, they are considered to be the same message.
  "MsgBody": [ // Message body, which consists of an element array. For
  details, see the field description.
    {
      "MsgType": "TIMTextElem", // Text
      "MsgContent": {
        "Text": "red packet"
      }
    },
    {
      "MsgType": "TIMFaceElem", // Emoji
      "MsgContent": {
        "Index": 6,
        "Data": "abc\\u0000\\u0001"
      }
    }
  ]
}
```



```

],
"CloudCustomData": "your cloud custom data",
"SupportMessageExtension": 0,
"OfflinePushInfo": {
  "PushFlag": 0, // Normal push
  "Desc": "Content to push offline",
  "Ext": "Passthrough content",
  "AndroidInfo": {
    "Sound": "android.mp3"
  },
  "ApnsInfo": {
    "Sound": "apns.mp3",
    "BadgeMode": 1, // If this field is left as default or is set to `0`,
the message is counted. If this field is set to `1`, the message is not
counted, that is, the icon number in the upper-right corner does not increase.
    "Title": "apns title", // APNs title
    "SubTitle": "apns subtitle", // APNs subtitle
    "Image": "www.image.com" // Image URL
  }
}
}
}

```

### Specifying the message sender

The app admin can specify a group member as the message sender in `From_Account` .

After receiving the message, other members will see that the message is sent from the group member specified by the app admin.

```

{
  "GroupId": "@TGS#2C5SZEAEF",
  "From_Account": "leckie", // Message sender (optional)
  "Random": 8912345, // A random number. If the random numbers of two messages
are the same within five minutes, they are considered to be the same message.
  "MsgBody": [ // Message body, which consists of an element array. For
details, see the field description.
    {
      "MsgType": "TIMTextElem", // Text
      "MsgContent": {
        "Text": "red packet"
      }
    },
    {
      "MsgType": "TIMFaceElem", // Emoji
      "MsgContent": {
        "Index": 6,
        "Data": "abc\\u0000\\u0001"
      }
    }
  ]
}

```

```

    }
  ]
}
```

### Specifying the message recipient

If `To_Account` is added to the message body, and a message recipient is specified in it, the message will be sent only to the specified recipient, and the message is excluded from the unread count. (Supported group types: Private, Public, ChatRoom)

#### Note

This feature is supported only in the Pro edition、Pro Plus edition、Enterprise edition.

```

{
  "GroupId": "@TGS#12DEVUDHQ",
  "Random": 2784275388,
  "MsgBody": [
    {
      "MsgType": "TIMCustomElem",
      "MsgContent": {
        "Data": "1cdddddddddq1"
      }
    }
  ],
  "To_Account": ["brennanli2", "brennanli3"] // Specify the message recipient
  (up to 50 recipients can be specified). If this field is used, the message is
  excluded from the unread count.
}
```

### Specifying that messages do not trigger conversation update

If `SendMsgControl` is set to `NoLastMsg`, it indicates not to trigger conversation update; if it is set to `NoUnread`, it indicates not to include the message in the unread count (this field cannot be used for audio-video groups (AVChatRoom). For meeting groups (Meeting), messages are excluded from the unread count by default, and you need to [enable the feature](#); otherwise, an error will be reported when `NoUnread` is set).

```

{
  "GroupId": "@TGS#2C5SZEAEF",
  "Random": 8912345, // A random number. If the random numbers of two messages
  are the same within five minutes, they are considered to be the same message.
  "SendMsgControl": ["NoLastMsg"], // Do not trigger conversation update.
  "MsgBody": [ // Message body, which consists of an element array. For
  details, see the field description.
    {
      "MsgType": "TIMTextElem", // Text
      "MsgContent": {
        "Text": "red packet"
      }
    }
  ]
}
```

```

    },
    {
      "MsgType": "TIMFaceElem", // Emoji
      "MsgContent": {
        "Index": 6,
        "Data": "abc\\u0000\\u0001"
      }
    }
  ]
}

```

### Specifying the message priority

You can specify the message priority. The default priority is `Normal`.

There are three priority options in descending order: `High`, `Normal`, and `Low`. They are case-sensitive.

```

{
  "GroupId": "@TGS#2C5SZEAEF",
  "Random": 8912345, // A random number. If the random numbers of two messages
are the same within five minutes, they are considered to be the same message.
  "MsgPriority": "High", // Message priority
  "MsgBody": [ // Message body, which consists of an element array. For
details, see the field description.
    {
      "MsgType": "TIMTextElem", // Text
      "MsgContent": {
        "Text": "red packet"
      }
    },
    {
      "MsgType": "TIMFaceElem", // Emoji
      "MsgContent": {
        "Index": 6,
        "Data": "abc\\u0000\\u0001"
      }
    }
  ]
}

```

### Forbidding callback for a message

When the callback switch is turned on, users can specify `ForbidCallbackControl` to control whether to initiate callback for a single message. By default, callback is initiated.

```

{
  "GroupId": "@TGS#2C5SZEAEF",
  "Random": 8912345, // A random number. If the random numbers of two messages
are the same within five minutes, they are considered to be the same message.

```

```

"ForbidCallbackControl":[
    "ForbidBeforeSendMsgCallback",
    "ForbidAfterSendMsgCallback"], // Callback forbidding control option
"MsgBody": [ // Message body, which consists of an element array. For
details, see the field description.
    {
        "MsgType": "TIMTextElem", // Text
        "MsgContent": {
            "Text": "red packet"
        }
    },
    {
        "MsgType": "TIMFaceElem", // Emoji
        "MsgContent": {
            "Index": 6,
            "Data": "abc\\u0000\\u0001"
        }
    }
]
}

```

### Sending group @ messages

The mentioned users (@) set in the `GroupAtInfo` field have a one-to-one and sequential correspondence to the mentioned @ users in the message body.

### Caution

Audio-video groups (AVChatroom) don't support the @all feature.

```

{
    "GroupId": "@TGS#2C5SZEAEF",
    "Random": 8912345, // A random number. If the random numbers of two
messages are the same within five minutes, they are considered to be the same
message.
    "MsgBody": [ // Message body, which consists of an element array. For
details, see the field description.
        {
            "MsgType": "TIMTextElem", // Text
            "MsgContent": {
                "Text": "red @all @tommy @brennanli packet"
            }
        }
    ],
    // It corresponds to @all @tommy @brennanli in the text information.
    "GroupAtInfo":[
        {
            "GroupAtAllFlag":1 // `1`: @ all ; `0`: @ a certain group member
        },
    ]
}

```

```
{
  "GroupAtAllFlag":0,
  "GroupAt_Account":"tommy" // @ a specific group member
},
{
  "GroupAtAllFlag":0,
  "GroupAt_Account":"brennanli"
}
]
```

### Specifying messages for online delivery without offline or roaming retention

If **OnlineOnlyFlag** in the message body is set to a value greater than 0, the message is for online delivery only, not for offline or roaming retention (not available for AVChatRoom or BChatRoom).

```
{
  "GroupId": "@TGS#2C5SZEAEF",
  "Random": 8912345, // A random number. If the random numbers of two
  messages are the same within five minutes, they are considered to be the same
  message.
  "OnlineOnlyFlag": 1, // The message is for online delivery only (only
  online group members will receive it), not for offline or roaming retention.
  "MsgBody": [ // Message body, which consists of an element array. For
  details, see the field description.
    {
      "MsgType": "TIMTextElem", // Text
      "MsgContent": {
        "Text": "red packet"
      }
    },
    {
      "MsgType": "TIMFaceElem", // Emoji
      "MsgContent": {
        "Index": 6,
        "Data": "abc\\u0000\\u0001"
      }
    }
  ]
}
```

### Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group to which the message will be sent

Random	Integer	Yes	A 32-bit unsigned integer. If the content and random numbers of two messages within five minutes are the same, the later message will be discarded as a repeated message.
MsgPriority	String	No	Message priority
MsgBody	Array	Yes	Message body. For more information, see <a href="#">Message Formats</a> .
From_Account	String	No	Message source account. If this field is not specified, the message sender is the app admin account used to call the API. Alternatively, apps can specify the message sender in this field to implement some special features. Note that if this field is specified, you must ensure that the account in this field exists.
OfflinePushInfo	Object	No	Information of offline push. For more information, see <a href="#">Message Formats</a> .
ForbidCallbackControl	Array	No	Message callback forbidding option, valid only for a single message. <code>ForbidBeforeSendMsgCallback</code> : callback before sending the message is forbidden; <code>ForbidAfterSendMsgCallback</code> : callback after sending the message is forbidden.
OnlineOnlyFlag	Integer	No	<code>1</code> : send to online members only; <code>0</code> (default value): send to all members. This field is not valid for audio-video groups (AVChatRoom).
SendMsgControl	Array	No	Message sending permission, only valid for the current message. <code>NoLastMsg</code> : do not trigger conversation update; <code>NoUnread</code> : do not include the message in the unread count. (If <code>OnlineOnlyFlag</code> is set to <code>1</code> for the message, this field cannot be used.)
CloudCustomData	String	No	Custom message data. It is saved in the cloud and will be sent to the receiver. Such data can be pulled after the app is uninstalled and reinstalled.
SupportMessageExtension	Integer	No	Whether the message supports message extension. <code>0</code> : No. <code>1</code> : Yes
To_Account	Array	No	Specify up to 50 message receivers. If this field is used, the message will be excluded from the unread count. It is available only in the Pro edition 、 Pro Plus edition、

			Enterprise edition for work groups (Work), public groups (Public), and meeting groups (Meeting).
TopicId	String	No	Topic ID, which indicates sending ordinary messages in the topic and applies only to topic-enabled communities.

Sample responses

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "MsgTime": 1497249503,
  "MsgSeq": 1,
  "MsgDropReason" : "MsgFreqCtrl"
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MsgTime	Integer	Message sending timestamp, corresponding to the backend server time
MsgSeq	Integer	Message sequence number, the unique identifier of a message
MsgDropReason	String	Reason for discarding the message. It is empty by default. Currently, only messages that exceed the frequency limit are discarded.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
------------	-------------

10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. This error occurs when, for example, a member in a public group tries to remove other users from the group (only the app admin can perform this operation).
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use a correct group ID.
10016	The app backend rejected this operation through a third-party callback.
10017	The message cannot be sent due to muting. Check whether the sender is muted.
10023	The frequency limit for message sending is reached. Try again later.
80002	The message content is too long. Currently, the maximum message length supported is 12 KB. Please adjust the message length.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Sending System Messages in a Group ([v4/group\\_open\\_http\\_svc/send\\_group\\_system\\_notification](#))

[Sending One-to-One Messages to One User](#) ([v4/openim/sendmsg](#))

[Sending One-to-One Messages to Multiple Users](#) ([v4/openim/batchsendmsg](#))

[Message Formats](#)

## Possible Callbacks

[After a Group Message Is Sent](#)



# Sending System Messages in a Group

Last updated : 2024-07-16 10:41:26

## Feature Overview

This API is used by app admins to push system notifications to group chats.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes. But only to all group members.
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

A non-audio-video group supports sending system notifications to specified group members, but an audio-video group (AVChatRoom) only supports sending system notifications to all group members.

For more information on the API ( `V2TIMGroupListener.onReceiveRESTCustomData` ) for clients to receive system notifications, see [Android](#) or [iOS](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/send_group_system_notification?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/send_group_system_notification	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

This example pushes a system notification to all members in a group.

```
{
  "GroupId": "@TGS#2C5SZEAEF",
  "Content": "Hello World" // Content of the notification
}
```

### To specific members

To specify who can receive the system notification, set the recipients in `ToMembers_Account`. Audio-video groups (AVChatRoom) only support sending system notifications to all group members, but not specified group members.

```
{
  "GroupId": "@TGS#2C5SZEAEF",
  "ToMembers_Account": [ // List of recipients. You can leave this field empty
to send the notification to all members.
    "peter",
    "leckie"
  ],
  "Content": "Hello World" // Content of the notification
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group to which the notification will be sent
ToMembers_Account	Array	Yes	List of recipients. Up to 500 recipient UserIDs are supported. You can leave this field empty to send the notification to all members.
Content	String	Yes	Content of the notification
TopicId	String	No	ID of the topic where system notices are sent. This field applies only to topic-enabled community groups.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use our [RESTful API Tester](#) to test your requests.

## References

Sending Ordinary Messages in a Group ([v4/group\\_open\\_http\\_svc/send\\_group\\_msg](#))

# Broadcast Message of Audio-Video Group

Last updated : 2025-01-14 11:59:12

## Feature Overview

This API is used to deliver broadcast messages to all the audio-video groups.

### Note

This feature is supported by the SDK of the Enhanced edition on v6.5.2803 or later and the SDK for web on v2.21.0 or later. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#)、 [Enterprise edition](#), go to the [console](#), select **Feature configuration > Group configuration > Group feature configuration**, and enable **Broadcast messaging of audio-video group**.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	No. Same as work groups (Work) in the new version.
Public	No
ChatRoom	No. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes. Messages are sent to all the audio-video groups.
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/send_broadcast_msg?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/send_broadcast_msg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

1 times/second

## Sample request

### Basic format

It is used to deliver broadcast messages to all the audio-video groups.

```
{
  "From_Account": "test", // Specify the message sender (optional)
  "Random": 8912345, // A random number. If the random numbers of two
  messages are the same within five minutes, they are considered to be the same
  message.
  "MsgBody": [
    {
      "MsgType": "TIMCustomElem", // Custom message
      "MsgContent": {
        "Data": "{ \\\"type\\\":1, \\\"content\\\":\\\"hello world\\\"}"
      }
    }
  ]
}
```

```
    }
  ],
  "CloudCustomData": "your cloud custom data"
}
```

**Caution**

`MsgBody` supports multiple message elements. If you need to call it more than once every second, the business can merge messages into one of up to 12,000 bytes.

**Request fields**

Field	Type	Required	Description
From_Account	String	No	Message source account. If this field is not specified, the message sender is the app admin account used to call the API. Alternatively, apps can specify the message sender in this field to implement some special features. Note that if this field is specified, you must ensure that the account in this field exists.
Random	Integer	Yes	A 32-bit unsigned integer. If the content and random numbers of two messages within five minutes are the same, the later message will be discarded as a repeated message.
MsgBody	Array	Yes	Message body. For more information, see <a href="#">Message Formats</a> .
CloudCustomData	String	No	Custom message data

**Sample response**

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "MsgSeq": 1283
}
```

**Response fields**

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

MsgSeq	Integer	Message sequence number, the unique identifier of a message
--------	---------	---

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	Insufficient operation permissions. For example, the switch is not enabled in the console, or the operating account is not the root account.
10023	The frequency limit for message sending is reached. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Sending Ordinary Messages in a Group ([v4/group\\_open\\_http\\_svc/send\\_group\\_msg](#))

[Message Formats](#)



# Importing One-to-One Messages

Last updated : 2024-07-16 10:41:26

## Feature Overview

This API imports one-to-one chat history to Tencent Cloud Chat\*\* in chronological order\*\*.

It imports your messages from other instant messaging platforms to Tencent Cloud Chat **in chronological order**.

It updates a conversation.

It does not trigger a callback.

It deduplicates the imported messages in a one-to-one conversation based on the `MsgSeq` , `MsgRandom` , and `MsgTimeStamp` fields. **When the values of all these three fields match their counterparts, the two messages are considered duplicates, regardless of their contents.** In addition, if the `MsgSeq` , `MsgRandom` , and `MsgTimeStamp` fields of the two messages matches each other, but `from_account` and `to_account` are opposite, the two messages are also considered duplicates.

Imported messages will not be overwritten by the same messages from later imports.

For one-to-one messages, the `MsgSeq` field is specified by the user when the message is sent. Its value can be repeated. It is not generated by the backend and not globally unique. **For group chat messages, the value of the `MsgSeq` field is generated by the backend, and each group maintains its respective `MsgSeq` which increases strictly from 1. For one-to-one messages, the historical messages of the same conversation are sorted by timestamp first, and then the historical messages of the same second are sorted by `MsgSeq` .**

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/openim/importmsg?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code>

	Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/openim/importmsg</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Importing real-time messages

```
{
  "SyncFromOldSystem": 5, // Imports real-time messages, marks them as unread
  and pushes them to the client.
  "From_Account": "lumotuwe1", // Account of the sender
  "To_Account": "lumotuwe2", // Account of the recipient
  "MsgSeq": 827092, // Sequence number of the message
  "MsgRandom": 1287657, // A random number assigned to the message
  "MsgTimeStamp": 1556178721, // UNIX timestamp in seconds
  "MsgBody": [ // Message body. This is a text message
    {
      "MsgType": "TIMTextElem", // Text message element
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ],
  "CloudCustomData": "your cloud custom data"
}
```

### Importing historical messages

```

{
  "SyncFromOldSystem": 2, // Imports historical messages, marks them as read
and does not push them to the client.
  "From_Account": "lumotuwei1", // Account of the sender
  "To_Account": "lumotuwei2", // Account of the recipient
  "MsgSeq": 827092, // Sequence number of the message
  "MsgRandom": 1287657, // A random number assigned to the message
  "MsgTimeStamp": 1556178721, // UNIX timestamp in seconds
  "MsgBody": [ // Message body. This is a text message
    {
      "MsgType": "TIMTextElem", // Text message element
      "MsgContent": {
        "Text": "hi, beauty" // Message content
      }
    }
  ],
  "CloudCustomData": "your cloud custom data"
}

```

## Request fields

Field	Type	Required	Description
SyncFromOldSystem	Integer	Yes	Valid values: <code>2</code> and <code>5</code> . <code>2</code> : Imports historical messages, marks them as read and does not push them to the client. <code>5</code> : Imports real-time messages, marks them as unread and pushes them to the client.
From_Account	String	Yes	<code>UserID</code> of the sender, which is used to specify the message sender
To_Account	String	Yes	<code>UserID</code> of the recipient
MsgSeq	Integer	No	Sequence number (32-bit unsigned integer) of the message. The backend will use this field to remove duplicate messages and sort messages in the same second. For details, see <b>Feature Description</b> . If this field is left empty, the backend will enter a random number.
MsgRandom	Integer	Yes	Random number (32-bit unsigned integer) of the message. It is used by the backend for removing duplicate message within a second. Make sure a random number is entered.
MsgTimeStamp	Integer	Yes	UNIX timestamp in seconds. It marks the time when the message was sent and is used to remove duplicate

			messages. For details, see <b>Feature Description</b> .
MsgBody	Array	Yes	Message body. For details on formats, please see <a href="#">Message Formats</a> . (Note: a message can contain multiple message elements, in which case <code>MsgBody</code> is an array.)
MsgType	String	Yes	TIM message object type. Valid values: <code>TIMTextElem</code> (text message) <code>TIMLocationElem</code> (location message) <code>TIMFaceElem</code> (emoji message) <code>TIMCustomElem</code> (custom message) <code>TIMSoundElem</code> (voice message) <code>TIMImageElem</code> (image message) <code>TIMFileElem</code> (file message) <code>TIMVideoFileElem</code> (video message)
MsgContent	Object	Yes	Different message object types ( <code>MsgType</code> ) have different formats ( <code>MsgContent</code> ). For details, see <a href="#">Message Formats</a> .
CloudCustomData	String	No	Custom message data. It is saved in the cloud and will be sent to the peer end. Such data can be pulled after the app is uninstalled and reinstalled.

Sample responses

```
{
  "ActionStatus" : "OK",
  "ErrorInfo" : "",
  "ErrorCode" : 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON request. Make sure the format is valid.
90002	The <code>MsgBody</code> in the JSON request does not meet message format requirements or <code>MsgBody</code> is not an array. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90003	The <code>To_Account</code> field is missing in the JSON request or it is not a string.
90005	The <code>MsgRandom</code> field is missing in the JSON request or it is not an integer.
90006	The <code>MsgTimeStamp</code> field is missing in the JSON request or it is not an integer.
90007	The <code>MsgBody</code> field in the JSON request is not an array. Change it to an array.
90008	The <code>From_Account</code> field is missing in the JSON request or it is not an integer.
90009	The request requires app admin permissions.
90010	The JSON request does not meet message format requirements. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90011	The number of recipients exceeded 500. Try to reduce the number of accounts in <code>To_Account</code> .
90012	The account specified in <code>To_Account</code> does not exist or is not registered. Make sure the account has been imported to Chat and is correct.
90026	The offline retention time of the message is incorrect. Messages cannot be retained offline for more than 7 days.
90030	The <code>SyncFromOldSystem</code> field is missing in the JSON request or it is not an integer.
90048	The requested account does not exist.
90992	Internal service error. Try again. If this error code is returned for all requests and third-party callback is enabled, make sure the app server returns the callback results to the Chat backend normally.
91000	Internal service error. Try again.

93000

The JSON packet exceeds the maximum size of 12 KB.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing Group Messages ([v4/group\\_open\\_http\\_svc/import\\_group\\_msg](#))

# Importing Group Messages

Last updated : 2024-07-16 10:41:26

## Feature Overview

This API is used to import group messages without triggering callbacks or sending notifications.

When you are migrating your app from another instant messaging system to Tencent Cloud Chat, you can use this API to import group message data.

## API Calling Description

### Applicable group types

Group Type	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

Above are the built-in group types of the Chat service. For more information, see [Group System](#).

### Note

Audio-video groups (AVChatRoom) do not support importing group messages. If you attempt to import group messages for an audio-video group, error 10007 will be returned. Therefore, members of an audio-video group cannot view the messages sent before they join the group.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/import_group_msg?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/import_group_msg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

A single request can import up to seven group messages.

After messages are imported through this API, the unread message count for all members will become `0`. To retain the unread message count, you need to import group members or set the unread message count for members after importing all messages.

The messages must be imported in ascending order by timestamp, and the timestamps of imported messages must be earlier than the current time, and later than the group creation time and the creation time of the latest message in the group. Otherwise, the import will fail.

```
{  
  "GroupId": "@TGS#2C5SZEAEF",
```



"RecentContactFlag":1, // Means to trigger conversation update (This field is not supported by AVChatRoom groups.)

"MsgList": [

{

"From\_Account": "leckie", // Message sender

"SendTime":1620808101,

"Random": 8912345, // Random number of the message (optional)

"MsgBody": [ // Message body, which consists of an element array.

For details, see the `TIMMessage` message object.

{

"MsgType": "TIMTextElem", // Text

"MsgContent": {

"Text": "red packet"

}

},

{

"MsgType": "TIMFaceElem", // Emoji

"MsgContent": {

"Index": 6,

"Data": "abc\\u0000\\u0001"

}

}

]

},

{

"From\_Account": "peter", // Message sender

"SendTime":1620892821,

"MsgBody": [ // Message body, which consists of an element array.

For details, see the `TIMMessage` message object.

{

"MsgType": "TIMTextElem", // Text

"MsgContent": {

"Text": "red packet"

}

}

]

}

]

}

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group for which to import messages

RecentContactFlag	Integer	No	Whether to trigger conversation update. The value <code>1</code> means to trigger conversation update. Conversation update is not triggered by default. This field is not supported by audio-video groups (AVChatRoom) and community groups.
MsgList	Object	Yes	List of the messages to import
From_Account	String	Yes	Message sender
SendTime	Integer	Yes	Message sending time
Random	Integer	No	A 32-bit unsigned integer. If the random numbers of two messages within five minutes are the same, the later message will be discarded as a repeated message.
MsgBody	Array	Yes	TIM message. For more information, see the definition of <code>TIMMsgElement</code> in <a href="#">Message Formats</a> .
MsgType	String	Yes	TIM message object type. Valid values: <code>TIMTextElem</code> (text message), <code>TIMFaceElem</code> (emoji message), <code>TIMLocationElem</code> (location message), <code>TIMCustomElem</code> (custom message)
MsgContent	Object	Yes	TIM message object. For more information, see the definition of <code>TIMMsgElement</code> in <a href="#">Message Formats</a> .
TopicId	String	No	ID of the topic to which messages are imported. This field applies only to topic-enabled community groups.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ImportMsgResult": [
    {
      "MsgSeq": 1,
      "MsgTime": 1620808101,
      "Result": 0
    },
    {
      "MsgSeq": 2,
      "MsgTime": 1620892821,
      "Result": 0
    }
  ]
}
```

```
    ]
  }
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorInfo	String	Error information
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ImportMsgResult	Array	Message import result
Result	Integer	Result of each message <code>0</code> : The message was imported successfully. <code>10004</code> : The sending time of the message is invalid. <code>80002</code> : The message content exceeds the limit of 12,000 bytes. Please adjust the message size.
MsgTime	Integer	Message timestamp
MsgSeq	Integer	Message sequence number, the unique identifier of a message

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. This error occurs when, for example, a member in a public group tries to remove other users from the group (only the app admin can perform this operation).
10010	The group does not exist or has been deleted.
10015	The group ID is invalid. Use the correct group ID.
10020	The message content is too long. Currently, the maximum message length supported is 12 KB. Please

adjust the message length.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Setting the Unread Message Count of a Member ([v4/group\\_open\\_http\\_svc/set\\_unread\\_msg\\_num](#))

# Sending Messages in a Official Account

Last updated : 2024-07-16 10:41:26

## Feature Overview

App Administrators can use this api to send ordinary messages to all subscribers following the official account. The maximum sending frequency for a single official account is 1 message/sec, with up to 2 broadcast messages per hour.

If the content of two messages is identical within 5 minutes, the latter will be discarded as a duplicate message.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/official_account_open_http_svc/send_official_account_msg?
sdkappid=88888888&identifier=admin&usersig=xxx&random=999999999&contenttype=json
```

### Request parameters

The table below only lists the parameters modified when calling this API and their descriptions. For more details on the parameters, please refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain for the country/region where your SDKAppID is located: China: console.tim.qq.com Singapore : adminapisgp.im.qcloud.com Seoul : adminapikr.im.qcloud.com Frankfurt : adminapiger.im.qcloud.com Silicon Valley : adminapiusa.im.qcloud.com Jakarta : adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/send_official_account_msg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created

identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a>
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a>
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: <code>json</code>

### Maximum calling frequency

200 queries/sec.

### Sample request packets

#### Basic Form

Sending Regular Official Account Messages.

```
{
  "Official_Account": "@TOA#_2C5SZEAEF",
  "Random": 9875,
  "MsgBody": [ // Message body, consists of an element array, see field
description for details, messages with the same content within five minutes are
considered duplicates
    {
      "MsgType": "TIMTextElem", // Text
      "MsgContent": {
        "Text": "red packet"
      }
    },
    {
      "MsgType": "TIMFaceElem", // Expression
      "MsgContent": {
        "Index": 6,
        "Data": "abc\\u0000\\u0001"
      }
    }
  ],
  "CloudCustomData": "your cloud custom data",
  "OfflinePushInfo": {
    "PushFlag": 0, // Normal push,
    "Desc": "Offline Push Content",
    "Ext": "This is the transmitted content",
    "AndroidInfo": {
      "Sound": "android.mp3"
    }
  }
}
```

```

    },
    "ApnsInfo": {
        "Sound": "apns.mp3",
        "BadgeMode": 1, // This field is omitted or 0 means counting is
        // required, 1 means this message does not require counting, i.e., the number on
        // the upper right corner icon does not increase
        "Title": "apns title", // apns title
        "SubTitle": "apns subtitle", // apns subtitle
        "Image": "www.image.com" // image url
    }
}
}

```

### Specifies that the message does not update the recent contact session

If SendMsgControl is specified in the message, setting NoLastMsg means it does not update the recent contact session.

```

{
    "Official_Account": "@TOA#_2C5SZEAEF",
    "Random": 9875,
    "SendMsgControl": ["NoLastMsg"], // Specifies that the message does not
    // trigger an update to the recent contact session.
    "MsgBody": [ // Message body, consists of an element array, see field
    // description for details, messages with the same content within five minutes are
    // considered duplicates
        {
            "MsgType": "TIMTextElem", // Text
            "MsgContent": {
                "Text": "red packet"
            }
        },
        {
            "MsgType": "TIMFaceElem", // Expression
            "MsgContent": {
                "Index": 6,
                "Data": "abc\\u0000\\u0001"
            }
        }
    ]
}

```

### Specifies that the message does not trigger a callback

With callback function enabled, users can control whether a single message triggers a callback through ForbidCallbackControl, by default, a callback is initiated.

```

{

```

```

"Official_Account": "@TOA#_2C5SZEAEF",
"Random": 9875,
"ForbidCallbackControl": [
    "ForbidBeforeSendMsgCallback",
    "ForbidAfterSendMsgCallback"], // Prohibit callback control option
"MsgBody": [ // Message body, consists of an element array, see field
description for details, messages with the same content within five minutes are
considered duplicates
    {
        "MsgType": "TIMTextElem", // Text
        "MsgContent": {
            "Text": "red packet"
        }
    },
    {
        "MsgType": "TIMFaceElem", // Expression
        "MsgContent": {
            "Index": 6,
            "Data": "abc\\u0000\\u0001"
        }
    }
]
}

```

### Specifies that the message will not be stored offline or for roaming

If `OnlineOnlyFlag` is specified in the message body, as long as its value is greater than 0, it indicates that the message will only be sent online and will not be stored for offline or roaming.

```

{
    "Official_Account": "@TOA#_2C5SZEAEF",
    "Random": 9875,
    "OnlineOnlyFlag": 1, // Indicates that the message is only sent online
(only online subscribers can receive it), and is not stored for offline or
roaming
    "MsgBody": [ // Message body, consists of an element array, see field
description for details, messages with the same content within five minutes are
considered duplicates
        {
            "MsgType": "TIMTextElem", // Text
            "MsgContent": {
                "Text": "red packet"
            }
        },
        {
            "MsgType": "TIMFaceElem", // Expression
            "MsgContent": {
                "Index": 6,

```



```
        "Data": "abc\\u0000\\u0001"
      }
    }
  ]
}
```

Request packet fields

Field	Type	Attribute	Description
Official_Account	String	Mandatory	Official Account Users sending messages.
Random	Integer	Mandatory	Unsigned 32-bit integer (range: 0 to 4294967295). If the content and Random value of two messages are the same within 5 minutes, the latter message will be considered a duplicate and discarded.
MsgBody	Array	Mandatory	Message body, for details refer to <a href="#">Message Format</a> .
OfflinePushInfo	Object	Optional	Offline push notification configuration, for details see <a href="#">Message Format</a> .
ForbidCallbackControl	Array	Optional	Message callback prohibition switch, applies only to individual messages, ForbidBeforeSendMsgCallback indicates prohibiting callback before sending a message, ForbidAfterSendMsgCallback indicates prohibiting callback after sending a message
OnlineOnlyFlag	Integer	Optional	1 indicates the message is sent only to online subscribers, default 0 indicates sending to all subscribers
SendMsgControl	Array	Optional	Message sending permissions, NoLastMsg applies only to individual messages, indicates not updating the latest conversation (If the message's OnlineOnlyFlag is set to 1, then this field should not be used.)
CloudCustomData	String	Optional	Custom message data (stored in the cloud, will be sent to the peer, and can be retrieved even after the app is uninstalled and reinstalled).

Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
```

```
"MsgTime": 1497249503,  
"MsgKey": "89541_1_1572870301"  
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing FAIL: Indicates failure
ErrorCode	Integer	Error Identifier: 0: Indicates success Non-zero: Indicates failure
ErrorInfo	String	Error message
MsgTime	Integer	Timestamp of when the message is sent, corresponding to the server time
MsgKey	String	Unique Message Identifier, used for recall. Length must not exceed 50 characters

## Error codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this interface is always 200. The real error code and error message are indicated by ErrorCode, ErrorInfo in the response body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error code	Description
10002	Internal server error. Please try again.
10004	Invalid parameters. Please check the request based on the error description.
10007	Insufficient operational permissions.
10010	The official account user does not exist, or it may have existed but has now been disbanded.
10015	Official Account User ID is invalid, please check if the Official Account User ID is correctly filled.
10016	The App backend refuses this operation through Third-party Callback.

10017	Unable to send messages due to being muted, please check if the sender has been set to mute.
10023	The frequency of sending messages exceeds the limit, please extend the interval between two message sending times.
80002	The message content is too long, currently supporting up to 12K Bytes of messages, please adjust the length of the message.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## Reference

[Message Format Description](#)

## Possible Trigger Callbacks

[Callback Before Sending a Official Account Message](#)

[Callback After Sending a Official Account Message](#)

# Historical Message

## Modifying Historical One-to-one Messages

Last updated : 2024-07-16 10:41:26

### API Calling Description

This API is used by the admin to modify historical one-to-one messages.

You can modify the `MsgBody` and `CloudCustomData` fields individually or at the same time for a message by overwriting the field values in a historical message with those specified in requests.

You can obtain the `MsgKey` of the one-to-one message to modify by the following means:

Enable [callback before sending a one-to-one message](#) or [callback after sending a one-to-one message](#) to record the `MsgKey` of each one-to-one message.

Use the [API for querying one-to-one messages](#) to query the `MsgKey` of the one-to-one message to modify.

For one-to-one messages sent through the RESTful APIs for [sending one-to-one messages to one user](#) and [sending one-to-one messages to multiple users](#), the message `MsgKey` is contained in response packets.

#### Note

Messages modified by this API cannot be restored.

### Sample request URL

```
https://xxxxxx/v4/openim/modify_c2c_msg?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>

v4/openim/modify_c2c_msg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Modifying only the `MsgBody` information of the message object

```
{
  "From_Account": "dramon1",
  "To_Account": "dramon2",
  "MsgKey": "1_2_3",
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hello"
      }
    }
  ]
}
```

Modifying only the `CloudCustomData` information of the message object

```
{
  "From_Account": "dramon1",
  "To_Account": "dramon2",
  "MsgKey": "1_2_3",
  "CloudCustomData": "your cloud custom data"
}
```

**Modifying both the `MsgBody` and `CloudCustomData` information of the message object**

```
{
  "From_Account": "dramon1",
  "To_Account": "dramon2",
  "MsgKey": "1_2_3",
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hello"
      }
    }
  ],
  "CloudCustomData": "your cloud custom data"
}
```

**Request fields**

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the message sender.
To_Account	String	Yes	<code>UserID</code> of the recipient
MsgKey	String	Yes	Unique identifier of the message to be modified. For how to obtain the <code>MsgKey</code> of the message, see the API description.
MsgBody	Array	No	Message body. For details on formats, see <a href="#">Message Formats</a> . (Note: a message can contain multiple message elements, in which case <code>MsgBody</code> is an array.)
CloudCustomData	String	No	Custom message data. It is saved in the cloud and will be sent to the receiver. Such data can be pulled after the app is uninstalled and reinstalled.

**Sample response**

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "succeed"
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
20001	Invalid request.
20002	<code>UserSig</code> or <code>A2</code> has expired.
20003	The <code>UserID</code> of the sender or recipient is invalid or does not exist. Make sure that the <code>UserID</code> has been imported into Chat.
20004	Network exception. Try again.
20005	Internal server error. Try again.
90001	Failed to parse the JSON request. Make sure the format is valid.
90002	The <code>MsgBody</code> in the JSON request does not meet message format requirements or <code>MsgBody</code> is not an array. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90003	The JSON request does not contain the <code>To_Account</code> field or the <code>To_Account</code> field is not a string.
90007	The <code>MsgBody</code> field in the JSON request is not an array. Change it to an array.
90009	The request requires app admin permissions.
90010	The JSON request does not meet message format requirements. For more information, see the

	<b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90012	The account specified in <code>To_Account</code> does not exist or is not registered. Make sure the account has been imported to Chat and is correct.
91000	Internal service error. Try again.
90992	Internal service error. Try again. If this error code is returned for all requests and webhook is enabled, make sure the app server returns the results to the Chat backend normally.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Sending One-to-One Messages to Multiple Users ([v4/openim/batchsendmsg](#))

Querying One-to-One messages ([v4/openim/admin\\_getroammsg](#))

Recalling One-to-One Messages ([v4/openim/admin\\_msgwithdraw](#))



# Modifying Historical Group Messages

Last updated : 2024-07-16 10:41:26

## API Calling Description

This API is used by the admin to modify historical group chat messages.

You can modify the `MsgBody` and `CloudCustomData` fields individually or at the same time for a message by overwriting the field values in a historical message with those specified in requests.

This API cannot be used to modify the historical messages of audio-video groups.

### Note

Messages modified by this API cannot be restored.

### Sample request URL

```
https://xxxxxx/v4/openim/modify_group_msg?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim/modify_group_msg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Modifying only the `MsgBody` information of the message object

```
{
  "GroupId": "@TGS#1HYEP2SHC",
  "MsgSeq": 23,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hello"
      }
    }
  ]
}
```

### Modifying only the `CloudCustomData` information of the message object

```
{
  "GroupId": "@TGS#1HYEP2SHC",
  "MsgSeq": 23,
  "CloudCustomData": "your cloud custom data"
}
```

### Modifying both the `MsgBody` and `CloudCustomData` information of the message object

```
{
  "GroupId": "@TGS#1HYEP2SHC",
  "MsgSeq": 23,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
```

```
        "Text": "hello"
      }
    },
    "CloudCustomData": "your cloud custom data"
  }
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the Group whose historical messages will be modified
MsgSeq	Integer	Yes	Sequence number of the message to be modified
MsgBody	Array	No	Message body. For format details, see <a href="#">Message Formats</a> . (Note: a message can contain multiple message elements, in which case <code>MsgBody</code> is an array.)
CloudCustomData	String	No	Custom message data. It is saved in the cloud and will be sent to the receiver. Such data can be pulled after the app is uninstalled and reinstalled.
TopicId	String	No	ID of the topic for which historical messages are modified. This field applies only to topic-enabled community groups.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "succeed"
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
20001	Invalid request.
20002	<code>UserSig</code> or <code>A2</code> has expired.
20004	Network exception. Try again.
20005	Internal server error. Try again.
90001	Failed to parse the JSON request. Make sure the format is valid.
90002	The <code>MsgBody</code> in the JSON request does not meet message format requirements or <code>MsgBody</code> is not an array. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90003	The JSON request does not contain the <code>To_Account</code> field or the <code>To_Account</code> field is not a string.
90007	The <code>MsgBody</code> field in the JSON request is not an array. Change it to an array.
90009	The request requires app admin permissions.
90010	The JSON request does not meet message format requirements. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
91000	Internal service error. Try again.
90992	Internal service error. Try again. If this error code is returned for all requests and webhook is enabled, make sure the app server returns the results to the Chat backend normally.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Sending Ordinary Messages in a Group ([v4/group\\_open\\_http\\_svc/send\\_group\\_msg](#))

Pulling Historical Messages ([group\\_open\\_http\\_svc/group\\_msg\\_get\\_simple](#))

# Pulling Historical One-to-one Messages

Last updated : 2024-07-16 10:41:26

## Feature Overview

This API is used by the app admin to query the message history of a one-to-one conversation from the perspective of a party of the conversation based on a specified time range.

The one-to-one conversation to be queried is specified by `Operator_Account` and `Peer_Account` in the request and the query is performed from the perspective of the party specified by `Operator_Account`. The query result contains the messages sent between both parties. The specific sender and recipient of each message are specified by `From_Account` and `To_Account` respectively.

In most cases, the query result is the same if you perform the query from the perspective of either party of a one-to-one conversation. However, there are four cases where the results are different (some messages in the conversation can be queried by one party but not by the other party):

One party of the conversation has cleared the message history of the conversation by calling the terminal API [clearC2CHistoryMessage\(\)](#).

One party of the conversation has deleted the conversation by calling the terminal API [deleteConversation\(\)](#), web/uni-app API [deleteConversation](#), or [server API for deleting a conversation](#) with `ClearRamble` being set to `1`.

One party of the conversation has deleted some messages in the conversation by calling the terminal API [deleteMessages\(\)](#) or web/uni-app API [deleteMessage](#).

For messages sent through [the API for sending one-to-one messages to one user](#) or [the API for sending one-to-one messages to multiple users](#), `SyncOtherMachine` is set to `2`, indicating that specified messages are not synced to the message history of the other party of the conversation.

The query result contains recalled messages indicated by the `MsgFlagBits` field.

The `IsPeerRead` field in the query result indicates whether the recipient sends the read receipt of the message. The field value is `1` only when the recipient calls the [sendMessageReadReceipts \(Android / iOS and Mac / Windows\)](#) or [sendMessageReadReceipt \(Web&\)](#) API.

If you want to recall a message, you can first call this API to query the `MsgKey` of the message and then call the [RESTful API for recalling one-to-one messages](#) to recall the message.

The time range of message records that can be queried depends on the roaming message storage period, which is seven days by default. You can modify the message roaming period via the Chat console. Extending the message roaming period is a value-added service. For more information, see [Message Storage](#).

If the total size of the messages within the requested time range exceeds the upper size limit (currently 13 KB) of the response, continued pulling is needed. You can see whether all the requested messages have been pulled by checking the `Complete` field in the response.

# API Calling Description

## Sample request URL

```
https://xxxxxx/v4/openim/admin_getroammsg?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim/admin_getroammsg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample requests and responses

For example, `user1` and `user2` had a conversation, and you want to query the conversation history from 2020-03-20 10:00:00 to 2020-03-20 11:00:00 from the perspective of user2.

### Sample request

```
{
  "Operator_Account": "user2",
  "Peer_Account": "user1",
  "MaxCnt": 100,
  "MinTime": 1584669600,
  "MaxTime": 1584673200
}
```

### Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Complete": 0,
  "MsgCnt": 12, //12 messages were returned for the pull.
  "LastMsgTime": 1584669680,
  "LastMsgKey": "549396494_2578554_1584669680",
  "MsgList": [
    {
      "From_Account": "user1",
      "To_Account": "user2",
      "MsgSeq": 549396494,
      "MsgRandom": 2578554,
      "MsgTimeStamp": 1584669680,
      "MsgFlagBits": 0,
      "IsPeerRead": 0,
      "MsgKey": "549396494_2578554_1584669680",
      "MsgBody": [
        {
          "MsgType": "TIMTextElem",
          "MsgContent": {
            "Text": "msg 1"
          }
        }
      ]
    },
    {
      "CloudCustomData": "your cloud custom data"
    }
  ],
  {
    "From_Account": "user2",
    "To_Account": "user1",
    "MsgSeq": 1054803289,
    "MsgRandom": 7201,
    "MsgTimeStamp": 1584669689,
```



```

        "MsgFlagBits": 0,
        "IsPeerRead": 0,
        "MsgKey": "1054803289_7201_1584669689",
        "MsgBody": [
            {
                "MsgType": "TIMTextElem",
                "MsgContent": {
                    "Text": "msg 2"
                }
            }
        ],
        "CloudCustomData": "your cloud custom data"
    },
    { ... } // The remaining ten messages are not listed for simplicity.
]
}

```

In the response, `"Complete": 0` indicates that not all messages generated within the time range have been pulled. Therefore, continued pulling is required.

**In the continued pulling request, the value of `MaxTime` must be changed to the value of `LastMsgTime` in the response, and the `LastMsgKey` in the response must be entered,** as shown below:

#### Sample continued pulling request

```

{
    "Operator_Account": "user2",
    "Peer_Account": "user1",
    "MaxCnt": 100,
    "MinTime": 1584669600,
    "MaxTime": 1584669680,
    "LastMsgKey": "549396494_2578554_1584669680"
}

```

#### Sample response

```

{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "Complete": 1,
    "MsgCnt": 5, // Five messages were returned for the pull.
    "LastMsgTime": 1584669601,
    "LastMsgKey": "1456_23287_1584669601",
    "MsgList": [
        {

```

```

    "From_Account": "user1",
    "To_Account": "user2",
    "MsgSeq": 1456,
    "MsgRandom": 23287,
    "MsgTimeStamp": 1584669601,
    "MsgFlagBits": 0,
    "IsPeerRead": 1,
    "MsgKey": "1456_23287_1584669601",
    "MsgBody": [
      {
        "MsgType": "TIMTextElem",
        "MsgContent": {
          "Text": "msg 13"
        }
      }
    ],
    "CloudCustomData": "your cloud custom data"
  },
  {
    "From_Account": "user2",
    "To_Account": "user1",
    "MsgSeq": 9806,
    "MsgRandom": 14,
    "MsgTimeStamp": 1584669602,
    "MsgFlagBits": 0,
    "IsPeerRead": 1,
    "MsgKey": "9806_14_1584669602",
    "MsgBody": [
      {
        "MsgType": "TIMTextElem",
        "MsgContent": {
          "Text": "msg 14"
        }
      }
    ],
    "CloudCustomData": "your cloud custom data"
  },
  { ... } // The remaining three messages are not listed for simplicity.
]
}

```

In the response, `"Complete": 1` indicates that all messages generated within the time range have been pulled.

If the value of `Complete` in the response is `0`, you need to continue pulling messages until the value of `Complete` becomes `1`.

## Request fields

Field	Type	Required	Description
Operator_Account	String	Yes	UserID of one party of the conversation for which messages are queried. The result may be different if you query messages for the other party of the same conversation. For more information, see the API description section.
Peer_Account	String	Yes	<code>UserID</code> of the other party in the conversation
MaxCnt	Integer	Yes	Number of messages to query
MinTime	Integer	Yes	Minimum value of the time range for message query, in seconds
MaxTime	Integer	Yes	Maximum value of the time range for message query, in seconds
LastMsgKey	String	No	<code>MsgKey</code> of the last message that was pulled previously. This field is required for continued pulling. For more information, see the preceding <a href="#">sample</a> .

## Sample response

Response to a successful request

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Complete": 1,
  "MsgCnt": 1,
  "LastMsgTime": 1584669680,
  "LastMsgKey": "549396494_2578554_1584669680",
  "MsgList": [
    {
      "From_Account": "user1",
      "To_Account": "user2",
      "MsgSeq": 549396494,
      "MsgRandom": 2578554,
      "MsgTimeStamp": 1584669680,
      "MsgFlagBits": 0,
      "IsPeerRead": 0,
      "MsgKey": "549396494_2578554_1584669680",
      "MsgBody": [
        {
          "MsgType": "TIMTextElem",
          "MsgContent": {
            "Text": "1"
          }
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "CloudCustomData": "your cloud custom data"
}
]
}

```

### Response to a failed request

```

{
  "ActionStatus": "FAIL",
  "ErrorInfo": "Fail to Parse json data of body, Please check it",
  "ErrorCode": 90001
}

```

### Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
Complete	Integer	Whether all messages have been pulled. <code>0</code> : no, continued pulling is required; <code>1</code> : yes
MsgCnt	Integer	Number of messages that were pulled this time
LastMsgTime	Integer	Time when the last message was pulled this time
LastMsgKey	String	Identifier of the last message pulled this time
MsgList	Array	List of returned messages
MsgFlagBits	Integer	Message attribute. <code>0</code> : normal message; <code>8</code> : recalled message
IsPeerRead	Integer	Whether the recipient has sent the read receipt of this message. Valid values: <code>0</code> for no and <code>1</code> for yes. For details, see the feature description of this API.
MsgBody	Array	Message body. For details on formats, see <a href="#">Message Formats</a> . (Note: A message can contain multiple message elements, in which case <code>MsgBody</code> is an array.)
CloudCustomData	String	Custom message data. It is saved in the cloud and will be sent to the peer end. Such data can be pulled after the app is uninstalled and reinstalled.

MsgKey	String	Message identifier. You can use this field when calling the <a href="#">RESTful API for recalling one-to-one messages</a> .
--------	--------	---

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON request. Make sure the format is valid.
90003	The <code>To_Account</code> field is missing in the JSON request or it is not a string.
90008	The <code>From_Account</code> field is missing in the JSON request or the account it specifies does not exist.
90009	The request requires app admin permissions.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[Sending One-to-One Messages to One User](#) (v4/openim/sendmsg)

[Sending One-to-One Messages to Multiple Users](#) (v4/openim/batchsendmsg)

[Recalling One-to-One Messages](#) (v4/openim/admin\_msgwithdraw)

# Pulling Historical Group Messages

Last updated : 2024-07-16 10:41:26

## Background

Tencent Chat group messages are sorted by seq, and seq is allocated according to the order in which group messages are received by the server. The seq is greater for group messages sent earlier and smaller for group messages sent later.

To pull all the messages of a group, you do not need to enter the seq for the initial pull. Instead, the server automatically returns the latest messages. For subsequent pulls, enter the previously returned smallest seq minus 1. If the value of `IsPlaceMsg` in the returned message is `1`, it indicates that the message with this seq has expired, failed to be stored, or been deleted.

## Feature Overview

This API allows the app administrator to pull the historical messages of a group.

## API Calling Description

### Applicable group types

Group Type ID	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

Above are the Chat built-in groups. For more information, please see [Group System](#).

### Note

Audio-video groups (AVChatRoom) do not support this API because the historical messages of this type of group cannot be stored.

## Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/group_msg_get_simple?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/group_msg_get_simple	The request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Pulls the historical messages of a group. The most recent `ReqMsgNumber` group messages will be returned.

```
{
  "GroupId": "@TGS#15ERQPAER",    // The ID of the group of which messages are
to be pulled
  "ReqMsgNumber": 2                // The number of messages to be pulled
}
```

### Pulling by seq

Pulls the historical messages of a group based on the specified seq.

The seq of the returned messages is less than or equal to the `ReqMsgNumber` of `ReqMsgSeq` .

```
{
  "GroupId": "@TGS#15ERQPAER",
  "ReqMsgSeq": 7803321,           // The maximum seq of the requested messages. The
messages with a seq less than or equal to `ReqMsgSeq` will be returned.
  "ReqMsgNumber": 2
}
```

### Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the group of which messages are to be pulled
ReqMsgNumber	Integer	Yes	The number of historical messages to be pulled. At present, a maximum of 20 historical messages can be returned per request. Therefore, please set the value of this field to 20 or less.
ReqMsgSeq	Integer	No	The maximum seq of the messages to be pulled
WithRecalledMsg	Integer	Optional	Whether to pull recalled messages. <code>1</code> : pull recalled messages. Recalled messages are not pulled by default.
TopicId	String	No	ID of the topic for which messages are pulled. This field applies only to topic-enabled communities.

### Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "@TGS#15ERQPAER",
  "IsFinished": 1,
  "RspMsgList": [
    {
```



```

    "From_Account": "144115197276518801",
    "IsPlaceMsg": 0,
    "MsgBody": [
      {
        "MsgContent": {
          "Data": "\\b\\u0001\\u0010\\u0006\\u001A\\u0006
MaoTong",

          "Desc": "MIF",
          "Ext": ""
        },
        "MsgType": "TIMCustomElem"
      },
      {
        "MsgContent": {
          "Data": "",
          "Index": 15
        },
        "MsgType": "TIMFaceElem"
      }
    ],
    "MsgPriority": 1,
    "MsgRandom": 51083293,
    "MsgSeq": 7803321,
    "MsgTimeStamp": 1458721802
  },
  {
    "From_Account": "144115198339527735",
    "IsPlaceMsg": 0,
    "MsgBody": [
      {
        "MsgContent": {
          "Data": "\\b\\u0001\\u0010\\u0006\\u001A\\u000F
Watermelon Girl",

          "Desc": "MIF",
          "Ext": ""
        },
        "MsgType": "TIMCustomElem"
      },
      {
        "MsgContent": {
          "Text": "Report"
        },
        "MsgType": "TIMTextElem"
      }
    ],
    "MsgPriority": 1,
    "MsgRandom": 235168582,

```

```

      "MsgSeq": 7803320,
      "MsgTimeStamp": 1458721797
    }
  ]
}

```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorInfo	String	Error information
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
<code>groupID</code>	<code>String</code>	The group ID in the request
IsFinished	Integer	Whether all the requested messages are returned. <code>1</code> : All the requested messages are returned. <code>0</code> : Not all requested messages are returned because the messages are too long or the number of messages is greater than 20. <code>2</code> : The requested messages are too long or the number of messages is greater than 20 and all the messages have expired.
MsgList	Array	A list of returned messages
From_Account	String	The UserID of the message sender
IsPlaceMsg	Integer	Whether a message is empty. If the message has been deleted or expired, <code>MsgBody</code> is empty and the value of this field is <code>1</code> . If the message has been recalled, the value of this field is <code>2</code> .
MsgPriority	Integer	Message priority, which is used for message deduplication. A value is entered when the client sends a message. If no value is entered, the server automatically generates one. <code>1</code> : high priority; <code>2</code> : normal priority, <code>3</code> : low priority; <code>4</code> : lowest priority
MsgRandom	Integer	Message random value, which is used for message deduplication. A value is entered when the client sends a message. If no value is entered, the server automatically generates one.
MsgSeq	Integer	The unique seq of the message. The smaller the value, the earlier the message was sent.
MsgTimeStamp	Integer	The timestamp when the message was sent, which follows the server time system

MsgBody	Array	Message body. For more information, see <a href="#">Message Formats</a> .
IsSystemMsg	Integer	Whether the message is a system message. 1 : Yes

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. The operator must have permissions to perform corresponding operations.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Setting the Unread Message Count of a Member ([v4/group\\_open\\_http\\_svc/set\\_unread\\_msg\\_num](#))

# Pulling Historical Official Account Messages

Last updated : 2024-07-16 10:41:26

## Background

Chat's Official Account messages are sorted by Seq, which is assigned according to the order in which the server receives the Official Account messages. Earlier messages have a smaller Seq, and later ones have a larger Seq. Chat generates a MsgKey for each Official Account message, in the format of "Seq\_1\_ServerTime".

If users want to pull all messages from a Official Account, they need to provide the LastMsgKey of the message. There is no need to fill in the LastMsgKey for the first pull, as the Server will automatically return the latest messages. For subsequent pulls, fill in the LastMsgKey with the LastMsgKey returned from the last request.

If the returned message's IsPlaceMsg is 1, it means that the message of this Seq has either expired, failed to store, or has been deleted.

## Feature Overview

App administrators can use this interface to pull historical messages from the official account.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/official_account_open_http_svc/official_account_msg_get_simple?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The table below only lists the parameters modified when calling this API and their descriptions. For more details on the parameters, please refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain for the country/region where your SDKAppID is located: China: <code>console.tim.qq.com</code>

	Singapore : adminapisgp.im.qcloud.com Seoul : adminapikr.im.qcloud.com Frankfurt : adminapiger.im.qcloud.com Silicon Valley : adminapiusa.im.qcloud.com Jakarta : adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/official_account_msg_get_simple	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a>
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a>
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: json

## Maximum calling frequency

200 queries/sec.

## Sample request packets

### Basic form

Pulling the official account's historical messages returns the latest ReqMsgNumber messages from the official account.

```
{
  "Official_Account": "@TOA#_15ERQPAER",    //Pull messages from the official
account user
  "ReqMsgNumber": 2        //Number of messages to pull
}
```

### Continue pulling by LastMsgKey

Returns messages earlier than the specified LastMsgKey.

```
{
  "Official_Account": "@TOA#_15ERQPAER",
  "LastMsgKey": "71_1_1698741698",          // Continuation MsgKey
  "ReqMsgNumber": 2
}
```

## Request packet fields

Field	Type	Attribute	Description
Official_Account	String	Mandatory	Official account user whose historical messages are to be pulled
LastMsgKey	String	Optional	The MsgKey of the last message fetched last time. This field must be filled for continuation pulling. See the method of filling above in <a href="#">Sample</a>
ReqMsgNumber	Integer	Optional	Number of requested messages
WithRecalledMsg	Integer	Optional	Whether to pull messages that have been revoked, filling in 1 indicates the need to pull messages after revocation; by default, messages after revocation are not pulled

## Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Official_Account": "@TOA#_15ERQPAER",
  "IsFinished": 1,
  "LastMsgKey": "71_1_1698741698"
  "RspMsgList": [
    {
      "From_Account": "144115197276518801",
      "IsPlaceMsg": 0,
      "MsgBody": [
        {
          "MsgContent": {
            "Data": "\\b\\u0001\\u0010\\u0006\\u001A\\u0006Cat's
Eye",
            "Desc": "MIF",
            "Ext": ""
          },
          "MsgType": "TIMCustomElem"
        },
        {

```

```
        "MsgContent": {
            "Data": "",
            "Index": 15
        },
        "MsgType": "TIMFaceElem"
    }
],
"MsgSeq": 71,
"MsgKey" : "71_1_1698741698"
"MsgTimeStamp": 1698741698
},
{
    "From_Account": "144115198339527735",
    "IsPlaceMsg": 0,
    "MsgBody": [
        {
            "MsgContent": {
                "Data":
"\b\u0001\u0010\u0006\u001A\u000FWatermelon Sister.",
                "Desc": "MIF",
                "Ext": ""
            },
            "MsgType": "TIMCustomElem"
        },
        {
            "MsgContent": {
                "Text": "Report"
            },
            "MsgType": "TIMTextElem"
        }
    ],
    "MsgSeq": 72,
    "MsgKey" : "72_1_1698741700"
    "MsgTimeStamp": 1698741700
}
]
}
```

Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing FAIL: Indicates failure

ErrorInfo	String	Error message
ErrorCode	Integer	Error Identifier: 0: Indicates success Non-zero: Indicates failure
Official_Account	String	Official Account Users in the Request
IsFinished	Integer	Whether all messages in the request range have been returned When all messages in the request range are successfully returned, the value is 1 When the message length is too long or the range is too large (exceeding 20), resulting in the inability to return all messages, the value is 0 When all messages before the request range have expired, the value is 2
RspMsgList	Array	The returned message list
IsPlaceMsg	Integer	Whether it is a hollow message, when the message is deleted or expired: If MsgBody is empty, this field is 1 For a recalled message, this field is 2
MsgKey	String	This tag identifies the message, which can be used for <a href="#">recalling Official Account Messages</a>
MsgSeq	Integer	Message seq, used to identify unique messages, the smaller the value, the earlier it is sent
MsgTimeStamp	Integer	Timestamp when the message was sent (in seconds), server time
MsgBody	Object / Array	Message Content, for details please refer to the <a href="#">Message Content MsgBody</a>

## Error codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this interface is always 200. The real error code and error message are indicated by ErrorCode, ErrorInfo in the response body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error code	Description
10002	Internal server error, please try again
10003	Illegal Request Command Word



10004	Invalid parameter, please check if the request is correct based on the error description
10007	Insufficient Operational Permissions, the operator must be a role that is authorized to perform the corresponding operations in the Official Account
10010	Official Account User does not exist, or may have existed in the past, but is now dissolved
10015	Invalid Official Account User ID, please check if the Official Account User ID is correctly filled in

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

# Delete Message

## Deleting Messages Sent by a Specified User

Last updated : 2024-07-16 10:41:26

### Feature Overview

This API can be used to recall the messages sent by a specified user among the last 1,000 messages.

### API Calling Description

#### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

#### Note

This API recalls messages that are in historical message storage, but audio-video groups (AVChatRoom) do not support historical message storage. Therefore, this API does not take effect when it is used to recall messages for a user in an audio-video group (but no error will be returned).

This API currently supports silent recall only. A recalled message will be marked as recalled on the server, and the client will not be notified of the recall. The client can know that the message is recalled only when it pulls roaming messages.

#### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/delete_group_msg_by_sender?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/delete_group_msg_by_sender	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Recall the messages sent by a specified user among the last 1,000 messages of a group:

```
{
  "GroupId": "@TGS#2C5SZEAEF",    // Required
  "Sender_Account": "leckie"      // Required
}
```

## Request fields

--	--	--	--

Field	Type	Required	Description
GroupId	String	Yes	ID of the group where messages are to be recalled
Sender_Account	String	Yes	ID of the sender of the messages to be recalled
TopicId	String	No	Topic ID, which indicates message recall in the corresponding topic and applies only to topic-enabled communities.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorInfo	String	Error information
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other

	users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Deleting Group Members ([v4/group\\_open\\_http\\_svc/delete\\_group\\_member](#))

# Clearing Historical Group Messages

Last updated : 2024-11-11 14:36:23

## Overview

This API is used to clear historical messages sent by users in group chats.

This API marks group messages as deleted so the SDK cannot pull them, rather than actually deleting them. The admin can still pull cleared historical messages through the "Pull Historical Group Messages" API.

## API Calling Description

### Applicable Group Types

Group Type ID	Support for This RESTful API
Private	Yes. The same as work groups (Work) in the new version
Public	Yes
ChatRoom	Yes. The same as meeting groups (Meeting) in the new version
AVChatRoom	Yes
Community	No

The above group types are built in Chat. For details, see [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/clear_group_msg?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists parameters involved in modification and their description when this API is called. For more parameter details, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The domain name corresponding to the country/region where your SDKAppID is located.

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/clear_group_msg</code>	Request API.
<code>sdkappid</code>	The SDKAppID assigned by the Chat console when an app is created.
<code>identifier</code>	It must be the app admin account. For more details, see <a href="#">App Admin</a> .
<code>usersig</code>	The signature generated by the app admin account. For specific operations, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format. The value is fixed to <code>json</code> .

## Maximum Call Frequency

200 calls per second.

## Sample Request

### Basic form

Directly clear historical messages of the current group chat.

```
{
  "GroupId": "@TGS#15ERQPAER"
}
```

### Clearing by MsgSeq

Clear historical messages of the group before the specified seq (including the seq).

```
{
  "GroupId": "@TGS#15ERQPAER",
  "MsgSeq": 7803321           // Clear messages less than or equal to MsgSeq.
}
```

## Request Fields

Field	Type	Attribute	Description
-------	------	-----------	-------------

GroupId	String	Required	The ID of the group whose historical messages are to be cleared.
MsgSeq	Integer	Optional	Historical messages less than or equal to MsgSeq are cleared. If this field is left blank, all historical messages are cleared.

## Sample Response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response Fields

Field	Type	Description
ActionStatus	String	Request result: OK: successful, FAIL: failed
ErrorInfo	String	Error information
ErrorCode	Integer	Error code: 0: successful Other values: failed

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The real error code and error information are indicated by ErrorCode and ErrorInfo in the response body.

For public error codes (60000 to 79999), see [Error Codes](#).

Private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Please try again.
10003	Invalid request command word.
10004	Invalid parameter. Please check if the request is correct based on the error description.
10007	Insufficient operation permissions. The operator must be the admin.



10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid Group ID. Please check if the Group ID is filled in correctly.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## Reference

Pull historical group chat messages ([v4/group\\_open\\_http\\_svc/group\\_msg\\_get\\_simple](#)).

# Withdraw Message

## Recalling One-to-One Messages

Last updated : 2025-02-28 18:00:15

### Feature Overview

This API enables administrators to recall one-to-one messages.

This API can recall all one-to-one messages, including those sent by clients or by the [v4/openim/sendmsg](#) and [v4/openim/batchsendmsg](#) RESTful APIs.

To recall one-to-one messages sent by clients, you can enable the [C2C.CallbackBeforeSendMsg](#) or [C2C.CallbackAfterSendMsg](#) callback API. Record the `MsgKey` of each one-to-one message via the callback API, and enter it in the `MsgKey` field of this API to recall the message. You can also use the [v4/openim/admin\\_getroammsg](#) API to query the `MsgKey` of the one-to-one message to be recalled, and enter it in the `MsgKey` field of this API to recall the message.

The `MsgKey` field in the responses to the [v4/openim/sendmsg](#) and [v4/openim/batchsendmsg](#) API calls is required to recall the one-to-one messages sent via these two APIs.

Once a one-to-one message is recalled by this API, it is recalled from offline storage, roaming storage, and the local cache of the sender's and recipient's clients.

This API can recall one-to-one messages sent at any time. There's no time limit.

#### Note

Please note that one-to-one messages recalled by this API cannot be restored.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/openim/admin_msgwithdraw?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim/admin_msgwithdraw	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

```
{
  "From_Account": "vinson",
  "To_Account": "dramon",
  "MsgKey": "31906_833502_1572869830"
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the message sender.
To_Account	String	Yes	<code>UserID</code> of the recipient
MsgKey	String	Yes	The unique identifier of the message to be recalled, which can be found in the responses to the <a href="#">v4/openim/sendmsg</a> and <a href="#">v4/openim/batchsendmsg</a> API calls.

## Sample response

Response to a successful request

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response to a failed request

```
{
  "ActionStatus": "FAIL",
  "ErrorInfo": "Fail to Parse json data of body, Please check it",
  "ErrorCode": 90001
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
20022	The message to recall does not exist. Please check.
20023	The message has been recalled.
90001	Failed to parse the JSON request. Make sure the format is valid.

90003	The <code>To_Account</code> field is missing in the JSON request packet or the account it specifies does not exist.
90008	The <code>From_Account</code> field is missing in the JSON request packet or the account it specifies does not exist.
90009	The request requires app admin permissions.
90054	Invalid <code>MsgKey</code> .
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Sending One-to-One Messages to One User](#) (v4/openim/sendmsg)

[Sending One-to-One Messages to Multiple Users](#) (v4/openim/batchsendmsg)

[Querying One-to-One Messages](#) (v4/openim/admin\_getroammsg)

# Recalling Group Messages

Last updated : 2025-02-28 18:00:15

## Feature Overview

This API enables admins to recall group messages.

This API enables the recall of all group messages valid for roaming, including group messages sent from a client and those sent with a RESTful API.

To recall group messages sent from a client, enable [After a Group Message Is Sent](#) first, which records the `GroupId` and `MsgSeq` of each group message, and then enter the `GroupId` and `MsgSeqList` obtained from the webhook in this API to recall the message. Alternatively, get information of group messages to be recalled with [Pulling Historical Messages](#), and then recall the message with this API.

To recall group messages sent with the RESTful API [Sending Ordinary Messages in a Group](#), get `MsgSeq` contained in the response packets of these APIs and then recall the messages.

Once a message is recalled by this API, it is recalled from the local caches of the sender's and recipients' clients.

This API can recall group messages sent for any period of time, but the group messages must be valid for roaming.

## API Calling Description

### Applicable group types

Group type ID	Group Message Recall
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes. Supported from Native SDK 7.4.4634 and Web SDK 3.1.2.
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/group_msg_recall?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/group_msg_recall	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

The message recall operation is irrevocable. Please call this API with caution.

```
{
  "GroupId": "@TGS#2J4SZEAE",
  "MsgSeqList": [
    {
      "MsgSeq": 100
    },
  ],
}
```

```
    {
      "MsgSeq":101
    }
  ]
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the Group whose historical messages will be modified
MsgSeqList	Array	Yes	The sequence list of messages to be recalled. Up to 10 message sequences can be contained in a request.
MsgSeq	Integer	Yes	The sequence of the message to be recalled.
TopicId	String	No	Topic ID, which indicates message recall in the corresponding topic and applies only to topic-enabled communities.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "RecallRetList": [
    {
      "MsgSeq":100,
      "RetCode":10030
    },
    {
      "MsgSeq":101,
      "RetCode":0
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed



ErrorInfo	String	Error information
RecallRetList	Array	Result details of the message recall request.
MsgSeq	Integer	The sequence of an individual recalled message.
RetCode	Integer	The recall result of an individual message. 0 for success and others for failures. See the error codes below for more details.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. The operator must have permissions to perform corresponding operations.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.
10030	The message requested does not exist.
10031	The message requested is beyond the allowed time range.
10032	The message requested cannot be recalled.
10033	This type of group does not support message recalls.
10041	The app (SDKAppID) is configured not to support group message recalls.

## API Debugging Tool

---

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Sending System Messages in a Group ([v4/group\\_open\\_http\\_svc/send\\_group\\_system\\_notification](#))

# Recalling Official Account Messages

Last updated : 2024-07-16 10:41:26

## Feature Overview

Administrator revokes official account message.

This interface can revoke all official account messages within the valid roaming period, including official account messages sent by the client and those issued by the REST API interface.

If you need to recall official account messages sent by the client, you can enable [Callback after sending a message to the official account](#), record the MsgKey of each official account message via this callback interface, and then fill in the MsgKeyList parameter of this interface for revocation. You can also use [Pulling historical messages of official account users](#) to query the relevant information of the official account messages to be revoked, and then use this interface for revocation.

If you need to revoke official account messages sent via the REST API [Official Account Users Send Broadcast Messages](#) interface, it's necessary to record the MsgKey field in the response packages of these interfaces for revocation.

After invoking this interface to recall a message, the local cache of the recipient's client will also be recalled.

This interface can recall official account messages without any time limit, meaning it can recall official account messages from any time, but the roaming time of the official account messages needs to be within the validity period.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/official_account_open_http_svc/official_account_msg_recall?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The table below only lists the parameters modified when calling this API and their descriptions. For more details on the parameters, please refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain for the country/region where your SDKAppID is located: China: <code>console.tim.qq.com</code>

	Singapore : adminapisgp.im.qqcloud.com Seoul : adminapikr.im.qqcloud.com Frankfurt : adminapiger.im.qqcloud.com Silicon Valley : adminapiusa.im.qqcloud.com Jakarta : adminapiidn.im.qqcloud.com
v4/official_account_open_http_svc/official_account_msg_recall	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a>
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a>
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: <code>json</code>

## Maximum calling frequency

200 queries/sec.

## Sample request packets

Once a message is recalled, it cannot be restored. Please be cautious when invoking this interface.

```
{
  "Official_Account": "@TOA#_2J4SZEAEI",
  "MsgKeyList": [
    "71_1_1698741698",
    "72_1_1698741698"
  ],
  "Reason": "Revocation Reason"
}
```

## Request packet fields

Field	Type	Attribute	Description

Official_Account	String	Mandatory	Operated Official Account User ID
MsgKeyList	Array	Mandatory	List of MsgKeys for messages being recalled, a single request can recall up to 10 messages
Reason	String	Optional	Reason for message recall, length restricted to a maximum of 100 bytes

### Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "RecallRetList": [
    {
      "MsgKey": "71_1_1698741698",
      "RetCode": 10030
    },
    {
      "MsgKey": "72_1_1698741699",
      "RetCode": 0
    }
  ]
}
```

### Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing FAIL: Indicates failure
ErrorCode	Integer	Error Identifier: 0: Indicates success Non-zero: Indicates failure
ErrorInfo	String	Error message
RecallRetList	Array	Detailed results of the message recall request
MsgKey	String	MsgKey of a single message being recalled
RetCode	Integer	Recall result of a single message: 0 means success; other values mean failure, refer to the error code explanations below

## Error codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this interface is always 200. The real error code and error message are indicated by `ErrorCode`, `ErrorInfo` in the response body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error code	Description
10002	Internal server error, please try again
10003	Illegal Request Command Word
10004	Invalid parameter, please check if the request is correct based on the error description
10007	Insufficient operational permissions
10010	Official Account User does not exist, or may have existed in the past, but is now dissolved
10015	Invalid Official Account User ID, please check if the Official Account User ID is correctly filled in
10030	The message requested for recall does not exist
10031	The message requested for recall exceeds the time limit
10032	The message requested for recall does not support recall operations

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

# Read Receipt

## Marking One-to-One Messages as Read

Last updated : 2025-02-28 18:00:15

### Feature Overview

This API is used to mark the messages of a one-to-one conversation as read.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/openim/admin_set_msg_read?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/openim/admin_set_msg_read	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

The admin specifies that dramon1 marks all the messages of its one-to-one conversation with dramon2 as read.

```
{
  "Report_Account": "dramon1",
  "Peer_Account": "dramon2"
}
```

## Request fields

Field	Type	Required	Description
Report_Account	String	Yes	<code>UserId</code> of the account for which to mark messages as read
Peer_Account	String	Yes	<code>UserId</code> of the other account in the one-to-one conversation
MsgReadTime	String	No	Timestamp (in seconds). All messages before the specified timestamp are marked as read. If no value is specified, the current timestamp is used.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed



ErrorInfo	String	Error information
-----------	--------	-------------------

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON request. Make sure the format is valid.
90003	The <code>To_Account</code> field is missing in the JSON request, or the account specified by <code>To_Account</code> does not exist.
90008	The <code>From_Account</code> field is missing in the JSON request, or the account specified by <code>From_Account</code> does not exist.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[Sending One-to-One Messages to One User](#) (v4/openim/sendmsg)

[Sending One-to-One Messages to Multiple Users](#) (v4/openim/batchsendmsg)

[Querying One-to-One Messages](#) (v4/openim/admin\_getroammsg)

[Recalling One-to-One Messages](#) (v4/openim/admin\_msgwithdraw)

# Pulling Group Message Read Receipt Details

Last updated : 2025-01-17 10:36:05

## Feature Overview

An app admin can call this API to pull the list of members who have or have not read a group message.

### Note:

This feature is available only in the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Group configuration > Group Message Read Receipts**, and enable the feature.

## API Calling Description

### Applicable group types

Group Type ID	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	No

Above are the Chat built-in groups. For more information, please see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_msg_receipt_detail?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/get_group_msg_receipt_detail	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Pull the list of members who have or have not read a group message in batches:

### Basic format

You can use the `Cursor` and `Count` fields to control the pulling-by-page mode:

`Count` : Specifies the maximum number of members in the `ReadList` or `UnreadList` array in the response.

Maximum value: 200

`Cursor` : Specifies a member position from which subsequent information is to be pulled. For the first request, the client request parameter `Cursor` must pass in `""`. For the last request, the server returns `1` for the `IsFinish` parameter, indicating that the information pulling ends.

```
{
  "GroupId": "@TGS#2TTV7VSII",
  "MsgSeq": 1,
  "Filter": 1,           // `0`: Pull the list of members who have read the
                        message; `1`: Pull the list of members who have not read the message
}
```

```
"Cursor":"","          // Position of the last member pulled. Enter "" for the
first request
"Count":5              // Maximum number of members to pull
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group of which message read receipts are to be pulled
MsgSeq	Integer	Yes	Seq of the message to pull
Filter	Integer	Yes	Whether to pull the list of members who have or have not read a group message. <code>0</code> : Pull the list of members who have read a group message; <code>1</code> : Pull the list of members who have not read a group message
Cursor	String	Yes	Position of the last member pulled. Enter "" for the first request.
Count	Integer	Yes	Maximum number of members that can be pulled each time. The maximum value is 200.

## Sample response

### Pull the list of members who have read a group message

```
{
  "ActionStatus": "OK",
  "Cursor": "0",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "IsFinish": 1,          // The list is fully pulled. No further pull is
required.
  "MsgSeq": 1,
  "ReadList": [          // List of members who have read the message
    {
      "Read_Account": "test1"
    }
  ]
}
```

### Pull the list of members who have not read the message

```
{
  "ActionStatus": "OK",
  "Cursor": "144115213529088617", // `Cursor` value to be passed in for the
next request
}
```

```
"ErrorCode": 0,
"ErrorInfo": "",
"IsFinish": 0,    // The list is not fully pulled, and further pull is
required.
"MsgSeq": 1,
"UnreadList": [  // List of members who have not read the message
  {
    "Unread_Account": "test"
  },
  {
    "Unread_Account": "test6"
  },
  {
    "Unread_Account": "test3"
  },
  {
    "Unread_Account": "test5"
  },
  {
    "Unread_Account": "test4"
  }
]
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorInfo	String	Error information
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
IsFinished	Integer	Whether the list is fully pulled. <code>0</code> : No; <code>1</code> : Yes
MsgSeq	Integer	Seq of the message to pull
ReadList	Array	List of messages who have read the message
Read_Account	String	Members who have read the message
UnreadList	Array	List of messages who have not read the message
Unread_Account	String	Members who have not read the message
Cursor	String	<code>Cursor</code> value to be passed in for the next request

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. The operator must have permissions to perform corresponding operations.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.
10062	The read receipt doesn't exist.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[Pulling group message read receipt information](#)

# Pulling Read Receipts for Group Messages

Last updated : 2025-01-14 11:59:12

## Feature Overview

This API is used to pull read receipts for group messages.

### Note:

This feature is available only in the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Group configuration > Group Message Read Receipts**, and enable the feature.

## API Calling Description

### Applicable group types

Group Type ID	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	No

Above are the Chat built-in group types. For more information, please see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_msg_receipt?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=999999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/get_group_msg_receipt	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Batch pull the read receipts for group messages, including the read count and unread count.

### Basic format

```
{
  "GroupId": "@TGS#2TTV7VSII",
  "MsgSeqList": [          // Pull the list of message `seq` values
    { "MsgSeq": 1 },
    { "MsgSeq": 2 }
  ]
}
```

## Request fields

Field	Type	Required	Description



GroupId	String	Yes	ID of the group of which message read receipts are to be pulled
MsgSeqList	Array	Yes	List of message <code>seq</code> values
MsgSeq	Integer	Yes	Seq of the message to pull

### Sample response

#### Basic format

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "GroupMsgReceiptList": [ // Read receipts
    {
      "Code": 0,
      "MsgSeq": 1,
      "ReadNum": 1, // Read count of the group message
      "UnreadNum": 6 // Unread count of the group message
    },
    {
      "Code": 0,
      "MsgSeq": 2,
      "ReadNum": 1,
      "UnreadNum": 6
    }
  ]
}
```

### Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorInfo	String	Error information
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
GroupMsgReceiptList	Array	Read receipt for the group message
Code	Integer	Error code for the read receipt. For example, <code>10062</code> indicates that the read receipt doesn't exist.
MsgSeq	Integer	Seq of the message to pull

ReadNum	Integer	Read count of the group message
UnreadNum	Integer	Unread count of the group message

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. The operator must have permissions to perform corresponding operations.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.
10062	The read receipt doesn't exist.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Pulling group message read receipt details ([v4/group\\_open\\_http\\_svc/get\\_group\\_msg\\_receipt\\_detail](#))

# Message Extension

## Pulling the Message Extension of a One-to-One Message

Last updated : 2025-01-14 11:59:12

### Feature Overview

App admins and conversation members can pull the message extension of a one-to-one message. Message extension is a set of custom key-value pairs.

#### Note

To use this feature, you need to [purchase the Pro edition](#)、[Pro Plus edition](#)、[Enterprise edition](#) and enable the **Set message extension** feature on the **Login and Message** page in the Chat console.

Up to 300 key-value pairs can be set for each one-to-one message.

For one-to-one messages for which key-value pairs are set, they need to be specified to support message extension when being sent. For operation details, see [Sending One-to-One Messages to One User](#).

### Sample request URL

```
https://xxxxxx/v4/openim_msg_ext_http_svc/get_key_values?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/openim_msg_ext_http_svc/get_key_values	Request API

sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Pull the message extension of a one-to-one message whose `MsgKey` is `44739199_12_1665388280` :

```
{
  "From_Account": "62768",
  "To_Account": "116400",
  "MsgKey": "44739199_12_1665388280"
}
```

Pull the message extension of a one-to-one message whose `MsgKey` is `44739199_12_1665388280` and `Seq` is greater than or equal to 2:

```
{
  "From_Account": "62768",
  "To_Account": "116400",
  "MsgKey": "44739199_12_1665388280",
  "StartSeq": 2
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	No	<code>UserID</code> of the sender (used to specify the message sender)
To_Account	String	Yes	<code>UserID</code> of the recipient

MsgKey	String	Yes	Unique identifier of the one-to-one message. For how to obtain the <code>MsgKey</code> of the message, see the API for sending one-to-one messages to one user.
StartSeq	Integer	No	Message extension will be pulled for messages whose <code>Seq</code> is greater than or equal to <code>StartSeq</code> .

Sample response

```
{
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ActionStatus": "OK",
  "CompleteFlag": 1,
  "LatestSeq": 2,
  "ClearSeq": 0,
  "ExtensionList": [
    {
      "Key": "k1",
      "Value": "v1",
      "Seq": 1
    },
    {
      "Key": "k2",
      "Value": "v2",
      "Seq": 1
    },
    {
      "Key": "k3",
      "Value": "v3",
      "Seq": 2
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful. <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information

ExtensionList	Array	List of key-value pairs returned, where the key-value pairs are sorted by <code>Seq</code> . Up to 200 key-value pairs can be returned per pull. If there are more than 200 key-value pairs, they need to be pulled in batches based on <code>StartSeq</code> .
CompleteFlag	Integer	<code>1</code> : All key-value pairs are returned. <code>0</code> : Not all key-value pairs are returned.
LatestSeq	Integer	Latest <code>Seq</code> of the backend, which is the maximum value of <code>Seq</code> among all key-value pairs.
ClearSeq	Integer	<code>Seq</code> for the clearing operation. All key-value pairs whose <code>Seq</code> is less than or equal to <code>ClearSeq</code> must be deleted locally on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10008	Invalid request: The request is not an Pro edition 、 Pro Plus edition、 Enterprise edition plan request.
23004	The one-to-one message does not exist.

# Configuring Message Extension for a One-to-One Message

Last updated : 2025-01-14 11:59:12

## Feature Overview

App admins and conversation members can configure message extension for common one-to-one messages.

Message extension is a set of custom key-value pairs.

### Note

To use this feature, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#)、 [Enterprise edition](#) and enable the **Set message extension** feature on the **Login and Message** page in the Chat console.

Up to 300 key-value pairs can be set for each one-to-one message.

For one-to-one messages for which key-value pairs are set, they need to be specified to support message extension when being sent. For operation details, see [Sending One-to-One Messages to One User](#).

## Sample request URL

```
https://xxxxxx/v4/openim_msg_ext_http_svc/set_key_values?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim_msg_ext_http_svc/set_key_values	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created

identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Configure message extension for a one-to-one message whose `MsgKey` is `44739199_12_1665388280` :

```
{
  "From_Account": "62768",
  "To_Account": "116400",
  "MsgKey": "44739199_12_1665388280",
  "OperateType": 1,
  "ExtensionList": [
    {"Key": "k1", "Value": "v1", "Seq": 0},
    {"Key": "k2", "Value": "v2", "Seq": 0},
    {"Key": "k3", "Value": "v3", "Seq": 0}
  ]
}
```

Delete certain message extension for a one-to-one message whose `MsgKey` is `44739199_12_1665388280` :

```
{
  "From_Account": "62768",
  "To_Account": "116400",
  "MsgKey": "44739199_12_1665388280",
  "OperateType": 2,
  "ExtensionList": [
    {"Key": "key1", "Value": "", "Seq": 1}
  ]
}
```

Clear all message extension for a one-to-one message whose `MsgKey` is `44739199_12_1665388280` :

```
{
```



```
"From_Account": "62768",
"To_Account": "116400",
"MsgKey": "44739199_12_1665388280",
"OperateType": 3
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	No	<code>UserID</code> of the sender (used to specify the message sender)
To_Account	String	Yes	<code>UserID</code> of the recipient
MsgKey	String	Yes	Unique identifier of the one-to-one message. For how to obtain the <code>MsgKey</code> of the message, see the API for sending one-to-one messages to one user.
OperateType	Integer	Yes	<code>1</code> : Setting key-value pairs for a message; <code>2</code> : Deleting certain key-value pairs for a message; <code>3</code> : Clearing all key-value pairs for a message
ExtensionList	Array	Yes (when <code>OperateType</code> is <code>1</code> or <code>2</code> )	Up to 20 key-value pairs can be set or deleted per request

Fields of the objects in `ExtensionList` are described as follows:

Field	Type	Required	Description
Key	String	Yes	Key in a key-value pair, which can be up to 100 bytes
Value	String	Yes	Value in a key-value pair, which can be up to 1,000 bytes
Seq	Integer	Yes (when the API is not called by an app admin)	Version number of the current key-value pair. <code>Seq</code> is <code>0</code> when the API is called by a conversation member to configure a key-value pair for the first time and is <code>1</code> when the configuration is successful. For subsequent key-value pair configuration, the latest <code>Seq</code> of the key-value pair needs to be passed back to the backend, which then checks whether the key-value pair has been modified by others. If the <code>Seq</code> does not match the key-value pair, the configuration fails, and an error code is returned. The backend does not verify the <code>Seq</code> when the API is called by an app admin.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ExtensionList": [
    {
      "ErrorCode": 0,
      "Extension": {
        "Key": "k1",
        "Value": "v1",
        "Seq": 1
      }
    },
    {
      "ErrorCode": 23001,
      "Extension": {
        "Key": "k2",
        "Value": "v1234",
        "Seq": 2
      }
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful. <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information
ExtensionList	Array	Message extension configuration result

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

--	--

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10008	Invalid request: The request is not an Pro edition 、 Pro Plus edition、 Enterprise edition plan request.
23001	Seq conflict: The key-value setting has been modified by others. You need to pull the latest Seq to request again.
23002	The configured one-to-one message does not support message extension.
23003	Too many key-value setting attempts (more than 200 attempts per minute per message)
23004	The configured one-to-one message does not exist.

# Pulling Message Extension of a Group Message

Last updated : 2025-01-14 11:59:12

## Feature Overview

App admins and group members can pull the message extension of a one-to-one message. Message extension is a set of custom key-value pairs.

### Note

To use this feature, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#) and enable the **Set message extension** feature on the **Login and Message** page in the Chat console.

Up to 300 key-value pairs can be set for each group message.

For group messages for which key-value pairs are set, they need to be specified to support message extension when being sent. For operation details, see [Sending Ordinary Messages in a Group](#).

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/openim_msg_ext_http_svc/group_get_key_values?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim_msg_ext_http_svc/group_get_key_values	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Pull the message extensions of the group message whose `MsgSeq` is 158:

```
{
  "GroupId": "@TGS#1YMVAB3IZ",
  "MsgSeq": 158
}
```

Pull the message extensions of the group message whose `MsgSeq` is 158 and `Seq` is equal to or greater than 2:

```
{
  "GroupId": "@TGS#1YMVAB3IZ",
  "MsgSeq": 158,
  "StartSeq": 2
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	Group ID
MsgSeq	Integer	Yes	Seq of the group message to be configured
StartSeq	Integer	No	Message extension will be pulled for messages whose Seq is greater than or equal to StartSeq .

Sample response

```
{
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ActionStatus": "OK",
  "CompleteFlag": 1,
  "LatestSeq": 2,
  "ClearSeq": 0,
  "ExtensionList": [
    {
      "Key": "key1",
      "Value": "1",
      "Seq": 1
    },
    {
      "Key": "key2",
      "Value": "2",
      "Seq": 1
    },
    {
      "Key": "key3",
      "Value": "3",
      "Seq": 2
    }
  ]
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
ExtensionList	Array	List of key-value pairs returned, where the key-value pairs are sorted by <code>Seq</code> . Up to 200 key-value pairs can be returned per pull. If there are more than 200 key-value pairs, they need to be pulled in batches based on <code>StartSeq</code> .
CompleteFlag	Integer	<code>1</code> : All key-value pairs are returned. <code>0</code> : Not all key-value pairs are returned.
LatestSeq	Integer	Latest <code>Seq</code> of the backend, which is the maximum value of <code>Seq</code> among all key-value pairs.
ClearSeq	Integer	<code>Seq</code> for the clearing operation. All key-value pairs whose <code>Seq</code> is less than or equal to <code>ClearSeq</code> must be deleted locally on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10008	Invalid request: The request is not an Pro edition 、 Pro Plus edition or Enterprise edition plan request.
23004	The group message does not exist.

# Configuring Message Extension for a Group Message

Last updated : 2024-07-16 10:40:20

## Feature Overview

App admins and group members can configure message extension for common group messages. Message extension is a set of custom key-value pairs.

### Note:

To use this feature, you need to [purchase the Ultimate edition](#) and enable the **Set message extension** feature on the **Login and Message** page in the Chat console.

Up to 300 key-value pairs can be set for each group message.

For group messages for which key-value pairs are set, they need to be specified to support message extension when being sent. For operation details, see [Sending Ordinary Messages in a Group](#).

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/openim_msg_ext_http_svc/group_set_key_values?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```



## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim_msg_ext_http_svc/group_get_key_values	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Configure message extensions for the group message whose `MsgSeq` is 158:

```
{
  "GroupId": "@TGS#1YMVAB3IZ",
  "MsgSeq": 158,
  "OperateType": 1,
  "ExtensionList": [
    {"Key": "key1", "Value": "value1", "Seq": 0},
  ]
}
```

```
    {"Key": "key2", "Value": "value2", "Seq": 0},  
  ]  
}
```

Delete message extensions for the group message whose `MsgSeq` is 158:

```
{  
  "GroupId": "@TGS#1YMVAB3IZ",  
  "MsgSeq": 158,  
  "OperateType": 2,  
  "ExtensionList": [  
    {"Key": "key1", "Value": "", "Seq": 1}  
  ]  
}
```

Clear all message extensions for the group message whose `MsgSeq` is 158:

```
{  
  "GroupId": "@TGS#1YMVAB3IZ",  
  "MsgSeq": 158,  
  "OperateType": 3  
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	Group ID
MsgSeq	Integer	Yes	Seq of the group message to be configured
OperateType	Integer	Yes	<code>1</code> : Setting key-value pairs for a message; <code>2</code> : Deleting certain key-value pairs for a message; <code>3</code> : Clearing all key-value pairs for a message
ExtensionList	Array	Yes (when <code>OperateType</code> is <code>1</code> or <code>2</code> )	Up to 20 key-value pairs can be set or deleted per request

Fields of the objects in `ExtensionList` are described as follows:

Field	Type	Required	Description
Key	String	Yes	Key in a key-value pair, which can be up to 100 bytes
Value	String	Yes	Value in a key-value pair, which can be up to 1,000 bytes

Seq	Integer	Yes (when the API is not called by an app admin)	Version number of the current key-value pair. <code>Seq</code> is <code>0</code> when the API is called by a group member to configure a key-value pair for the first time and is <code>1</code> when the configuration is successful. For subsequent key-value pair configuration, the latest <code>Seq</code> of the key-value pair needs to be passed back to the backend, which then checks whether the key-value pair has been modified by others. If the <code>Seq</code> does not match the key-value pair, the configuration fails, and an error code is returned. The backend does not verify the <code>Seq</code> when the API is called by an app admin.
-----	---------	--	---

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ExtensionList": [
    {
      "ErrorCode": 0,
      "Extension": {
        "Key": "key1",
        "Value": "value1",
        "Seq": 1
      }
    },
    {
      "ErrorCode": 23001,
      "Extension": {
        "Key": "key2",
        "Value": "value1234",
        "Seq": 2
      }
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed

ErrorInfo	String	Error information
ExtensionList	Array	Message extension configuration result

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10008	Invalid request: The request is not an Ultimate edition plan request.
23001	Seq conflict: The key-value setting has been modified by others. You need to pull the latest Seq to request again.
23002	The configured group message does not support message extension.
23003	Too many key-value setting attempts (more than 200 attempts per minute per message)
23004	The configured group message does not exist.

# Pushing to All Users

## API for Pushing to All Users

Last updated : 2025-03-20 16:08:31

### Push to All Users Overview

The push to all users service is a group of RESTful APIs implemented based on the Chat communication architecture. This service is used to meet message push requirements of the application, including pushing messages to all users, pushing messages by tag, and pushing messages by attribute. Clients can receive pushed messages by using the online and offline push features (Android background notifications and APNs) of the SDK. The group push service also supports offline storage of messages, helping operators achieve their operational goals more efficiently.

#### Note:

An account can receive messages pushed to all users only after it has been logged in to or manually imported. The feature of pushing to all users is available only to users with Pro edition 、 Pro Plus edition or Enterprise edition accounts. See [Configuration Change Ticket](#) to apply for this feature. The feature will be enabled **48 hours** after your application is approved.

### Basic Features

Push messages to all users of the application.

Push messages by user tag.

Push messages by user attribute.

### Strengths

The group push service is provided based on the client Chat SDK and Chat backend, which guarantee messaging capabilities and system availability.

Message delivery is guaranteed based on the Chat SDK, and the service provides simple message broadcasting capabilities for applications.

You can set up to 10 push attributes for each user. Each attribute can be set separately without affecting the others.

The service pushes messages based on multiple attributes connected by AND or OR logic.

The service pushes messages based on multiple tags connected by AND or OR logic.

The service pushes messages only to online users and retains messages offline for up to 7 days(withdrawal and roaming are not supported).

You can customize messages.

You can specify the sender account.

## Use Cases

### Pushing messages to all users

Example 1: a game application plans to provide special offers on Christmas and needs to push the notification to all users. In this case, the group push service can be used to boost efficiency. In addition, to inform more users of this promotion, you can set the offline message retention period. In this way, even if some users are not online when the message is pushed, they can still receive the message when they go online within this offline retention period. This improves the quality and effectiveness of the promotion.

Example 2: a livestreaming application plans to launch a large-scale live marketing activity and needs to push the notification to all users. In this case, you can push the notification to all users starting 7 days before the activity and set the offline message retention period to 7 days. When the activity starts, you can push the notification to all users again, without specifying an offline retention period. In this way, all online users can receive the notification and join the live room, attracting more users into the live room.

### Pushing messages to users by user tag

For example, a financial product plans to push a financial planning service to users who follow "stock A" or "stock B". In this case, you can push messages by tag:

1. When a user follows "stock A" or "stock B", add the corresponding tag to the user by calling the API for adding tags.
2. When a user unfollows "stock A" or "stock B", delete the corresponding tag of the user by calling the API for deleting tags.
3. Set the push condition to "stock A" or "stock B" (by using the TagsOr feature) in the API for pushing messages. In this way, all users that follow "stock A" or "stock B" can receive the messages.

### Pushing messages by user attribute

For example, users of a game are classified into non-members, ordinary members, gold members, and platinum members. Assume the operator plans to push an activity to platinum members in Shenzhen. In this case, they can push the message by attribute:

1. Set the application attribute names. In this example, the game users have two attributes: membership tier and city. You can set attribute 0 as the membership tier and attribute 1 as the city.
2. When the membership tier of a user changes (for example, when the membership expires or the user purchases a membership), modify the membership tier attribute of the user by calling the API for setting user attributes. For example, when a user purchases a gold membership, set the membership tier attribute of the user to "gold member".
3. When the city where a user is located changes, modify the city attribute of the user by calling the API for setting user attributes. For example, when the city changes from Beijing to Shanghai, set the city attribute of the user to

"Shanghai".

4. During the activity, you can call the push API and set the membership tier attribute to platinum and the city attribute to Shenzhen. Then, all platinum members in Shenzhen will receive the pushed message.

**Note:**

In this scenario, pushing by attribute is preferable to pushing by tag. If you push the message by tag, when the city of the user changes from Shenzhen to Guangzhou, the application needs to delete the "Shenzhen" tag of the user and then add the "Guangzhou" tag, which involves two API calls. However, if you push the message by attribute, only one API call is required to modify the city attribute of the user.

## Related APIs

[Pushing Messages to All Users](#)

[Setting Application Attribute Names](#)

[Obtaining Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Obtaining User Attributes](#)

[Adding User Tags](#)

[Obtaining User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags for Users](#)

# Pushing to All Users

Last updated : 2025-03-20 16:10:25

Pushing to all users is an excellent tool for application user operations. It not only supports sending specific content to all users, but also can send personalized content to specific user groups based on tags and attributes, such as member events, and regional notifications. This helps effectively attract, convert, and activate new users.

## Feature Overview

This API can be used to push messages to all users.

This API can be used to push messages by user attribute.

This API can be used to push messages by user tag.

When the admin pushes messages, the message sender displayed to the recipients is the admin.

When the admin specifies an account to push messages to other accounts, the sender displayed to the recipients is not the admin but the account specified by the admin.

This API supports offline storage of messages, but not message roaming.

It takes time to push messages to all users. The required time depends on the number of accounts. Typically, it is within one minute.

This API allows you to push messages only to online users by setting the `MsgLifeTime` parameter to `0`.

### Note:

The "pushing to all users" feature is only available on the IM Pro edition 、 Pro Plus edition or Enterprise edition. To use it, [purchase the Pro edition 、 Pro Plus edition or Enterprise edition](#). For more information, see [Pricing](#).

## API Call Description

The feature of pushing to all users is available only to users with Pro edition 、 Pro Plus edition or Enterprise edition accounts. See [Configuration Change Ticket](#) to apply for this feature. The feature will be enabled **48 hours** after your application is approved.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_push?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

--	--



Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. Chinese mainland: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/all_member_push/im_push	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.
sdkappid	The <code>SDKAppID</code> assigned by the IM console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Calling frequency

This API includes pushing to all and by attribute and tag. By default, it can be called 100 times every day. The interval between two pushes must be greater than one second.

## Sample request

### Pushing messages to all users

The admin pushes messages to all users and retains the messages offline for 120 seconds.

```
{
  "From_Account": "admin",
  "MsgRandom": 56512,
  "MsgLifeTime": 120,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ]
}
```

```
]
}
```

The admin specifies an account for pushing to all users and retains the messages offline for 120 seconds. In the sample, the sender account is `xiaoming` :

```
{
  "From_Account": "xiaoming",
  "MsgRandom": 3674128,
  "MsgLifeTime": 120,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ]
}
```

The admin specifies an account for pushing to all users, sets offline push information, and retains the messages offline for 120 seconds.

```
{
  "From_Account": "xiaoming",
  "MsgRandom": 3674128,
  "MsgLifeTime": 120,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ],
  "OfflinePushInfo": {
    "PushFlag": 0,
    "Desc": "Content to push offline",
    "Ext": "Passthrough content",
    "AndroidInfo": {
      "Sound": "android.mp3"
    },
    "ApnsInfo": {
      "Sound": "apns.mp3",
      "BadgeMode": 1, // If this field is left as default or is set to `0`,
                        // the message is counted. If this field is set to `1`, the message is not
    }
  }
}
```

counted, that is, the badge counter in the upper-right corner does not increase.

```
"Title": "apns title", // APNs title
"SubTitle": "apns subtitle", // APNs subtitle
"Image": "www.image.com" // Image URL
}
}
}
```

The admin pushes messages to all users and retains the messages offline for 120 seconds.

```
{
  "From_Account": "admin",
  "MsgRandom": 21302570,
  "MsgLifeTime": 120,
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ]
}
```

### Pushing messages by user tag

The admin pushes messages to users tagged with "Stock A" and "Stock B" and retains the messages offline for 120 seconds.

```
{
  "From_Account": "admin",
  "MsgRandom": 214,
  "MsgLifeTime": 120,
  "Condition": {
    "TagsAnd": ["Stock A", "Stock B"]
  },
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ]
}
```

The admin pushes messages to users tagged with "Stock A" and "Stock B" and retains the messages offline for 120 seconds.

```
{
  "From_Account": "admin",
  "MsgRandom": 124032,
  "MsgLifeTime": 120,
  "Condition":{
    "TagsOr": ["Stock A","Stock B"]
  },
  "MsgBody":[
    {
      "MsgType": "TIMTextElem",
      "MsgContent":{
        "Text": "hi, beauty"
      }
    }
  ]
}
```

### Pushing messages by user attribute

The admin pushes messages to Shenzhen Platinum Premier users and retains the messages offline for 120 seconds.

```
{
  "From_Account": "admin",
  "MsgRandom": 389475,
  "MsgLifeTime": 120,
  "Condition":{
    "AttrsAnd": {
      "Membership Level": "Platinum Premier members",
      "city": "Shenzhen"
    }
  },
  "MsgBody":[
    {
      "MsgType": "TIMTextElem",
      "MsgContent":{
        "Text": "hi, beauty"
      }
    }
  ]
}
```

The admin pushes messages to Shenzhen Platinum Premier users and retains the messages offline for 120 seconds.

```
{
  "From_Account": "admin",
```

```

"MsgRandom": 9312457,
"MsgLifeTime": 120,
"Condition":{
  "AttrsAnd": {
    "Membership Level": "Platinum Premier users",
    "city": "Shenzhen"
  }
},
"MsgBody": [
  {
    "MsgType": "TIMTextElem",
    "MsgContent":{
      "Text": "hi, beauty"
    }
  }
]
}

```

## Request fields

Field	Type	Attribute	Description
Condition	Object	Optional	Valid values: AttrsOr AttrsAnd TagsOr TagsAnd AttrsOr and AttrsAnd can coexist, and TagsOr and TagsAnd can coexist. However, tag conditions and attribute conditions cannot coexist. If no condition is specified, messages are pushed to all users.
MsgRandom	Integer	Required	Random number (32-bit unsigned integer) of the message. It is used by the backend for message deduplication within a second. Make sure a random number is entered.
MsgBody	Array	Required	Message body. For more information on the message format, see <a href="#">Message Formats</a> . Note that MsgBody is an array that can contain multiple message elements.
MsgType	String	Required	TIM message object type. Valid values: TIMTextElem (text message) TIMLocationElem (location message) TIMFaceElem (emoji message) TIMCustomElem (custom message) TIMSoundElem (voice message)

			<code>TIMImageElem</code> (image message) <code>TIMFileElem</code> (file message) <code>TIMVideoFileElem</code> (video message)
MsgContent	Object	Required	Different message object types ( <code>MsgType</code> ) have different formats ( <code>MsgContent</code> ). For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
MsgLifeTime	Integer	Optional	Offline message storage duration, in seconds. The maximum duration is 604,800 seconds (7 days). The default value is 0, which indicates that messages are not stored offline and will be pushed only to online users.
From_Account	String	Optional	Account of the message sender
AttrsOr	Object	Optional	A set of attribute conditions connected by OR. Note that attribute conditions and tag conditions cannot be used at the same time.
AttrsAnd	Object	Optional	A set of attribute conditions connected by AND. Note that attribute conditions and tag conditions cannot be used at the same time.
TagsOr	Array	Optional	Union of tag conditions. A tag is a string of up to 50 bytes. Note that attribute conditions and tag conditions cannot be used at the same time. The <code>TagsOr</code> condition can contain up to ten tags.
TagsAnd	Array	Optional	Intersection of tag conditions. A tag is a string of up to 50 bytes. Note that attribute conditions and tag conditions cannot be used at the same time. The <code>TagsAnd</code> condition can contain up to ten tags.
OfflinePushInfo	Object	Optional	The information to be pushed offline. For more information, see <a href="#">Message Formats</a> .

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TaskId": "1400123456_144115212910570789_4155518400_15723514"
}
```

## Response fields

Field	Type	Description

ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error message
TaskId	String	Push task ID

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error message. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90002	The <code>MsgBody</code> field in the JSON request does not meet message format requirements or it is not an array. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90005	The <code>MsgRandom</code> field is missing in the JSON request or it is not an integer.
90007	The <code>MsgBody</code> field in the JSON request is not an array. Change it to an array.
90009	The request requires app admin permissions.
90010	The JSON request does not meet message format requirements. For more information, see the <b>Message Element TIMMsgElement</b> section in <a href="#">Message Formats</a> .
90020	The tag length exceeds the limit (the maximum length allowed is 50 bytes).
90022	<code>TagsOr</code> and <code>TagsAnd</code> in the push conditions contain repeated tags.
90024	Pushes are too frequent. The interval between two pushes must be greater than 1 second.
90026	Incorrect offline message storage period. The value cannot exceed 7 days.
90032	The number of tags in the push conditions exceeds 10, or the number of tags in the tag adding request exceeds 10.
90033	Invalid attribute.

90039	Message push by attribute and message push by tag are mutually exclusive.
90040	A tag in the push conditions is null.
90045	The feature of pushing to all users is not enabled.
90047	The number of pushes exceeds the daily quota (default quota: 100).
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)



# Setting Application Attribute Names

Last updated : 2025-01-14 14:31:39

## Feature Overview

You can set a maximum of 10 custom user attributes for each app. This API is used to set the name of each attribute. After you set attribute names, they can be used for push by user attribute and other purposes.

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_set_attr_name?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/all_member_push/im_set_attr_name	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	SDKAppID assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Maximum call frequency

100 times/second

## Sample request

Set attribute `0` of the app to `sex`, attribute `1` to `city`, and attribute `2` to `country`.

```
{
  "AttrNames": {
    "0": "sex",
    "1": "city",
    "2": "country"
  }
}
```

## Request fields

Field	Type	Required	Description
Digital key	String	Yes	Attribute number (0 to 9).
Attribute name	String	Yes	The attribute name cannot exceed the length limit of 50 bytes. An app can have a maximum of 10 push attributes (numbered from 0 to 9), and users can customize the meaning of each attribute.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description

ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Getting Application Attribute Names

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to get application attribute names. Before calling this API, you need to [set application attribute names](#).

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_get_attr_name?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/all_member_push/im_get_attr_name	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	SDKAppID assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always json .

Maximum call frequency

100 times/second

Sample request

```
{ }
```

Request fields

None.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "AttrNames": {
    "0": "sex",
    "1": "city",
    "2": "Membership level"
  }
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. OK : Successful. FAIL : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information
AttrNames	Object	A series of "key:value" pairs. Each "key:value" pair indicates the name of the corresponding attribute. For example, "0":"xxx" indicates that the name of attribute 0 is xxx.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Getting User Attributes

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used to obtain user attributes. To call this API, you must use the admin account. You can obtain a maximum of 100 users' attributes at a time. You need to [set app attribute names](#) before calling the API.

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise editionn. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise editionn](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_get_attr?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/all_member_push/im_get_attr	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	SDKAppID assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Maximum call frequency

100 times/second

## Sample request

```
{
  "To_Account": [
    "Zhang Xiaohong",
    "Chen Xiaoming",
    "abc"
  ]
}
```

## Request fields

Field	Type	Required	Description
To_Account	Array	Yes	List of target user accounts

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "UserAttrs": [
    {
      "To_Account": "Zhang Xiaohong",
      "Attrs": {
        "sex": "Female",
        "city": "New York"
      }
    },
    {
      "To_Account": "abc",
      "Attrs": {}
    }
  ],
}
```



```
{
  "To_Account": "Chen Xiaoming",
  "Attrs": {
    "sex": "M",
    "city": "Shenzhen"
  }
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information
UserAttrs	Array	List of user tags
To_Account	String	User account
Attrs	Object	Attribute content

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

## API Debugging Tool

---

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Setting User Attributes

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to set attributes for users. Each request can set attributes for a maximum of 100 users. Before using this API, be sure to [set app attribute names](#).

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_set_attr?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/all_member_push/im_set_attr	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	SDKAppID assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always json .

Maximum call frequency

100 times/second

Sample request

```
{
  "UserAttrs":
  [
    {
      "To_Account": "xiaojun012",
      "Attrs": {
        "sex": "attr1",
        "city": "attr2"
      }
    },
    {
      "To_Account": "xiaojun013",
      "Attrs": {
        "city": "attr3",
        "sex": "attr4"
      }
    }
  ]
}
```

Request fields

Field	Type	Required	Description
To_Account	String	Yes	Target user account
Attrs	Object	Yes	Attribute set. Each attribute is a key-value pair, with the key being the attribute name and the value being the user’s corresponding attribute value. A user attribute value cannot exceed 50 bytes.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
90033	Invalid attribute.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

---

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Deleting User Attributes

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to delete attributes for users. Each request can delete attributes for a maximum of 100 users. Before using this API, be sure to [set app attribute names](#).

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_remove_attr?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/all_member_push/im_remove_attr	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	SDKAppID assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always json .

Maximum call frequency

100 times/second

Sample request

```
{
  "UserAttrs": [
    {
      "To_Account": "xiaojun013",
      "Attrs": [
        "sex",
        "city"
      ]
    },
    {
      "To_Account": "xiaojun012",
      "Attrs": [
        "sex",
        "city"
      ]
    }
  ]
}
```

Request fields

Field	Type	Required	Description
To_Account	String	Yes	Target user account
Attrs	Array	Yes	Tag set. Note that you only need to specify the attribute names here. For more information on the format and meanings of Attrs , see <a href="#">Setting Application Attribute Names</a> .

Sample response

```
{
```



```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
90033	Invalid attribute.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

---

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Getting User Tags

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to obtain user tags. Up to 100 users' tags can be obtained at a time.

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#)、 [Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_get_tag?  
usersig=xxx&identifier=admin&sdkappid=888888888&random=999999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/all_member_push/im_get_tag	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.
sdkappid	<code>SDKAppID</code> assigned by the Chat console when an app is created

random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Maximum call frequency

100 times/second

## Sample request

```
{
  "To_Account": [
    "xiaojun012",
    "xiaojun013"
  ]
}
```

## Request fields

Field	Type	Required	Description
To_Account	Array	Yes	List of target user accounts

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "UserTags": [
    {
      "To_Account": "xiaojun012",
      "Tags": ["a", "b"]
    },
    {
      "To_Account": "xiaojun013",
      "Tags": ["a", "c"]
    }
  ]
}
```

## Response fields

Field	Type	Description
-------	------	-------------

ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information
UserTags	Array	List of user tags
To_Account	String	User account
Tags	Array	Tag content

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Adding User Tags

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to add tags for users.

### Note:

Each request can add tags for a maximum of 100 users. In the request body, a maximum of 10 tags can be added for each user.

A maximum of 100 tags can be set for each user. If a user has more than 100 tags, you need to delete old tags before adding new ones for the user.

The maximum length of a single tag is 50 bytes.

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#)、 [Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_add_tag?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com

v4/all_member_push/im_add_tag	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.
sdkappid	<code>SDKAppID</code> assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Maximum call frequency

100 times/second

## Sample request

```
{
  "UserTags": [
    {
      "To_Account": "xiaojun012",
      "Tags": ["a", "b"]
    },
    {
      "To_Account": "xiaojun013",
      "Tags": ["a", "b"]
    }
  ]
}
```

## Request fields

Field	Type	Required	Description
To_Account	String	Yes	Target user account
Tags	Array	Yes	Tag set

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
}
```



```
"ErrorCode":0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Getting User Tags](#)

[Deleting User Tags](#)

[Deleting All Tags of a User](#)

# Deleting User Tags

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to delete tags for users. Each request can delete tags for a maximum of 100 users.

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#)、 [Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_remove_tag?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/all_member_push/im_remove_tag	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	SDKAppID assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Maximum call frequency

100 times/second

## Sample request

```
{
  "UserTags": [
    {
      "To_Account": "xiaojun012",
      "Tags": ["a", "b"]
    },
    {
      "To_Account": "xiaojun013",
      "Tags": ["a", "b"]
    }
  ]
}
```

## Request fields

Field	Type	Required	Description
To_Account	String	Yes	Target user account
Tags	Array	Yes	Tag set

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
-------	------	-------------

ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting All Tags of a User](#)

# Deleting All Tags of a User

Last updated : 2025-01-14 14:31:39

## Feature Overview

This API is used by the admin to delete all tags for users. Each request can delete all tags for a maximum of 100 users.

## API Calling Description

Pushing to all users is available only to the Pro edition 、 Pro Plus edition、 Enterprise edition. To use it, you need to [purchase the Pro edition 、 Pro Plus edition、 Enterprise edition](#), go to the [console](#), choose **Feature Configuration > Login and Message > Push to all users**, and enable the feature.

### Sample request URL

```
https://xxxxxx/v4/all_member_push/im_remove_all_tags?
usersig=xxx&identifier=admin&sdkappid=88888888&random=99999999&contenttype=json
```

### Request parameters

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/all_member_push/im_remove_all_tags	Request API
usersig	Signature generated in the app admin account. For more information, see <a href="#">Generating UserSig</a> .
identifier	The app administration account.

sdkappid	<code>SDKAppID</code> assigned by the Chat console when an app is created
random	A random 32-bit unsigned integer
contenttype	The value is always <code>json</code> .

## Maximum call frequency

100 times/second

## Sample request

```
{
  "To_Account": [
    "xiaojun012",
    "xiaojun013"
  ]
}
```

## Request fields

Field	Type	Required	Description
To_Account	Array	Yes	Target user account

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code
ErrorInfo	String	Error information

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. **ErrorCode** and **ErrorInfo** in the response represent the actual error code and error information. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[API for Pushing to All Users](#)

[Pushing to All Users](#)

[Setting Application Attribute Names](#)

[Getting Application Attribute Names](#)

[Setting User Attributes](#)

[Deleting User Attributes](#)

[Getting User Attributes](#)

[Adding User Tags](#)

[Getting User Tags](#)

[Deleting User Tags](#)



# Session Related

## Conversation List

### Pulling a conversation list

Last updated : 2024-07-16 10:52:14

## Feature Overview

This API is used to pull a conversation list by page.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/recentcontact/get_list?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/get_list	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .

usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

```
{
  "From_Account": "id1",
  "TimeStamp": 0,
  "StartIndex": 0,
  "TopTimeStamp": 0,
  "TopStartIndex": 0,
  "AssistFlags": 7
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	<code>UserID</code> of the account for which to pull a conversation list
TimeStamp	Integer	Yes	Start time of general conversations. Enter <code>0</code> for the first page.
StartIndex	Integer	Yes	Starting point of general conversations. Enter <code>0</code> for the first page.
TopTimeStamp	Integer	Yes	Start time of pinned conversations. Enter <code>0</code> for the first page.
TopStartIndex	Integer	Yes	Starting point of pinned conversations. Enter <code>0</code> for the first page.
AssistFlags	Integer	Yes	Flag bits of conversations: Bit 0: Whether to support pinned conversations Bit 1: Whether to return an empty conversation Bit 2: Whether to support paginating pinned conversations

## Sample response

```
{
  "SessionItem": [
```

```
{
  "Type": 1,
  "To_Account": "id2",
  "MsgTime": 1630997627,
  "TopFlag": 1
},
{
  "Type": 2,
  "GroupId": "id3",
  "MsgTime": 1630997628,
  "TopFlag": 1
},
{
  "Type": 1,
  "To_Account": "id4",
  "MsgTime": 1630997630,
  "TopFlag": 0
},
{
  "Type": 2,
  "GroupId": "id5",
  "MsgTime": 1630997650,
  "TopFlag": 0
}
],
"CompleteFlag": 1,
"TimeStamp": 1631012800,
"StartIndex": 0,
"TopTimeStamp": 1631012800,
"TopStartIndex": 0,
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
SessionItem	Array	Array of conversation objects
Type	Integer	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
To_Account	String	<code>UserID</code> of the other conversation participant, which will be returned only for a one-to-one conversation

GroupId	String	Group ID, which will be returned only for a group conversation
MsgTime	Integer	Conversation duration
TopFlag	Integer	Flag of conversation pinning. <code>0</code> : General conversation; <code>1</code> : Pinned conversation
CompleteFlag	Integer	Completion flag. <code>1</code> : All conversation are returned; <code>0</code> : Pulling has not finished yet.
TimeStamp	Integer	Start time of the next pulled page for a general conversation, which is sent to the Chat backend via the <code>TimeStamp</code> field of the request during pulling-by-page
StartIndex	Integer	Starting point of the next pulled page for a general conversation, which is sent to the Chat backend via the <code>StartIndex</code> field of the request during pulling-by-page
TopTimeStamp	Integer	Start time of the next pulled page for a pinned conversation, which is sent to the Chat backend via the <code>TopTimeStamp</code> field of the request during pulling-by-page
TopStartIndex	Integer	Starting point of the next pulled page for a pinned conversation, which is sent to the Chat backend via the <code>TopStartIndex</code> field of the request during pulling-by-page
ActionStatus	String	The request result. <code>OK</code> : successful. <code>FAIL</code> : failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error message
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

--	--

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Session Unread Count

## Setting the Unread Message Count of a Member

Last updated : 2025-03-10 10:30:48

### Chat Note

This document is only for users who are migrating their apps to Tencent Cloud Chat.

## Feature Description

This API is used by app admins to set the unread message count of a group member. When this API is called, no callback is triggered and no notification is sent.

When you are migrating your app from another Chat system to Tencent Cloud Chat, you can use this API to set the unread message count of group members.

## API Call Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work group (Work) in the new version.
Public	Yes.
ChatRoom	No. Same as meeting group (Meeting) in the new version.
AVChatRoom	No.
Community	Yes.

These are the built-in group types in Chat. For detailed information, see [Group System](#).

### Note

ChatRoom and AVChatroom (audio-video) groups do not support unread message counts. Therefore, you cannot set an unread message count for members of these two group types. If you try to do so, no error will be returned.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/set_unread_msg_num?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table only describes the modified parameters when this API is called. For more information on other parameters, please see [RESTful API Introduction](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/group_open_http_svc/set_unread_msg_num	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, please see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated in the app admin account. For details on how to generate the signature, please see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format. The value is always json .

## Maximum call frequency

200 calls per second

## Sample request packet

This example sets the unread message count of a specified group member.

If the unread message count specified by this API is greater than the current number of messages in the group, the

unread message count will be set to the current number of messages in the group.

```
{
  "GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
  "Member_Account": "bob", // Target group member (required)
  "UnreadMsgNum":5 // Unread message count of the target member
}
```

Request packet fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the target group
Member_Account	String	Yes	Target group member
UnreadMsgNum	Integer	Yes	Unread message count of the target member

Sample response packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response packet fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed.
ErrorInfo	String	Error information
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), please see [Error Codes](#). The following table describes the error codes specific to this API:

--



Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Make sure to use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

Sending System Messages in a Group ([v4/group\\_open\\_http\\_svc/send\\_group\\_system\\_notification](#))

# Querying Unread One-to-One Message Count

Last updated : 2024-07-16 10:52:14

## Feature Overview

This API is used to query the unread message count of a one-to-one conversation or all one-to-one conversations.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/openim/get_c2c_unread_msg_num?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/openim/get_c2c_unread_msg_num	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

`contenttype`Request format, which should always be `json`.

## Maximum call frequency

200 calls per second

## Querying the total unread one-to-one message count of an account

### Sample request

This example shows how the admin queries the total unread one-to-one message count of `dramon1`. Only `To_Account` is required.

```
{
  "To_Account": "dramon1"
}
```

### Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "AllC2CUnreadMsgNum": 12
}
```

## Querying the unread message counts of multiple one-to-one conversations at a time

### Sample request

This example shows how the admin queries the unread message counts of `dramon1`'s conversations with `dramon2` and `teacher`.

```
{
  "To_Account": "dramon1",
  "Peer_Account": [
    "dramon2",
    "teacher"
  ]
}
```

### Sample response

```
{
  "ActionStatus": "OK",
```

```
"ErrorInfo": "",
"ErrorCode": 0,
"C2CUnreadMsgNumList": [
  {
    "Peer_Account": "dramon2",
    "C2CUnreadMsgNum": 12
  },
  {
    "Peer_Account": "teacher",
    "C2CUnreadMsgNum": 12
  }
]
```

## Request fields

Field	Type	Required	Description
To_Account	String	Yes	<code>UserID</code> of the user to query
Peer_Account	Array	No	<code>UserID</code> of the other party in the conversation to query This field is required to query a specific one-to-one conversation. The array can contain up to 10 <code>UserID</code> values.

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
AllC2CUnreadMsgNum	Integer	Total unread message count of all conversations
C2CUnreadMsgNumList.Peer_Account	String	<code>UserID</code> of the other party in the one-to-one conversation
C2CUnreadMsgNumList.C2CUnreadMsgNum	Integer	Unread message count in the one-to-one conversation
ErrorList.Peer_Account	String	Target account for which the query failed
ErrorList.ErrorCode	Integer	Error code. <code>70107</code> indicates that the account does not exist.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON request. Make sure the format is valid.
90003	The <code>To_Account</code> field is missing in the JSON request, or the account specified by <code>To_Account</code> does not exist.
90008	The <code>From_Account</code> field is missing in the JSON request, or the account specified by <code>From_Account</code> does not exist.

## References

[Sending One-to-One Messages to One User](#) (v4/openim/sendmsg)

[Sending One-to-One Messages to Multiple Users](#) (v4/openim/batchsendmsg)

[Querying One-to-One Messages](#) (v4/openim/admin\_getroammsg)

[Recalling One-to-One Messages](#) (v4/openim/admin\_msgwithdraw)

# Delete Session

## Deleting a conversation

Last updated : 2024-07-16 10:52:15

### Feature Overview

This API is used to delete a conversation. It can also clear roaming messages.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/recentcontact/delete?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/delete	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

200 calls per second

Sample request

Deleting a one-to-one conversation

```
{
  "From_Account": "id1",
  "Type": 1,
  "To_Account": "id2",
  "ClearRamble": 1
}
```

Deleting a group conversation

```
{
  "From_Account": "id1",
  "Type": 2,
  "ToGroupid": "id2",
  "ClearRamble": 1
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	<code>UserID</code> of the account for which to delete a conversation
Type	Integer	Yes	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
To_Account	String	No	UserID of the other conversation participant, which will be assigned a value only for a one-to-one conversation
ToGroupid	string	No	Group ID, which will be assigned a value only for a group conversation.
ClearRamble	Integer	No	Whether to clear roaming messages. <code>1</code> : yes; <code>0</code> : no

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
ActionStatus	String	The request result. <code>OK</code> : successful. <code>FAIL</code> : failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error message
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.



# API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Session Grouping Tag

## Creating Conversation Group Data

Last updated : 2025-01-14 14:35:32

### Feature Overview

Conversation group mark data is independent of recent contacts. You can use RESTful APIs to add, delete, modify, and query conversation group, standard conversation mark, custom conversation mark, and other fields. Conversation group marks support up to 1,000 conversations, and up to 20 conversation groups can be added for each user. This API can be used to update conversation group data. Only the Pro edition 、 Pro Plus edition、 Enterprise edition supports conversation group data operations.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/recentcontact/create_contact_group?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/create_contact_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created

identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "user15",
  "GroupContactItem": [
    {
      "GroupName": "test0",
      "ContactItem": [
        {
          "Type": 1,
          "To_Account": "user0"
        }
      ]
    }
  ]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UID of the requester
GroupContactItem	Array	Yes	Conversation group to be added. Only one conversation group can be added at a time.
GroupName	String	Yes	Name of the conversation group to be added, which can contain up to 32 bytes
ContactItem	Array	Yes	List of conversations to be added
Type	Integer	Yes	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group

			conversation
ToGroupId	String	No	Group ID
To_Account	String	No	User ID

Sample response

```
{
  "GroupResultItem": [
    {
      "GroupItem": {
        "GroupName": "test2",
        "GroupId": 2
      },
      "ResultItem": [
        {
          "ContactItem": {
            "Type": 1,
            "To_Account": "user1"
          },
          "ResultCode": 0,
          "ResultInfo": ""
        }
      ]
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
GroupResultItem	Array	Result of the conversation group adding
GroupItem	Object	Conversation group object

GroupName	String	Conversation group name
GroupId	Integer	Conversation group ID
ResultItem	Array	Operation result
ContactItem	Integer	Conversation object
Type	Integer	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
ToGroupId	String	Group ID
To_Account	String	User ID
ResultCode	Integer	Error code in the conversation operation result. <code>0</code> : Successful
ResultInfo	String	Error description in the conversation operation result

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.
51006	When you are modifying conversation mark, the number of conversations is empty or exceeds the upper limit of 100.
51007	Failed to replace GroupID with GroupCode because an internal error occurred or the group was

	disbanded.
51008	The total number of conversations in the conversation group exceeds 1,000.
51009	The conversation group does not exist when a deletion attempt is made.
51010	The number of conversation groups exceeds the upper limit of 20.
51011	The conversation group name contains more than 32 bytes.
51012	Exceeded the maximum number of conversations pinned to the top.
51013	If the requested app is not of the Pro edition 、 Pro Plus edition、 Enterprise edition, an error will be reported when a standard conversation mark or conversation group is modified.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Updating Conversation Group Data

Last updated : 2025-01-14 14:35:32

## Feature Overview

Conversation group mark data is independent of recent contacts. You can use RESTful APIs to add, delete, modify, and query conversation group, standard conversation mark, custom conversation mark, and other fields. Conversation group marks support up to 1,000 conversations. This API can be used to update conversation group data. Only the Pro edition 、 Pro Plus edition、 Enterprise edition supports conversation group data operations.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/recentcontact/update_contact_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/update_contact_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .

usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "user20",
  "UpdateType": 1,
  "UpdateGroup": {
    "UpdateGroupType": 2,
    "OldGroupName": "test1",
    "ContactUpdateItem": [
      {
        "ContactOptType": 1,
        "ContactItem": {
          "Type": 1,
          "To_Account": "user1"
        }
      }
    ]
  }
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UID of the requester
UpdateType	Integer	Yes	<code>1</code> : Adding a conversation to or deleting a conversation from a group
UpdateGroup	Object	Yes	Adding a conversation to or deleting a conversation from a group
UpdateGroupType	Integer	Yes	Update type. <code>1</code> : Updating the group name; <code>2</code> Updating the conversation group



OldGroupName	String	Yes (when <code>UpdateGroupType</code> is 1)	Group name to be updated
NewGroupName	String	Yes (when <code>UpdateGroupType</code> is 1)	Updated group name, which can be up to 32 bytes
ContactUpdateItem	Array	No	Conversation to be updated
ContactOptType	Integer	Required	Update type. <code>1</code> : Adding a conversation to a group; <code>2</code> : Deleting a conversation from a group.
ContactItem	Object	Yes	Conversation object
Type	Integer	Yes	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
ToGroupId	String	Yes	Group ID
To_Account	String	Yes	User ID

## Sample response

```
{
  "UpdateType": 1,
  "UpdateGroupResult": {
    "UpdateGroupType": 2,
    "GroupName": "test1",
    "GroupId": 1,
    "OldGroupName": "test1",
    "ContactResultItem": [
      {
        "ContactOptType": 1,
        "ContactItem": {
          "Type": 1,
          "To_Account": "user1"
        },
        "ResultCode": 0,
        "ResultInfo": ""
      }
    ]
  },
  "ActionStatus": "OK",
  "ErrorCode": 0,
```

```
"ErrorInfo": ""
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
UpdateType	Integer	Required
UpdateGroupResult	Object	Group update result
UpdateGroupType	Integer	Yes
GroupName	String	Current group name
OldGroupName	String	Original group name
GroupId	Integer	Current group ID
ContactResultItem	Array	Conversation operation result
ContactOptType	Integer	Required
ContactItem	Object	Conversation object
Type	Integer	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
ToGroupId	String	Group ID
To_Account	String	User ID
ResultCode	Integer	Error code in the conversation operation result. <code>0</code> : Successful
ResultInfo	String	Error description in the conversation operation result

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.
51006	When you are modifying conversation mark, the number of conversations is empty or exceeds the upper limit of 100.
51007	Failed to replace GroupID with GroupCode because an internal error occurred or the group was disbanded.
51008	The total number of conversations in the conversation group exceeds 1,000.
51009	The conversation group does not exist when a deletion attempt is made.
51010	The number of conversation groups exceeds the upper limit of 20.
51011	The conversation group name contains more than 32 bytes.
51012	Exceeded the maximum number of conversations pinned to the top.
51013	If the requested app is not of the <a href="https://write.woa.com/document/108053313338515456">https://write.woa.com/document/108053313338515456</a> , an error will be reported when a standard conversation mark or conversation group is modified.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Deleting Conversation Group Data

Last updated : 2025-01-14 14:35:32

## Feature Overview

This API is used to delete users' conversation group data. Only the Pro edition 、 Pro Plus edition、 Enterprise edition supports conversation group data operations.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/recentcontact/del_contact_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/del_contact_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

```
{
  "From_Account": "user15",
  "GroupName": ["test0"]    // Currently, only one conversation group can be
                             deleted at a time
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	UID of the requester
GroupName	Array	Yes	List of conversation groups to be deleted. Currently, only one conversation group can be deleted at a time.

## Sample response

```
{
  "GroupItem": [
    {
      "GroupName": "test0",
      "GroupId": 2
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": ""
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed

ErrorCode	Integer	Error code. 0 : Successful; other values: Failed
ErrorInfo	String	Error information
GroupItem	Array	Group object to be deleted
GroupName	String	Name of the conversation group to be deleted
GroupId	Integer	ID of the conversation group to be deleted

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.
51006	When you are modifying conversation mark, the number of conversations is empty or exceeds the upper limit of 100.
51007	Failed to replace GroupID with GroupCode because an internal error occurred or the group was disbanded.
51008	The total number of conversations in the conversation group exceeds 1,000.
51009	The conversation group does not exist when a deletion attempt is made.
51010	The number of conversation groups exceeds the upper limit of 20.

51011	The conversation group name contains more than 32 bytes.
51012	Exceeded the maximum number of conversations pinned to the top.
51013	If the requested app is not of the Pro edition 、 Pro Plus edition、 Enterprise edition, an error will be reported when a standard conversation mark or conversation group is modified.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Creating or Updating Conversation Mark Data

Last updated : 2025-01-14 14:35:32

## Feature Overview

Conversation group mark data is independent of recent contacts. You can use RESTful APIs to add, delete, modify, and query conversation group, standard conversation mark, custom conversation mark, and other fields. Conversation marks support up to 1,000 conversations. This API can be used to create or update standard or custom conversation marks. Only the Pro edition 、 Pro Plus edition、 Enterprise edition supports standard conversation mark operations, and custom conversation mark operations have no such restrictions.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/recentcontact/mark_contact?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/mark_contact	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .



usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "user0",
  "MarkItem": [
    {
      "OptType": 3,
      "ContactItem": {
        "Type": 1,
        "To_Account": "user1"
      },
      "SetMark": [1, 2, 3],
      "CustomMark": "abcd"
    }
  ]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UID of the requester
MarkItem	Array	Yes	Array of conversation marks. The array can contain up to 100 conversation marks.
OptType	Integer	Yes	Operation type. <code>1</code> : Setting standard conversation marks; <code>2</code> : Setting custom conversation marks; <code>3</code> : Setting both standard and custom conversation marks
ContactItem	Object	Yes	Conversation object to be operated
Type	Integer	Yes	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation

ToGroupId	String	No	Group ID
To_Account	String	No	User ID
ClearMark	Array	No	Sequence number of the bit for which a standard mark is to be canceled. The value <code>n</code> indicates to cancel the mark value for bit <code>n</code> .
SetMark	Array	No	Sequence number of the bit for which a standard mark is to be set. The value <code>n</code> indicates to set the mark value for bit <code>n</code> .
CustomMark	String	No	Set a custom conversation mark. Up to 256 bytes are supported.

A standard conversation mark has 64 bits in total: the first 32 bits are internally used by Chat, and the last 32 bits are extended bits that can be customized by customers. Internal bits are described as follows:

Bit 0: Mark a conversation as "favorite"

Bit 1: Mark a conversation (important conversation) as unread

Bit 2: Mark a conversation as collapsed

Bit 3: Mark a conversation as hidden

Sample response

```
{
  "ResultItem": [
    {
      "OptType": 3,
      "ContactItem": {
        "Type": 1,
        "To_Account": "user1"
      },
      "ResultCode": 0,
      "ResultInfo": ""
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response fields

Field	Type	Description

ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
ResultItem	Array	Conversation mark operation result
OptType	Integer	Yes
ContactItem	Integer	Conversation object
Type	String	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
ToGroupId	String	Group ID
To_Account	String	Recipient's user ID
ResultCode	Integer	Error code in the conversation operation result. <code>0</code> : Successful
ResultInfo	String	Error description in the conversation operation result

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.

51006	When you are modifying conversation mark, the number of conversations is empty or exceeds the upper limit of 100.
51007	Failed to replace GroupID with GroupCode because an internal error occurred or the group was disbanded.
51008	The total number of conversations in the conversation group exceeds 1,000.
51009	The conversation group does not exist when a deletion attempt is made.
51010	The number of conversation groups exceeds the upper limit of 20.
51011	The conversation group name contains more than 32 bytes.
51012	Exceeded the maximum number of conversations pinned to the top.
51013	If the requested app is not of the Pro edition 、 Pro Plus edition、 Enterprise edition, an error will be reported when a standard conversation mark or conversation group is modified.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Searching for Conversation Group Marks

Last updated : 2025-01-14 14:35:32

## Feature Overview

This API is used to query the conversation group mark data of a specified conversation.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/recentcontact/search_contact_group?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/search_contact_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "yahaha",
  "ContactItem": [
    {
      "Type": 1,
      "To_Account": "xixi"
    }
  ]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UID of the requester
ContactItem	Array	List of conversation objects to be searched. Up to 300 conversations can be searched at a time.	
Type	String	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation	
ToGroupId	String	Group ID	
To_Account	String	User ID	

Sample response

```
{
  "ContactResultItem": [
    {
      "Type": 1,
      "To_Account": "xixi",
      "StandardMark": "11111110",
      "CustomMark": "xxyyyzzzffffffffxxx111111",
      "ContactGroupId": [

```

```
        1,
        2
    ],
    "Timestamp": 1673500546
  }
],
"GroupItem": [
  {
    "GroupName": "test1x",
    "GroupId": 1
  },
  {
    "GroupName": "test10",
    "GroupId": 2
  }
],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
ContactItem	Integer	Conversation object
Type	String	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
ToGroupId	String	Group ID
To_Account	String	User ID
Timestamp	Integer	Last update timestamp of the conversation group
StandardMark	String	Standard mark capability bit, for example, <code>011111110</code>
CustomMark	String	Custom mark
ContactGroupId	Array	Group to which the conversation belongs

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.
51006	When you are modifying conversation mark, the number of conversations is empty or exceeds the upper limit of 100.
51007	Failed to replace GroupID with GroupCode because an internal error occurred or the group was disbanded.
51008	The total number of conversations in the conversation group exceeds 1,000.
51009	The conversation group does not exist when a deletion attempt is made.
51010	The number of conversation groups exceeds the upper limit of 20.
51011	The conversation group name contains more than 32 bytes.
51012	Exceeded the maximum number of conversations pinned to the top.
51013	If the requested app is not of the Pro edition 、 Pro Plus edition、 Enterprise edition, an error will be reported when a standard conversation mark or conversation group is modified.

## API Debugging Tool



---

Use the [RESTful API online debugging tool](#) to debug this API.

# Pulling Conversation Group Mark Data

Last updated : 2025-01-14 14:35:32

## Feature Overview

This API is used to pull the conversation group mark data of users in batches.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/recentcontact/get_contact_group?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/recentcontact/get_contact_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "user0",
  "StartIndex": 0
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UID of the requester
StartIndex	Integer	Yes	Start index for the current page pulling. Enter <code>0</code> for the first pull and enter the value of <code>NextStartIndex</code> returned last time.

Sample response

```
{
  "ContactItem": [
    {
      "Type": 1,
      "To_Account": "teacher4",
      "StandardMark": "111001",
      "ContactGroupId": [
        1,
        2
      ],
      "Timestamp": 1670843110
    },
    {
      "Type": 2,
      "ToGroupId": "@TGS#1N3RSUYG2",
      "StandardMark": "111001",
      "CustomMark": "abcd",
      "Timestamp": 1672998266
    }
  ],
  "GroupItem": [
```

```
{
  "GroupName": "test1x",
  "GroupId": 1
},
{
  "GroupName": "test10",
  "GroupId": 2
}
],
"CompleteFlag": 1,
"NextStartIndex": 0,
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": ""
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
ContactItem	Integer	Conversation object
Type	String	Conversation type. <code>1</code> : One-to-one conversation; <code>2</code> : Group conversation
ToGroupId	String	Group ID
To_Account	String	User ID
Timestamp	Integer	Last update timestamp of the conversation group
StandardMark	String	Standard mark capability bit, for example, <code>011111110</code>
CustomMark	String	Custom mark
ContactGroupId	Array	Group to which the conversation belongs
CompleteFlag	Integer	Required
NextStartIndex	Integer	Required

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
50001	The requested UserID has not been imported into the Tencent Cloud Chat backend. Please import.
50002	Incorrect request parameter. Check your request according to the error description.
50003	The request requires app admin permissions.
50004	Internal server error. Please try again.
50005	Network timeout. Try again later.
51006	When you are modifying conversation mark, the number of conversations is empty or exceeds the upper limit of 100.
51007	Failed to replace GroupID with GroupCode because an internal error occurred or the group was disbanded.
51008	The total number of conversations in the conversation group exceeds 1,000.
51009	The conversation group does not exist when a deletion attempt is made.
51010	The number of conversation groups exceeds the upper limit of 20.
51011	The conversation group name contains more than 32 bytes.
51012	Exceeded the maximum number of conversations pinned to the top.
51013	If the requested app is not of the Pro edition 、 Pro Plus edition、 Enterprise edition, an error will be reported when a standard conversation mark or conversation group is modified.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Group Related

## Group Management

### Getting All Groups in an App

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to obtain the IDs of all groups in an app.

## API Calling Description

### Applicable group types

Tencent Cloud Chat provides different types of built-in groups. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_appid_group_list?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/group_open_http_svc/get_appid_group_list	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is

	created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

1 times/second

## Sample request

### Basic format

If the total number of groups in the app exceeds 10,000, a maximum of 10,000 group IDs are returned. To obtain all group IDs, you must pull them by page.

```
{ }
```

### Pulling by page

You can use the `Limit` and `Next` fields to control the paged pulling:

The `Limit` field specifies the maximum number of groups in `GroupIdList` in the response packet, which cannot exceed 10,000.

The `Next` field is used to control pagination. For the initial pagination request, `Next` is set to 0. For subsequent requests, it is set to the previously returned `Next` value. If the returned `Next` value is 0, all groups have been pulled.

For example, if paged pulling is requested and 20 group IDs are displayed per page, the request parameters for the first page are `{ "Limit" : 20, "Next" : 0 }`, whereas those for the second page are `{ "Limit" : 20, "Next" : Previously returned Next value }`, and so on.

The value of `Limit` or `Next` does not affect `TotalCount` in the response packet.

```
{
  "Limit": 1000,
  "Next": 0
}
```

### Specifying the group type

You can specify the type of groups to be pulled, such as Public, Private, ChatRoom, AVChatRoom, or BChatRoom.

```
{
  "GroupType" : "Public" // Type of groups to be pulled. If this parameter is
not specified, all types of groups will be pulled.
}
```

## ALL IN ONE

```
{
  "Limit": 1000,
  "Next": 0,
  "GroupType" : "Public" // Type of groups to be pulled. If this parameter is
not specified, all types of groups will be pulled.
}
```

## Request fields

Field	Type	Required	Description
Limit	Integer	No	Maximum number of group IDs to be obtained, which cannot exceed 10,000. If no value is specified, the maximum value 10000 is used by default.
Next	Integer	No	Paged pulling flag when the number of groups is too large. It is initially set to 0 and subsequently to the <code>Next</code> value returned in the previous response. If the returned <code>Next</code> value is 0, all groups have been pulled.
GroupType	String	No	To obtain a specified type of groups, you can use <code>GroupType</code> for filtering. In this case, the returned <code>TotalCount</code> value indicates the total number of groups of the specified type in the app. If this field is not specified, all types of groups are obtained. Possible group types are Public, Private, ChatRoom, AVChatRoom, BChatRoom, and Community.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TotalCount": 2,
  "GroupIdList": [
    {
      "GroupId": "@TGS#2J4SZEAE"
    },
  ],
}
```



```
{
  "GroupId": "@TGS#2C5SZEAEF"
},
"Next": 4454685361
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
TotalCount	Integer	Total number of existing groups in the app. To obtain a specified type of groups, you can use <code>GroupType</code> for filtering. In this case, the returned <code>TotalCount</code> value indicates the total number of groups of the specified type in the app. If this field is not specified, all types of groups are obtained. For example, assume that the app has total 50,000 groups, including 20,000 public groups. If <code>GroupType</code> in the request packet is set to <code>Public</code> , <code>TotalCount</code> in the response packet is 20,000 regardless of the <code>Limit</code> and <code>Offset</code> values. In addition, groups in <code>GroupIdList</code> are all public groups.
GroupIdList	Array	List of obtained group IDs.
Next	Integer	Paged pulling flag.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	A parameter is incorrect. To correct it, check request parameters such as <code>GroupType</code> based on the error description.

10007	The operator does not have the necessary permissions for this operation. Only the app admin can call this API.
10018	The response packet exceeds the length limit of 1 MB due to excessive request content. Try to reduce the amount of data in individual single requests.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Getting the Groups a User Has Joined ([v4/group\\_open\\_http\\_svc/get\\_joined\\_group\\_list](#))

# Creating a Group

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to create groups.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes. To use an audio-video group (AVChatroom), users (including the group owner) must use the SDK to request to join the group.
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

If the group type is specified as audio-video group (AVChatRoom):

When creating a group of this type, the group creator cannot add users to the group. If a group member is specified during the group creation, error 10007 will be returned. Users can only request to join this type of group. If a group owner is specified, the group owner must use the SDK API to join the group every time he or she uses the group.

If the group type is specified as a type other than an audio-video group (AVChatRoom):

If neither a group owner nor a group member list is specified when the request is sent, the number of groups that can be created is unrestricted.

If a group owner or group member list is specified when the request is sent, the specified group owner or group members are automatically added to the group.

If the total number of groups in the app exceeds 100,000, certain fees will be charged. For more information, see [Pricing](#).

The number of groups that a user can join at the same time is limited. If the number is reached, joining or creating a group will fail. For more information, see [Pricing](#).

## Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/create_group?sdkappid=88888888&identifier=adm
```

## Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/create_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Create a group. The `Owner_Account` field is optional. If it is not specified, the group will have no group owner. If you want to specify a group owner, ensure that the UserId of the group owner [has been imported](#).

```
{
  "Owner_Account": "leckie", // UserId of the group owner (optional)
  "Type": "Public", // Group type: Private, Public, ChatRoom, AVChatRoom, or Commun
  "Name": "TestGroup" // Group name (required)
}
```

### Containing only basic group information

Create a group and specify basic group information, such as the group introduction and group notice.

```
{
  "Owner_Account": "leckie", // UserId of the group owner (optional)
  "Type": "Public", // Group type: Private, Public, ChatRoom, AVChatRoom, or Commun
  "Name": "TestGroup", // Group name (required)
  "Introduction": "This is group Introduction", // Group introduction (optional)
  "Notification": "This is group Notification", // Group notice (optional)
  "FaceUrl": "http://this.is.face.url", // Group profile photo URL (optional)
  "MaxMemberCount": 500, // Maximum number of group members (optional)
  "ApplyJoinOption": "FreeAccess" // Method for handling requests to join the grou
}
```

### Containing only group member information

Create a group and specify to initialize the group member list, which is described in the request packet description table. When group members are initialized, only the `Member_Account` and `Role` fields can be set.

```
{
  "Name": "TestGroup", // Group name (required)
  "Type": "Public", // Group type: Private, Public, ChatRoom, or Community (AVChatR
  "MemberList": [ // Initial group member list, which contains a maximum of 100 mem
    {
      "Member_Account": "bob", // Member (required)
      "Role": "Admin" // Role assigned to the member (optional). Currently, onl
    },
    {
      "Member_Account": "peter"
    }
  ]
}
```

### Custom group ID

To simplify group IDs, Tencent Cloud allows apps to customize group IDs when creating groups through the RESTful API.

```
{
  "Owner_Account": "leckie", // UserId of the group owner (optional)
  "Type": "Public", // Group type: Private, Public, ChatRoom, AVChatRoom, or Commun
  "GroupId": "MyFirstGroup", // User-defined group ID (optional)
}
```

```
"Name": "TestGroup"    // Group name (required)
}
```

### Containing only custom group information

Create a group and specify group custom fields. The `AppDefineData` field is unavailable by default and needs to be enabled in the [Chat console](#) before use. For details, see the request packet field description table.

```
{
  "Name": "TestGroup", // Group name (required)
  "Type": "Public", // Group type: Private, Public, ChatRoom, AVChatRoom, or Commun
  "AppDefinedData": [ // Group custom field (optional)
    {
      "Key": "GroupTestData1", // Key of the app custom field
      "Value": "xxxxx" // Value of the custom field
    },
    {
      "Key": " GroupTestData2",
      "Value": "abc\\u0000\\u0001" // The custom field supports binary data.
    }
  ]
}
```

### Containing only custom group member information

Create a group and specify group member custom fields. By default, `AppMemberDefinedData` is not available and needs to be enabled in the [Chat console](#) before use. For details, see the table for request packet fields.

```
{
  "Owner_Account": "leckie", // UserId of the group owner (optional)
  "Type": "Public", // Group type: Private, Public, ChatRoom, or Community (AVChatR
  "Name": "TestGroup", // Group name (required)
  "MemberList": [
    {
      "Member_Account": "bob",
      "AppMemberDefinedData": [ // Group member custom field (optional)
        {
          "Key": "MemberDefined1", // Group member custom key
          "Value": "MemberData1" // Value of the group member custom field
        },
        {
          "Key": "MemberDefined2",
          "Value": "MemberData2"
        }
      ]
    },
    {
      "Member_Account": "peter",
```

```

        "AppMemberDefinedData": [
            {
                "Key": "MemberDefined1",
                "Value": "MemberData1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "MemberData2"
            }
        ]
    }
]
}

```

### Containing only the topic-enabled parameter format

Create a group and specify the type as a topic-enabled community.

```

{
    "Type": "Community", // Group type (required)
    "Name": "TestCommunityGroup", // Group name (required)
    "SupportTopic": 1 // Whether the topic option is supported. V
}

```

### ALL IN ONE

```

{
    "Owner_Account": "leckie", // UserId of the group owner (optional)
    "Type": "Public", // Group type: Private, Public, ChatRoom, or Community (AVChatR
    "GroupId": "MyFirstGroup", // User-defined group ID (optional)
    "Name": "TestGroup", // Group name (required)
    "Introduction": "This is group Introduction", // Group introduction (optional)
    "Notification": "This is group Notification", // Group notice (optional)
    "FaceUrl": "http://this.is.face.url", // Group profile photo URL (optional)
    "MaxMemberCount": 500, // Maximum number of group members (optional)
    "ApplyJoinOption": "FreeAccess", // Method for handling requests to join the grou
    "AppDefinedData": [ // Group custom field (optional)
        {
            "Key": "GroupTestData1", // Key of the app custom field
            "Value": "xxxxx" // Value of the custom field
        },
        {
            "Key": "GroupTestData2",
            "Value": "abc\\u0000\\u0001" // The custom field supports binary data.
        }
    ],
    "MemberList": [ // Initial group member list, which contains a maximum of 100 mem
        {

```

```

"Member_Account": "bob", // Member (required)
"Role": "Admin", // Role assigned to the member. Currently, only the Admin role is supported.
"AppMemberDefinedData": [ // Group member custom field (optional)
  {
    "Key": "MemberDefined1", // Group member custom key
    "Value": "MemberData1" // Value of the group member custom field
  },
  {
    "Key": "MemberDefined2",
    "Value": "MemberData2"
  }
],
},
{
  "Member_Account": "peter",
  "AppMemberDefinedData": [
    {
      "Key": "MemberDefined1",
      "Value": "MemberData1"
    },
    {
      "Key": "MemberDefined2",
      "Value": "MemberData2"
    }
  ]
}
]
}

```

## Request fields

Field	Type	Required	Description
Owner_Account	String	Optional	Group owner ID, which must be an <a href="#">imported</a> account. The value will be automatically added to group members. If this field is not specified, the group will have no group owner, and group members need to call the group joining API when using an audio-video group (AVChatroom) group.
Type	String	Required	Group type, including Public, Work (Private), Meeting, AVChatRoom, and Community.
GroupId	String	Optional	To simplify group IDs and make them easier to remember and share, we allow apps to <a href="#">customize group IDs</a> when creating groups through the RESTful API.



Name	String	Required	Group name, whose maximum length is 30 bytes. This field is UTF-8-encoded, and one Chinese character occupies 3 bytes.
Introduction	String	Optional	Group introduction, whose maximum length is 240 bytes. This field is UTF-8-encoded, and one Chinese character occupies 3 bytes.
Notification	String	Optional	Group notice, whose maximum length is 300 bytes. This field is UTF-8-encoded, and one Chinese character occupies 3 bytes.
FaceUrl	String	Optional	URL of the group profile photo, whose maximum length is 100 bytes.
MaxMemberCount	Integer	Optional	Maximum number of group members. The default value is the upper limit of the paid package. For example, it is 20 for a Developer edition. If you upgrade your package, you need to modify this field according to the basic information of the modified group.
ApplyJoinOption	String	Optional	<p>Method for handling requests to join the group, which can be <code>FreeAccess</code> , <code>NeedPermission</code> , or <code>DisableApply</code> . The default method is <code>NeedPermission</code> .</p> <p>This field is valid only for <a href="#">groups</a> that can be joined by users. Community groups currently do not support this field.</p>
AppDefinedData	Array	Optional	Group-level custom field. This field is unavailable by default and needs to be enabled in the <a href="#">Chat console</a> . For details, see <a href="#">Custom Fields</a> .
MemberList	Array	Optional	Initial group member list, which contains a maximum of 100 members. For details on group member information fields, see <a href="#">Group member profile</a> .
AppMemberDefinedData	Array	Optional	Group member custom fields. This field is unavailable by default and needs to be enabled in the <a href="#">Chat console</a> . For details, see <a href="#">Custom Fields</a> .
SupportTopic	Integer	No	Whether the topic option is supported. This field is valid only for communities.

## Sample response

Basic form, containing only basic group information, containing only group member information, or containing only custom information

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "@TGS#2J4SZEAEEL"
}
```

Containing only the topic-enabled parameter format

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "@TGS#_@TGS#cRDH3HIM62CP",
  "HugeGroupFlag": 0,
  "Type": "Community"
}
```

Custom group ID and ALL IN ONE

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "MyFirstGroup"
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
GroupId	String	Group ID after successful creation, which is assigned by the Chat backend.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid request command word.
10004	Invalid parameter. Check the error description and fix the issue.
10005	The number of members imported in the request packet exceeds 100. Reduce the number of members imported in the <code>MemberList</code> parameter.
10006	The operation exceeds the frequency limit. Please reduce the call frequency. This error is usually caused by too much net group increase in a single day or too frequent calls to obtain all groups in the app.
10007	Insufficient operation permission. Check the request parameters based on the error information. For example, the specified group type does not allow adding members but <code>MemberList</code> was specified in the request.
10008	The request is invalid, probably because verification of the signing information in the request failed. Please try again or <a href="#">submit a ticket</a> .
10016	The app backend rejected this operation through a webhook. Check the returned value of your webhook "Before a group is created".
10021	The group ID has already been used by another user. Select another group ID.
10025	You have already used this group ID. Disband the existing group first or select another group ID.
10036	The number of audio-video groups (AVChatRoom) created exceeds the limit. Delete some of the groups or purchase an upgrade by referring to <a href="#">Pricing</a> .
10037	The request specifies Owner_Account, but the number of groups that the group owner has created or joined exceeds the limit. The group owner needs to leave some groups first or refer to <a href="#">Pricing</a> to purchase an upgrade.
10038	The number of members imported in the request packet exceeds the limit. Reduce the number of members imported in the <code>MemberList</code> parameter or purchase an upgrade by referring to <a href="#">Pricing</a> .
10058	You are now using the Developer edition, and the free quota of 100 groups is exceeded. To create more groups, you need to purchase a package.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Disbanding a group ([v4/group\\_open\\_http\\_svc/destroy\\_group](#))

## Possible Callbacks

[Before a Group Is Created](#)

[After a Group Is Created](#)

# Disbanding a Group

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to disband a group.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/destroy_group?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/destroy_group</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

The group disbanding operation is irrevocable. Please call this API with caution.

```
{
  "GroupId": "@TGS#2J4SZEDEL"
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the Group whose historical messages will be modified

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Creating a Group ([v4/group\\_open\\_http\\_svc/create\\_group](#))

## Possible Webhooks

[After a Group Is Disbanded](#)。



# Getting the Groups a User Has Joined

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by app admins to obtain the list of groups a certain user has joined. The information of work groups (Work) and audio-video groups (AVChatRoom) that the user has joined but are not activated is not pulled by default.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version. The information of work groups that the user has joined but are not activated is not returned by default.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes. The information of audio-video groups (AVChatRoom) that the user has joined is not returned by default. If you specify to pull audio-video group information, the obtained group information may not be complete. Audio-video groups do not store all group member data.
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_joined_group_list?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/get_joined_group_list	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

A basic request is used to obtain the information about groups that a user has joined. The group information contains only the IDs of users in the group.

```
{
  "Member_Account": "leckie"
}
```

### Pulling by page

You can use the `Limit` and `Offset` fields to control the pagination mode:

The `Limit` field specifies the maximum number of groups in `GroupIdList` in the response packet, which cannot exceed 5,000.

The `Offset` field specifies the sequence number of the group in the entire group list, starting from which the information is read. Offset is 0 by default. If pagination is requested (the page number starts from 1), the `Offset`

value of each page should be: (Page number – 1) × Number of groups displayed on each page.

For example, to display 20 group members on each page, the request parameters for the first page should be

```
{"Limit": 20, "Offset": 0}
```

, the request parameters for the second page should be 

```
{"Limit": 20,
```

```
"Offset": 20}
```

, and so on.

The value of `Limit` or `Offset` does not affect `TotalCount` in the response packet.

```
{
  "Member_Account": "leckie",
  "Limit": 10, // Number of groups to be pulled. If this field is not
  specified, all groups will be pulled.
  "Offset": 0 // Sequence number of the group starting from which information
  is pulled.
}
```

### Specifying the group type

You can specify the type of groups to be pulled, for example, Public (social networking group for strangers), Private (same as Work groups in the new version), and ChatRoom (same as Meeting groups in the new version). If AVChatRoom (audio-video group) is specified, you may obtain an incomplete list of members.

```
{
  "Member_Account": "leckie",
  "GroupType" : "Public" // Type of groups to be pulled. If this parameter is
  not specified, all types of groups will be pulled.
}
```

### Pulling specified information

To specify the basic information fields to be pulled, set `GroupBaseInfoFilter`.

To specify the group member information in the group that needs to be pulled, set `SelfInfoFilter`.

```
{
  "Member_Account": "leckie",
  "WithHugeGroups":1, // Supports pulling the information of audio-video groups
  (AVChatRoom).
  "WithNoActiveGroups":1, // Supports pulling the information of Private groups
  (same as Work groups in the new version) that the user has joined but are not
  activated.
  "Limit": 10, // Number of groups to be pulled. If this field is not
  specified, all groups will be pulled.
  "Offset": 0, // Sequence number of the group starting from which information
  is pulled.
  "ResponseFilter": {
    "GroupBaseInfoFilter": [ // Basic information fields to be pulled
      "Type",
      "Name",
      "Introduction",

```

```

        "Notification"
    ],
    "SelfInfoFilter": [ // Member's personal information in the group
        "Role", // Role in the group
        "JoinTime" // Time when the member joined the group
    ]
}
}

```

### Pulling topic-enabled community groups

```

{
    "Member_Account": "107867", // User account to be queried (required)
    "SupportTopic": 1 // Whether the specified group type supports topics. This
    field is supported only by community groups.
}

```

### ALL IN ONE

```

{
    "Member_Account": "leckie",
    "WithHugeGroups": 1,
    "WithNoActiveGroups": 1,
    "ResponseFilter": {
        "GroupBaseInfoFilter": [
            "Type",
            "Name",
            "Introduction",
            "Notification",
            "FaceUrl",
            "CreateTime",
            "Owner_Account",
            "LastInfoTime",
            "LastMsgTime",
            "NextMsgSeq",
            "MemberNum",
            "MaxMemberNum",
            "ApplyJoinOption",
            "MuteAllMember"
        ],
        "SelfInfoFilter": [
            "Role",
            "JoinTime",
            "MsgFlag",
            "MsgSeq"
        ]
    }
}

```

```
}
}
```

## Request fields

Field	Type	Required	Description
Member_Account	String	Yes	User account to be queried
WithHugeGroups	Integer	No	Whether to pull the information of audio-video groups (AVChatRoom) that the user has joined. <code>0</code> : No (default value); <code>1</code> : Yes
WithNoActiveGroups	Integer	No	Whether to pull the information of Private groups (same as Work groups in the new version) that the user has joined but are not activated. <code>0</code> : No (default value); <code>1</code> : Yes
Limit	Integer	No	Number of groups pulled each time. If no value is specified, all groups will be pulled.
Offset	Integer	No	Sequence number of the group starting from which information is pulled.
GroupType	String	No	Type of groups to be pulled, for example, Public (social networking group for strangers), Private (same as Work groups in the new version), ChatRoom (same as Meeting groups in the new version), AVChatRoom (audio-video groups), and Community. If no value is specified, all types of groups are pulled.
ResponseFilter	Object	No	It contains two filters: <code>GroupBaseInfoFilter</code> and <code>SelfInfoFilter</code> . <code>GroupBaseInfoFilter</code> specifies the basic information fields to pull. For more information, see <a href="#">Group basic information</a> . <code>SelfInfoFilter</code> specifies members' personal information to be pulled from groups. For more information, see <a href="#">Group member profile</a> .
SupportTopic	Integer	No	Whether the specified group type supports topics. <code>1</code> : Yes; <code>0</code> : No. If this field is specified, <code>GroupType</code> must be <code>Community</code> .

## Sample response

### Basic or paged pull request

```
{
```

```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0,
"TotalCount": 2, // This value is the total number of groups that meet the
conditions, regardless of the `Limit` and `Offset` settings.
"GroupIdList": [
  {
    "GroupId": "@TGS#2J4SZEAEEL"
  },
  {
    "GroupId": "@TGS#2C5SZEAEF"
  }
]
```

### Specifying the group type

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TotalCount": 1,
  "GroupIdList": [
    {
      "GroupId": "@TGS#2J4SZEAEEL"
    }
  ]
}
```

### Pulling specified information

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TotalCount": 2,
  "GroupIdList": [
    {
      "GroupId": "@TGS#16UMONKGG",
      "Introduction": "",
      "Name": "d",
      "Notification": "",
      "SelfInfo": {
        "JoinTime": 1588148506,
        "Role": "Member"
      },
      "Type": "Private"
    }
  ]
}
```

```

    },
    {
      "GroupId": "@TGS#3FCOX2MGW",
      "Introduction": "",
      "Name": "TestGroup",
      "Notification": "",
      "SelfInfo": {
        "JoinTime": 1588041114,
        "Role": "Member"
      },
      "Type": "ChatRoom"
    }
  ]
}

```

### Pulling topic-enabled community groups

```

{
  "ActionStatus": "OK",
  "ErrorInfo": "ok",
  "ErrorCode": 0,
  "TotalCount": 1,
  "GroupIdList": [
    {
      "GroupId": "@TGS#_@TGS#cMOQ7HIM62CD",
      "Type": "Community",
      "SupportTopic": 1,
      "GrossTopicNextMsgSeq": 3,
      "SelfInfo": {
        "GrossTopicReadSeq": 2
      }
    }
  ]
}

```

### ALL IN ONE

```

{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TotalCount": 1, // This value is the total number of groups that meet the
conditions, regardless of the `Limit` and `Offset` settings.
  "GroupIdList": [
    {
      "ApplyJoinOption": "DisableApply",

```

```
"CreateTime": 1585718204,
"FaceUrl": "",
"GroupId": "@TGS#16UMONKGG",
"Introduction": "",
"LastInfoTime": 1588148506,
"LastMsgTime": 0,
"MaxMemberNum": 200,
"MemberNum": 1,
"Name": "d",
"NextMsgSeq": 2,
"Notification": "",
"Owner_Account": "",
"SelfInfo": {
  "JoinTime": 1588148506,
  "MsgFlag": "AcceptAndNotify",
  "Role": "Member",
  "MsgSeq": 1
},
"MuteAllMember": "Off",
"Type": "Private"
}
]
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
TotalCount	Integer	Number of groups that the user has joined
GroupIdList	Array	Group information pulled. The returned result is the information filtered based on the filtering fields set in the filter. For more information on the fields, see <a href="#">Group Data Structure</a> .

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.



For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	A system error occurred. Try again or contact technical support.
10003	The request command is invalid. Try again or contact technical support.
10004	A parameter is invalid. Check the <code>ErrorInfo</code> field in the response packet to see whether the required fields have been specified or whether the fields are set according to protocol requirements.
10018	The response packet length exceeds the maximum packet length of 1 MB because too much content was requested. Try to reduce the amount of data requested at a time.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Group Information

## Getting Group Profiles

Last updated : 2024-07-16 11:15:59

### Feature Overview

This API is used by the app admin to query the detailed group information based on one or more group IDs.

### API Calling Description

#### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

#### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_info?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located.

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_group_info</code>	The request API that is called.
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

The following example is used to obtain the detailed group information, including all the basic group information, group member information, and custom information. The request contains only a list of group IDs. Each request supports the query for information on a maximum of 50 groups.

```
{
  "GroupIdList": [ // The list of group IDs specified for the query. This
    parameter is required.
    "@TGS#1NVTZEAE4",
    "@TGS#1CXTZEAE4"
  ]
}
```

### Pulling specified information

To specify the basic information fields to be pulled, set `GroupBaseInfoFilter`.

To specify the member information fields to be pulled, set `MemberInfoFilter`.

To specify the group-specific custom fields to be pulled, set `AppDefinedDataFilter_Group`. This field is unavailable by default and must be enabled additionally. For more information, see Request packet fields.

To specify the group's member-specific custom fields to be pulled, set `AppDefinedDataFilter_GroupMember`. This field is unavailable by default and must be enabled additionally. For more information, see Request packet fields. If filters are set, only the fields specified by filters are pulled.

```
{
  "GroupIdList": [ // The list of group IDs specified for the query. This
parameter is required.
    "@TGS#1NVTZEAE4",
    "@TGS#1CXTZEAE4"
  ],
  "ResponseFilter": { // The filters that you specify for the response.
    "GroupBaseInfoFilter": [ // Add this array if you want to query the
basic information.
      "Type",
      "Name",
      "Introduction",
      "Notification"
    ],
    "MemberInfoFilter": [ // Add this array if you want to query the member
information.
      "Account", // Member ID
      "Role"
    ],
    "AppDefinedDataFilter_Group": [ // The filter for group-specific custom
fields. This filter is disabled by default.
      "GroupTestData1",
      "GroupTestData2"
    ],
    "AppDefinedDataFilter_GroupMember": [ // The filter for group member-
specific custom fields. This filter is disabled by default.
      "MemberDefined2",
      "MemberDefined1"
    ]
  }
}
```

## Request fields

Field	Type	Required	Description
GroupIdList	Array	Required	The list of group IDs that you specify for the query.
ResponseFilter	Object	Optional	It contains three filters: <code>GroupBaseInfoFilter</code> ,

			<code>MemberInfoFilter</code> , and <code>AppDefinedDataFilter_Group</code> .
<code>GroupBaseInfoFilter</code>	Array	Optional	Basic information field filter, which specifies the basic information fields to query. For more information, see <a href="#">Group basic data</a> .
<code>MemberInfoFilter</code>	Array	Optional	Member information field filter, which specifies the member information fields to query. For more information, see <a href="#">Group member information</a> .
<code>AppDefinedDataFilter_Group</code>	Array	Optional	The group-specific custom field filter, which specifies the group-specific custom fields to query. For more information, see <a href="#">Custom fields</a> .
<code>AppDefinedDataFilter_GroupMember</code>	Array	Optional	The group member-specific custom field filter, which specifies the group member-specific custom fields to be query. For more information, see <a href="#">Custom fields</a> .

## Sample response

### Basic format

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "", // The ErrorInfo parameter here is meaningless. You need to
check the ErrorInfo parameter of each group.
  "ErrorCode": 0, // The ErrorCode parameter here is meaningless. You need to
check the ErrorCode parameter of each group.
  "GroupInfo": [ // A group information array is returned. Only one group is
listed here for simplicity.
    {
      "GroupId": "@TGS#2J4SZEAE",
      "ErrorCode": 0, // The result specific to this group
      "ErrorInfo": "" , // The result specific to this group
      "Type": "Public", // Group type
      "Name": "MyFirstGroup", // Group name
      "Appid":1400001001,// SDKAppID of the Chat app
      "Introduction": "TestGroup", // Group introduction
      "Notification": "TestGroup", // Group notice
      "FaceUrl": "http://this.is.face.url", // Group profile photo
      "Owner_Account": "leckie", // Group owner ID
      "CreateTime": 1426976500, // Group creation time in UTC
    }
  ]
}
```

```

    "LastInfoTime": 1426976500, // The UTC time when the group
information was last updated
    "LastMsgTime": 1426976600, // The UTC time when the last message in
the group was sent
    "NextMsgSeq": 1234,
    "MemberNum": 2, // Current number of members in the group
    "MaxMemberNum": 50, // Maximum number of members in the group
    "ApplyJoinOption": "FreeAccess", // Method of handling requests to
join the group
    "MuteAllMember": "On", // Whether to mute all members in the group
    "AppDefinedData": [ // Group-specific custom fields
        {
            "Key": "GroupTestData1", // Key of the custom field
            "Value": "xxxx" // Value of the custom field
        },
        {
            "Key": "GroupTestData2",
            "Value": "abc\\u0000\\u0001" // The custom field supports
binary data.
        }
    ],
    "MemberList": [ // Group member list
        {
            "Member_Account": "leckie", // Member ID
            "Role": "Owner", // The role of the member in the group
            "JoinTime": 1425976500, // The UTC time when the member
joined the group
            "MsgSeq": 1233,
            "MsgFlag": "AcceptAndNotify", // Indicates whether the member
blocks group messages.
            "LastSendMsgTime": 1425976500, // The last time in UTC when
the member sent a message in the group
            "MuteUntil": 1431069882, // UTC time when the muting period
expires
            "AppMemberDefinedData": [ // Custom group member fields
                {
                    "Key": "MemberDefined1",
                    "Value": "ModifyDefined1"
                },
                {
                    "Key": "MemberDefined2",
                    "Value": "ModifyDefined2"
                }
            ]
        },
        {
            "Member_Account": "peter",

```

```

        "Role": "Member",
        "JoinTime": 1425976500, // Time when the member joined the
group
        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500, // Last time when the member
sent a message
        "MuteUntil": 0, // The value `0` indicates that the member is
not muted, and other values indicate the time when the member will be unmuted.
        "AppMemberDefinedData": [ // Custom group member fields
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    }
}
]
}
}

```

### Pulling specified information

```

{
    "ActionStatus": "OK",
    "ErrorInfo": "", // The ErrorInfo parameter here is meaningless. You need to
check the ErrorInfo parameter of each group.
    "ErrorCode": 0, // The ErrorCode parameter here is meaningless. You need to
check the ErrorCode parameter of each group.
    "GroupInfo": [ // A group information array is returned. Only one group is
listed here for simplicity.
        {
            "GroupId": "@TGS#2J4SZEAE",
            "ErrorCode": 0, // The result specific to this group
            "ErrorInfo": "" , // The result specific to this group
            "Type": "Public", // Group type
            "Name": "MyFirstGroup", // Group name
            "Introduction": "TestGroup", // Group introduction
            "Notification": "TestGroup", // Group notice
            "AppDefinedData": [ // Group-specific custom fields
                {
                    "Key": "GroupTestData1", // Key of the custom field
                    "Value": "xxxx" // Value of the custom field
                }
            ]
        }
    ]
}

```

```
    },
    {
        "Key": "GroupTestData2",
        "Value": "abc\\u0000\\u0001" // The custom field supports
binary data.
    }
],
"MemberList": [ // Group member list
    {
        "Member_Account": "leckie", // Member ID
        "Role": "Owner", // The role of the member in the group
        "AppMemberDefinedData": [ // Custom group member fields
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    }
],
{
    "Member_Account": "peter",
    "Role": "Member",
    "AppMemberDefinedData": [ // Custom group member fields
        {
            "Key": "MemberDefined1",
            "Value": "ModifyDefined1"
        },
        {
            "Key": "MemberDefined2",
            "Value": "ModifyDefined2"
        }
    ]
}
]
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed



ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
GroupInfo	Array	Group information array returned, including basic group information fields, group member information fields, group-specific custom fields, and group member-specific custom fields. For more information on the fields, see <a href="#">Group Data Structure</a> .

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	A system error occurred. Try again or contact technical support.
10003	The request command is invalid. Try again or contact technical support.
10004	A parameter is invalid. Check the <code>ErrorInfo</code> field in the response packet to see whether the required fields have been specified or whether the fields are set according to protocol requirements.
10007	Insufficient operation permissions. Check whether the operator is the app admin.
10010	The group does not exist or has been deleted.
10015	The group ID is invalid. Please check the group ID.
10018	The response packet length exceeds the maximum packet length of 1 MB because too much content was requested. Try to reduce the amount of data requested at a time.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Getting Group Member Profiles ([v4/group\\_open\\_http\\_svc/get\\_group\\_member\\_info](#))

---

Modifying the Profile of a Group ([v4/group\\_open\\_http\\_svc/modify\\_group\\_base\\_info](#))

Modifying the Profile of a Group Member ([v4/group\\_open\\_http\\_svc/modify\\_group\\_member\\_info](#))

# Modifying the Profile of a Group

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to modify the profile of a specified group.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/modify_group_base_info?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/modify_group_base_info</code>	Request API.
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Modify basic group information

Modify basic group information, such as the group name and group announcement.

```
{
  "GroupId": "@TGS#1NVTZEAE4", // Group whose basic profile you want to modify
  (required)
  "Name": "NewName", // Group name (optional)
  "Introduction": "NewIntroduction", // Group introduction (optional)
  "Notification": "NewNotification", // Group notice (optional)
  "FaceUrl": "http://this.is.new.face.url", // Group profile photo (optional)
  "MaxMemberNum": 500, // Maximum number of group members (optional)
  "ApplyJoinOption": "NeedPermission", // Method for applying to join the group
  (optional)
  "MuteAllMember": "On" // Mutes all members (optional). "On": Enable, "Off":
  Disable
}
```

### Set custom group information

Set a group's custom field information. By default, custom information is not available and needs to be enabled in the

[Chat console](#). For details, see the description table for request packet fields.

```
{
  "GroupId": "@TGS#1NVTZEAE4", // Group whose basic profile you want to modify
  (required)
  "AppDefinedData": [ // Custom field (optional)
    {
      "Key": "GroupTestData1", // Custom field key to be modified
      "Value": "NewData" // New value of the custom field
    }
  ]
}
```

### Delete custom group information

Delete a group's custom field information that you have set.

```
{
  "GroupId": "@TGS#1NVTZEAE4", // Group whose basic profile you want to modify
  (required)
  "AppDefinedData": [ // Custom field (optional)
    {
      "Key": "GroupTestData2",
      "Value": "" // If this parameter is empty, the custom field is to be
deleted.
    }
  ]
}
```

### ALL IN ONE

```
{
  "GroupId": "@TGS#2J4SZEAE4", // Group whose basic profile you want to modify
  (required)
  "Name": "NewName", // Group name (optional)
  "Introduction": "NewIntroduction", // Group introduction (optional)
  "Notification": "NewNotification", // Group notice (optional)
  "FaceUrl": "http://this.is.new.face.url", // Group profile photo (optional)
  "MaxMemberNum": 500, // Maximum number of group members (optional)
  "ApplyJoinOption": "NeedPermission", // Method for applying to join the group
  (optional)
  "MuteAllMember": "On", // Mutes all members, which is optional. `On`: Enable.
  `Off`: Disable
  "AppDefinedData": [ // Custom field (optional)
    {
      "Key": "GroupTestData1", // Custom field key to be modified
      "Value": "NewData" // New value of the custom field
    },
  ],
}
```

```

{
  "Key": "GroupTestData2",
  "Value": "" // If this parameter is empty, the custom field is to be
deleted.
}
]
}

```

## Request fields

Field	Type	Required	Description
GroupId	String	Required	ID of the group whose basic information you want to modify.
Name	String	Optional	The maximum length is 30 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
Introduction	String	Optional	The maximum length is 240 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
Notification	String	Optional	The maximum length is 300 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
FaceUrl	String	Optional	URL of the group profile photo, whose maximum length is 100 bytes.
MaxMemberNum	Integer	No	Maximum number of group members For work groups (Work), public groups (Public), meeting groups (Meeting), and communities, see the upper limit on the number of group members in <a href="#">Group System</a> . For audio-video groups (AVChatRoom), this field is invalid and doesn't need to be specified.
ApplyJoinOption	String	Optional	Method for handling requests to join the group. Valid values: <code>FreeAccess</code> , <code>NeedPermission</code> , and <code>DisableApply</code> . Community groups currently do not support this field.
AppDefinedData	Array	Optional	By default, this field is not available. To learn how to enable group-level custom fields, see <a href="#">Custom Fields</a> .
MuteAllMember	String	Optional	Mutes all members in a group and allows only the group admin, group owner, and system admin to speak.
InviteJoinOption	String	Optional	Method for handling invitations to join the group. Valid values: <code>FreeAccess</code> (directly inviting users to the group without approval and other operations), <code>NeedPermission</code> (requiring approval from

			the group admin or owner), and DisableInvite (not supporting SDK invitations to the group). This option is not supported by the AVChatRoom group type.
From_Account	String	Optional	Operator source account, optional. If this field is not filled in, the operator who modifies the group information is the App admin account used when calling this interface. In addition, the App can also "forge" the operator who modifies the group information through this field to implement some special functional requirements. If you specify this field, you must ensure that the account in the field exists.
EnablePermission	Integer	Optional	Only communities that support topics can set this field. Whether to enable the permission group as the basis for permission judgement: 0: Disable 1: Enable
DefaultPermission	Integer	Optional	Only groups that enable the permission group as the basis for permission judgement can set this field. The default permissions of a group member in the group.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10026	The command word of the SDKAppID request is disabled. Please <a href="#">submit a ticket</a> .
10007	Insufficient operation permissions. Check whether the operator is the app admin or whether the operator has the permission to modify the fields in the request.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Deleting Group Members ([v4/group\\_open\\_http\\_svc/delete\\_group\\_member](#))



# Importing a Group Profile

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API allows the app admin to import group data without triggering callbacks or delivering notifications. When your app needs to be migrated to Chat from another instant messaging system, you can use this API to import existing group data.

## API Calling Description

### Applicable group types

Group Type	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

Above are the built-in group types of the Chat service. For more information, see [Group System](#).

### Note

AVChatRoom groups do not support importing basic group data. If you attempt to import basic group data for AVChatRoom groups, error 10007 is returned. To achieve the effect of importing basic group data, you can [create a group](#) and [modify basic group data](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/import_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/import_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Import a group. You can use `CreateTime` to specify the group creation time.

```
{
  "Owner_Account": "leckie", // User ID of the group owner (optional)
  "Type": "Public", // Group type: Private, Public, ChatRoom, or Community (required)
  "Name": "TestGroup", // Group name (required)
  "CreateTime": 1448357837 // Group creation time (optional). If this field is not
}
```

### Containing only the topic-enabled parameter format

Import a group and specify its type as a topic-enabled community group.

```
{
  "Type": "Community", // Group type (required)
  "Name": "test_import_group", // Group name (required)
  "SupportTopic": 1// Whether the topic option is supported. Valid values: `1`: yes
}
```

### Specifying other optional fields

You can specify optional fields such as `Introduction` and `Notice` . The request format is the same as that of a group creation request.

```
{
  "Owner_Account": "leckie", // User ID of the group owner (optional)
  "Type": "Public", // Group type: Private, Public, ChatRoom, or Community (required)
  "GroupId": "MyFirstGroup", // User-defined group ID for external display (optional)
  "Name": "TestGroup", // Group name (required)
  "Introduction": "This is group Introduction", // Group introduction (optional)
  "Notification": "This is group Notification", // Group notice (optional)
  "FaceUrl": "http://this.is.face.url",
  "MaxMemberCount": 500, // Maximum number of group members (optional)
  "ApplyJoinOption": "FreeAccess", // Method for handling requests to join the group
  "CreateTime": 1448357837, // Group creation time (optional). If this field is not specified, the system will automatically generate a creation time.
  "AppDefinedData": [ // Group custom field (optional)
    {
      "Key": "GroupTestData1", // Key of the app custom field
      "Value": "xxxxx" // Value of the custom field
    },
    {
      "Key": "GroupTestData2",
      "Value": "abc\\u0000\\u0001" // The custom field supports binary data.
    }
  ]
}
```

### Request fields

Field	Type	Required	Description
Owner_Account	String	Optional	Group owner ID, which will be automatically added to group members. If this field is not specified, the group will have no group owner.
Type	String	Required	Group type, which can be Public, Private (Work), ChatRoom, or Community.
GroupId	String	Optional	To simplify group IDs and make them easy to remember,

			Tencent Cloud allows apps to customize group IDs during group creation through RESTful APIs. For details, see <a href="#">Group Systems</a> .
Name	String	Required	Group name, whose maximum length is 30 bytes.
Introduction	String	Optional	Group introduction, whose maximum length is 240 bytes.
Notification	String	Optional	Group notice, whose maximum length is 300 bytes.
FaceUrl	String	Optional	URL of the group profile photo, whose maximum length is 100 bytes.
MaxMemberCount	Integer	Optional	Maximum number of group members, which is 6,000 at the maximum. The default value is 2,000. Currently, a community group supports up to 100,000 group members.
ApplyJoinOption	String	Optional	Method for handling requests to join the group. Valid values: <code>FreeAccess</code> , <code>NeedPermission</code> (default), and <code>DisableApply</code> . This field is not supported by community groups.
AppDefinedData	Array	Optional	Group custom field. By default, this field is unavailable and needs to be enabled before use. For details, see <a href="#">Group Systems</a> .
CreateTime	Integer	Optional	Group creation time
SupportTopic	Integer	No	Whether the topic option is supported. This field is valid only for communities.

## Sample response

### Basic format

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "@TGS#2J4SZEAE"
}
```

### Containing only the topic-enabled parameter format

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "@TGS#_@TGS#c4YHCIIM62CX"
```

```
}
```

Specifying other optional fields

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupId": "MyFirstGroup"
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
GroupId	String	Group ID after successful creation, which is assigned by the Chat backend or specified by users.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. This error occurs when, for example, a member in a public group tries to remove other users from the group (only the app admin can perform this operation).
10021	The group ID is already in use. Specify another group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Setting the unread message count of a member ([v4/group\\_open\\_http\\_svc/set\\_unread\\_msg\\_num](#))

Importing group members ([v4/group\\_open\\_http\\_svc/import\\_group\\_member](#))

Disbanding a group ([v4/group\\_open\\_http\\_svc/destroy\\_group](#))

# Group Member Management

## Adding Group Members

Last updated : 2024-07-16 11:15:59

### Feature Overview

This API allows the app administrator to add new members to a group.

### API Calling Description

#### Applicable group types

Group Type	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

Above are the built-in group types of Tencent Cloud Chat. For more information, see [Group System](#).

#### Note

Audio-video groups (AVChatRoom) do not support this API. If you use this API on an audio-video group, error 10007 will be returned. The only way for users to join this type of group is to apply to join.

#### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/add_group_member?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/add_group_member	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Used to invite users to a group. A single request supports adding up to 300 members. By default, the backend delivers group system notifications to all group members, except for private groups (or work groups in the new version) that have not been activated.

```
{
  "GroupId": "@TGS#2J4SZEAE", // (Required) Target group
  "MemberList": [ // Up to 300 members can be added at a time.
    {
      "Member_Account": "tommy" // The ID of the member to be added (required)
    },
    {
      "Member_Account": "jared"
    }
  ]
}
```



```
    }  
  }  
}
```

### Adding members silently

When `Silence` is set to `1`, the system does not notify anyone after successfully adding members.

```
{  
  "GroupId": "@TGS#2J4SZEAE", // (Required) Target group  
  "Silence": 1, // Whether to add members silently (optional)  
  "MemberList": [ // Up to 300 members can be added at a time.  
    {  
      "Member_Account": "tommy" // The ID of the member to be added (required)  
    },  
    {  
      "Member_Account": "jared"  
    }  
  ]  
}
```

### Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the target group
Silence	Integer	Yes	Whether to add members silently. <code>0</code> : No. <code>1</code> : Yes. The default value is <code>0</code> .
MemberList	Array	Yes	A list of the members to be added
Member_Account	String	Yes	The UserID of the member to be added

### Sample response

```
{  
  "ActionStatus": "OK",  
  "ErrorInfo": "",  
  "ErrorCode": 0,  
  "MemberList": [  
    {  
      "Member_Account": "tommy",  
      "Result": 1 // The result of adding the member. 0: failed. 1:  
successful. 2: already in the group.  
    },  
    {  
      "Member_Account": "jared",  

```

```
      "Result": 1
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MemberList	Array	The result of adding members
Member_Account	String	The UserID of the member
Result	Integer	The result of adding the member. <code>0</code> : Failed. <code>1</code> : Successful. <code>2</code> : Already in the group. <code>3</code> : Pending approval by the invitee or admin.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10005	The number of member accounts in the request packet exceeds 300.
10007	No operation permissions. In this case, check whether the group type supports user invitation. For example, AVChatRoom and BChatRoom groups do not allow anyone to invite others to the groups.
10014	The users in the request cannot be added to the group because the group is already full. In this case, try deleting some <code>Member_Account</code> in the request or change the value of the <code>MaxMemberNum</code>

	field in the basic group information. For information, see the <b>Basic group information</b> section in <a href="#">Group basic information</a> .
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use a correct group ID.
10016	The developer backend has rejected this operation through a webhook.
10019	The UserID does not exist. Make sure all the accounts specified in <code>Member_Account</code> are correct.
10026	The command word of the <code>SDKAppID</code> request is disabled. Contact customer service.
10037	The number of groups that the invited user has joined exceeds the limit. In this case, check and delete the <code>Member_Account</code> that has joined excessive groups, or purchase an upgrade based on the your need. For information about Chat plans, see <a href="#">Pricing</a> .

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Deleting Group Members ([v4/group\\_open\\_http\\_svc/delete\\_group\\_member](#))

## Possible Webhooks

[Before Inviting a User to a Group](#)

[After a User Joins a Group](#)

[After a Group Is Full](#)

# Deleting Group Members

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by app admins to delete group members.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

Audio-video groups (AVChatRoom) do not support deleting group members. If you attempt to delete members of an audio-video group, error 10004 will be returned. To achieve the effect of deleting group members, admins can [batch mute and unmute group members](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/delete_group_member?
sdkappid=88888888&identifier=admin&usersig=xxx&random=999999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/delete_group_member	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Used to delete group members. A single request supports deleting up to 100 group members. By default, the backend sends system notifications on the group member deletion to all group members, except for Private groups (same as Work groups in the new version) that have not been activated.

Even if a user to be deleted is not in the group, the API will still return success.

```
{
  "GroupId": "@TGS#2J4SZEAE", // Target group (required)
  "MemberToDel_Account": [ // List of group members to be deleted. A maximum of 100
    "tommy",
    "jared"
  ]
}
```

## Silent deletion

When `Silence` is set to `1`, the system will not notify anyone after successfully deleting a member.

```
{
  "GroupId": "@TGS#2J4SZEAE", // Target group (required)
  "Silence": 1, // Whether to delete members silently (optional)
  "MemberToDel_Account": [ // List of group members to be deleted. A maximum of 100
    "tommy",
    "jared"
  ]
}
```

## Specifying the reason for deletion

Specify the reason for deleting a user from a group, which will be reflected in the delivered notification.

```
{
  "GroupId": "@TGS#2J4SZEAE", // Target group (required)
  "Reason": "kick reason", // Reason for deleting a user from a group (optional)
  "MemberToDel_Account": [ // List of group members to be deleted. A maximum of 100
    "tommy",
    "jared"
  ]
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the target group
Silence	Integer	No	Whether to adopt silent deletion. <code>0</code> : No (default value); <code>1</code> : Yes. Silent deletion means that no members in the group, except the deleted member, will be notified when a group member is deleted.
Reason	String	No	Reason for deleting a user from a group
MemberToDel_Account	Array	Yes	Group members to be deleted

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

```
}

```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

References

---

Adding Group Members ([v4/group\\_open\\_http\\_svc/add\\_group\\_member](#))

## Possible Webhooks

[After a User Leaves a Group](#)



# Banning Group Members

Last updated : 2025-01-14 14:36:49

## Feature Overview

This API is used to ban a member of an audio-video group. A banned member cannot receive messages nor rejoin the audio-video group during the ban duration.

### Note :

Audio-video group member banning is supported by SDK Enhanced edition v6.6 or later and web SDK v2.22.0 or later. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#), go to the [console](#), choose **Group configuration** > **Group feature configuration**, and enable **Audio-video group member banning**.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	No. Same as work groups (Work) in the new version.
Public	No
ChatRoom	No. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/ban_group_member?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/ban_group_member	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Banning members of an audio-vid eo group

```
{
  "GroupId": "@TGS#aJRGc4MH6", // ID of the group whose members are to be
banned
  "Members_Account":[brennanli3", "brennanli12], // Account IDs of the
members to be banned. Up to 20 account IDs are supported per request.
  "Duration":3600, // Ban duration, in seconds
  "Description": "you are banned because of irregularities" // Ban
information, which can be up to 1,000 bytes in length
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group whose members are to be banned
Members_Account	Array	Yes	Account IDs of the members to be banned. Up to 20 account IDs are supported per request.
Duration	Integer	Yes	Ban duration, in seconds. It is a 32-bit unsigned integer.
Description	String	No	Ban information, which can be up to 1,000 bytes in length

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.

10004	Invalid parameter. Check the error description and troubleshoot the issue.
10005	The number of members to be banned exceeds the upper limit of 20 members per request.
10007	Insufficient operation permissions. For example, the switch is not enabled in the console, or the operating account is not the root account.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Unbanning Group Members

Last updated : 2025-01-14 14:36:49

## Feature Overview

This API is used to unban a member of an audio-video group. An unbanned member can rejoin the audio-video group to receive messages.

### Note:

Audio-video group member unbanning is supported by SDK Enhanced edition v6.6 or later and web SDK v2.22.0 or later. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#), go to the [console](#), choose **Feature configuration > Group configuration > \*\* Group feature configuration\*\***, and disable **Audio-video group member banning**.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	No. Same as work groups (Work) in the new version.
Public	No
ChatRoom	No. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/unban_group_member?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/unban_group_member	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Unbanning members of an audio-video group

```
{
  "GroupId": "@TGS#aJRGc4MH6", // ID of the group whose members are to be
  unbanned
  "Members_Account": [brennanli3, "brennanli12"], // Account IDs of the
  members to be unbanned. Up to 20 account IDs are supported per request.
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group whose members are to be unbanned
Members_Account	Array	Yes	Account IDs of the members to be unbanned. Up to 20 account IDs are supported per request.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10005	The number of members to be unbanned exceeds the maximum limit of 20 members per request.

10007	Insufficient operation permissions. For example, the switch is not enabled in the console, or the operating account is not the root account.
-------	--

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.



# Bulk Muting and Unmuting

Last updated : 2024-07-16 11:15:59

## Feature Overview

The app administrator can mute certain group members for a specific period of time.

The app administrator can unmute certain group members.

When muted users leave and then enter the group again, they remain muted.

## API Calling Description

### Applicable group types

Group Type	Support for This RESTful API
Private	No. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

Above are the built-in group types of the Chat service. For more information, see [Group System](#).

### Note

Private groups (i.e., work groups in the new version) do not support muting members.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/forbid_send_msg?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/forbid_send_msg	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Muting members

You can set a specific period of time in `MuteTime` to mute specified members.

```
{
  "GroupId": "@TGS#2C5SZEAEF",
  "Members_Account": [ // Up to 500 member accounts
    "peter",
    "leckie"
  ],
  "MuteTime": 60 // Muting period, in seconds
}
```

### Unmuting members

To unmute members, set `MuteTime` to `0`.

```
{
  "GroupId": "@TGS#2C5SZEAEF",
```

```
"Members_Account": [ // Up to 500 member accounts
    "peter",
    "leckie"
],
"MuteTime": 0 // 0 indicates to unmute members.
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the group to be queried
Members_Account	Array	Yes	The member accounts to be muted. A maximum of 500 accounts are supported.
MuteTime	Integer	Yes	Muting period of the unsigned integer type, in seconds. 0 : Unmute; 4294967295 : Permanent muting
TopicId	String	Optional	ID of the topic whose muting status is to be set. This field applies only to topic-enabled community groups.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. OK : successful; FAIL : failed
ErrorCode	Integer	Error code. 0 : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Getting the List of Muted Group Members ([v4/group\\_open\\_http\\_svc/get\\_group\\_muted\\_account](#))

# Getting the List of Muted Group Members

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to get the list of muted users in a group based on the group ID.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_muted_account?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_group_muted_account</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Getting the list of muted members in a group based on the group ID

```
{
  "GroupId": "@TGS#1KGZ2RAEU"
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Required	ID of the group for which the list of muted members is to be obtained
TopicId	String	Optional	ID of the topic for which the list of muted members is to be obtained. This field applies only to topic-enabled communities.

## Sample response

```
{
  "ActionStatus": "OK",
}
```

```
"ErrorCode": 0,
"GroupId": "@TGS#2FZNNRAEU",
"MutedAccountList": [ // List of muted users in the group
  {
    "Member_Account": "tommy", // User ID
    "MutedUntil": 1458115189 // Muting stop time (UTC - Coordinated
Universal Time)
  },
  {
    "Member_Account": "peter",
    "MutedUntil": 1458115189
  }
]
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MutedAccountList	Array	Array of muted users' information, including user IDs and muting stop time (UTC - Coordinated Universal Time)

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other

	users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Bulk Muting and Unmuting ([v4/group\\_open\\_http\\_svc/forbid\\_send\\_msg](#))



# Changing Group Owner

Last updated : 2024-07-16 11:15:59

## Feature Overview

The app admin can call this API to transfer the group ownership to others.

For a group without an owner, the app admin can call this API to specify a user as the owner of the group.

The new group owner must be a member of the group.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No (See the note below)
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

**Note**

Audio-video groups (AVChatRoom) do not support the transfer of group ownership. If you call this API for such a group, error code 10007 will be returned.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/change_group_owner?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/change_group_owner	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Transfer the group ownership to another group member. The new group owner must be a member of the group.

```
{
  "GroupId": "@TGS#1NVTZEAE4", // (Required) ID of the group whose ownership
is to be transferred
  "NewOwner_Account": "peter" // (Required) ID of the new group owner
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group whose ownership is to be transferred

NewOwner_Account	String	Yes	ID of the new group owner
------------------	--------	-----	---------------------------

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	A system error occurred. Try again or contact technical support.
10004	A parameter is invalid. Check the <code>ErrorInfo</code> field in the response packet to see whether the required fields have been specified or whether the fields are set according to protocol requirements.
10007	Insufficient operation permissions. Check whether the operator is the app admin.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly.

# API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Recalling group messages ([v4/group\\_open\\_http\\_svc/group\\_msg\\_recall](#))

# Querying the Roles of Users in a Group

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to batch obtain the roles of users in a group.

## API Calling Description

### Applicable group types

Group Type ID	Support for This RESTful API
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

Above are the preset group types of Chat. For more information, see [Group System](#).

### Note

Audio-video groups (AVChatRoom) do not support this API. If this API is called for an audio-video group, a 10007 error will be returned. For an audio-video group, you can call the API for [Getting Group Member Profiles](#) to query member roles.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_role_in_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/get_role_in_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Obtain the list of groups a user has joined. The group information includes the group ID and the IDs of group members whose roles are to be queried.

```
{
  "GroupId": "@TGS#2C5SZEAEF",
  "User_Account": [ // Up to 500 member accounts are supported
    "leckie",
    "peter",
    "wesley"
  ]
}
```

## Request fields

Field	Type	Required	Description
-------	------	----------	-------------

GroupId	String	Yes	The ID of the group to be queried
User_Account	Array	Yes	User accounts to be queried. Up to 500 accounts are supported.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "UserIdList": [ // Result
    {
      "Member_Account": "leckie",
      "Role": "Owner" // Member role: Owner/Admin/Member/NotMember
    },
    {
      "Member_Account": "peter",
      "Role": "Member"
    },
    {
      "Member_Account": "wesley",
      "Role": "NotMember"
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
UserIdList	Array	Member roles that are pulled, including Owner, Admin, Member, and NotMember

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. The operator must have permissions to perform corresponding operations.
10010	The group does not exist or has been deleted.
10015	Failed to parse the JSON packet. Check whether the packet complies with JSON specifications.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Getting the Groups a User Has Joined ([v4/group\\_open\\_http\\_svc/get\\_joined\\_group\\_list](#))



# Importing Group Members

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used to import group members without triggering callbacks or delivering notifications. When your app needs to be migrated to Chat from another instant messaging system, you can use this API to import existing group member data.

## API Calling Description

### Applicable group types

Group Type	Applicable
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as meeting groups (Meeting) in the new version.
AVChatRoom	No
Community	Yes

Above are the built-in group types of the Chat service. For more information, see [Group System](#).

### Note

Generally, importing group members is not required for audio-video groups (AVChatRoom), so audio-video groups do not support this API. If you attempt to use this API on an audio-video group, error 10007 will be returned.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/import_group_member?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/import_group_member	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

You can call this API to initiate a request to import members to a group. A single request can import up to 300 members. If the unread message count specified by this API is greater than the current number of messages of the group, the unread message count is set to the current number of messages of the group.

### Caution

Before importing a member, ensure that the time that the member joined the group is later than the group creation time and earlier than the current time. Otherwise, the member cannot be imported.

```
{
  "GroupId": "@TGS#2J4SZEAE", // (Required) Target group
  "MemberList": [ // Up to 300 members can be added at a time.
    {
      "Member_Account": "tommy", // (Required) ID of the member to be added
```

```
        "Role": "Admin", // (Optional) Role of the group member to be imported.
        // Currently, the only supported role is Admin.
        "JoinTime": 1448357837, // (Optional) Time when the group member to be
        // imported joined the group
        "UnreadMsgNum": 5 // (Optional) Unread message count of the
        // member
    },
    {
        "Member_Account": "jared",
        "JoinTime": 1448357857,
        "UnreadMsgNum": 2
    }
]
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the target group
MemberList	Array	Yes	A list of the members to be added
Member_Account	String	Yes	Group member accounts to be imported
Role	String	No	Role of the group member to be imported. Currently, only <code>Admin</code> is supported. If this field is not specified, the value <code>Member</code> (ordinary user) is used.
JoinTime	Integer	No	Time when the group member to be imported joined the group
UnreadMsgNum	Integer	No	Unread message count of the group member to be imported

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "MemberList": [
    {
      "Member_Account": "tommy",
      "Result": 1 // Result of the import. 0: Failed; 1: Succeeded; 2:
      // Already a group member
    },
    {

```

```
        "Member_Account": "jared",
        "Result": 1
    }
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MemberList	Array	The result of adding members
Member_Account	String	Account of the group member
Result	Integer	Result of the import. 0: Failed; 1: Succeeded; 2: Already a group member

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10005	The request packet carries too many accounts.
10006	Reached the operation frequency limit. Please reduce the call rate.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been disbanded.

10015

Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Deleting Group Members ([v4/group\\_open\\_http\\_svc/delete\\_group\\_member](#))

# Group Member Information

## Getting Group Member Profiles

Last updated : 2025-03-10 10:30:48

### Overview

This API is used by the app admin to obtain group member profiles based on the group ID.

### API Calling Description

#### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as the work group (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No
Community	Yes. Getting group member profiles by using the <code>Next</code> field.

These are the preset group types in Chat. For more information, see [Group System](#).

#### Note

Due to the large number of members in a community group, the pagination getting method is changed to the batch getting method based on the `Next` field.

#### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_member_info?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table only describes the modified parameters when this API is called. For more information on other parameters, please see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_group_member_info	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

A basic request is used to obtain detailed group member information, including group member profiles and custom group member fields. The request requires only the group ID.

```
{
  "GroupId": "@TGS#1NVTZEAE4"  // Group ID (required)
}
```

### Pagination

You can use the `Limit` and `Offset` fields to control the pagination mode:

`Limit`: specifies the maximum number of members in the `MemberList` array in the response. Maximum value: 200; recommended value: 100.

**Offset** : specifies from which group member to start pulling information. If the page number starts from **1** , the value of **Offset** for each page should be  $(\text{page number} - 1) \times \text{number of group members to display on each page}$  .

For example, to display 20 group members on each page, the request parameters for the first page should be

`{"Limit": 20, "Offset": 0}` , the request parameters for the second page should be `{"Limit": 20, "Offset": 20}` , and so on.

```
{
  "GroupId": "@TGS#1NVTZEAE4", // Group ID (required)
  "Limit": 100, // Maximum number of members to pull information
  "Offset": 0 // Sequence number of the member from whom to start pulling
information
}
```

### Note

The community currently does not support paging to get the group member details.

### Batch

You can use the **Limit** and **Next** fields to control the batch mode:

**Limit** : specifies the maximum number of members in the **MemberList** array in the response. Maximum value: 100

**Next** : specifies a member position from which subsequent information is to be pulled. For the first request, the client request parameter **Next** must pass in "". For the last request, the server returns "" for the **Next** parameter, indicating that the information pulling ends.

For intermediate requests, the client request parameter **Next** uses the last value of **Next** returned by the server. This is similar to Redis scan cursor queries.

For example, if you are to perform batch pulling, the parameters in the first request should be `{"Limit" : 20, "Next" : ""}` , and the server returns the following:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "144115265295492787",
  "MemberList": [
    ....
  ]
}
```

The parameters in the second request should be `{"Limit" : 20, "Next" : "144115265295492787"}` .

And so on, until the server replies with **Next** as "", indicating that there is no subsequent member information, and the client should end the query.

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group ID (required)
```



```
"Limit": 100, // Maximum number of members to pull information
"Next": "" // Start pulling from the position where the last pulling ends
}
```

### Note

Only the community supports obtaining group member details in batches.

### Specifying information to pull

You can use the `MemberInfoFilter` filter field to specify fields to pull. Fields that are not specified in it will not be pulled.

```
{
  "GroupId": "@TGS#1NVTZEAE4", // Group ID (required)
  "MemberInfoFilter": [ // Information to pull, where `Member_Account` is
    included by default. If this field is not specified, all group member
    information will be pulled.
    "Role",
    "JoinTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "MuteUntil",
    "NameCard"
  ]
}
```

### Pulling the information of members in the specified role

You can use the `MemberRoleFilter` filter field to specify the role of members to pull information. If this field is not specified, the information of members in all roles will be pulled.

```
{
  "GroupId": "@TGS#37AB3PAEC", // Group ID (required)
  "MemberRoleFilter": [ // Member role filter
    "Owner",
    "Member"
  ]
}
```

### Pulling custom group member fields

You can use the `AppDefinedDataFilter_GroupMember` filter field to specify the custom group member fields to pull. Fields that are not specified in it will not be pulled.

```
{
  "GroupId": "@TGS#37AB3PAEC", // Group ID (required)
  "AppDefinedDataFilter_GroupMember": [ // Filter for custom group member
    fields
  ]
}
```

```
    "MemberDefined2" // Key of a custom group member field
  ]
}
```

Response to an ALL IN ONE request

```
{
  "GroupId": "@TGS#1NVTZEAE4", // Group ID (required)
  "MemberInfoFilter": [ // Information to pull. If this field is not specified,
all group member information will be pulled.
    "Role",
    "JoinTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "MuteUntil",
    "NameCard"
  ],
  "MemberRoleFilter": [ // Member role filter
    "Owner",
    "Member"
  ],
  "AppDefinedDataFilter_GroupMember": [ // Filter for custom group member fields
    "MemberDefined2", // Key of a custom group member field
    "MemberDefined1"
  ],
  "Limit": 100, // Maximum number of members to pull information
  "Offset": 0 // Sequence number of the member from whom to start pulling
information
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group to pull member information
MemberInfoFilter	Array	No	Information to pull. If this field is not specified, all group member information will be pulled. For details on group member information fields, see <a href="#">Group member profile</a> .
MemberRoleFilter	Array	No	Role of group members to pull information. If this field is not specified, the information of members in all roles will be pulled. The

			member role can be <code>Owner</code> , <code>Admin</code> , or <code>Member</code> .
AppDefinedDataFilter_GroupMember	Array	No	This field is omitted by default. It specifies the custom group member fields to pull. For more information, see the <b>Custom Fields</b> section in <a href="#">Group System</a> .
Limit	Integer	No	Maximum number of members to pull information at a time. The value cannot exceed 6000. If this field is not specified, the information of all members in the group will be obtained.
Offset	Integer	No	Sequence number of the member from whom to start pulling information. If this field is set to <code>0</code> , the information is pulled starting from the first member.
Next	String	No	Position of the last member pulled. This field is required for a community group. A community group does not support the <code>Offset</code> field. It uses the <code>Next</code> field instead. For the first call, "" must be passed in for <code>Next</code> . For subsequent calls, the last value of <code>Next</code> must be passed in.

Sample response

Response to a basic or pagination request

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "144115265295492787", // This field is returned only for a community
group
  "MemberNum": 2, // Total number of members in the group
  "MemberList": [ // Group member list
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // Time when the member joined the group
      "MsgSeq": 1233,
      "MsgFlag": "AcceptAndNotify",
    }
  ]
}
```

```

        "LastSendMsgTime": 1425976500, // Last time when the member sent a
message
        "MuteUntil": 1431069882, // Muting end time in seconds
        "AppMemberDefinedData": [ // Custom group member fields
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    },
    {
        "Member_Account": "peter",
        "Role": "Member ",
        "JoinTime": 1425976500,
        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500,
        "MuteUntil": 0, // `0`: the member is not muted; other values: the
time when the member will be unmuted
        "AppMemberDefinedData": [ // Custom group member fields
            {
                "Key": "MemberDefined1",
                "Value": "ModifyDefined1"
            },
            {
                "Key": "MemberDefined2",
                "Value": "ModifyDefined2"
            }
        ]
    }
]
}

```

### Response to a request pulling specified fields

```

{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "MemberNum": 2, // Total number of members in the group
    "MemberList": [ // Group member list
        {
            "Member_Account": "bob",

```

```

        "Role": "Owner",
        "JoinTime": 1425976500, // Time when the member joined the group
        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500, // Last time when the member sent a
message
        "MuteUntil": 1431069882, // Muting end time in seconds
    },
    {
        "Member_Account": "peter",
        "Role": "Member ",
        "JoinTime": 1425976500,
        "MsgSeq": 1233,
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 1425976500,
        "MuteUntil": 0, // `0`: the member is not muted; other values: the
time when the member will be unmuted
    }
]
}

```

### Pulling the information of members in the specified role

```

{
    "ActionStatus": "OK", // The request succeeded.
    "ErrorCode": 0, // Return code
    "MemberList": [
        {
            "JoinTime": 1450680436, // Time when the member joined the group
            "LastSendMsgTime": 0, // Last time when the member sent a message
            "Member_Account": "Test_1", // Member account
            "MsgFlag": "AcceptNotNotify", // Type of member messages being
blocked
            "MsgSeq": 1, // Sequence number of the member's read message
            "NameCard": "", // Member's contact card
            "Role": "Owner", // Member's role
            "MuteUntil": 0 // `0`: the member is not muted; other values: the
time when the member will be unmuted
        },
        {
            "JoinTime": 1450680436,
            "LastSendMsgTime": 0,
            "Member_Account": "Test_6",
            "MsgFlag": "AcceptNotNotify",
            "MsgSeq": 1,
            "NameCard": "",
            "Role": "Admin",

```

```

        "MuteUntil": 0
    }
],
"MemberNum": 8 // Total number of members in the group
}

```

### Pulling custom group member fields

```

{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "MemberNum": 2, // Total number of members in the group
    "MemberList": [ // Group member list
        {
            "Member_Account": "bob",
            "Role": "Owner",
            "JoinTime": 1425976500, // Time when the member joined the group
            "MsgSeq": 1233,
            "MsgFlag": "AcceptAndNotify",
            "LastSendMsgTime": 1425976500, // Last time when the member sent a
message
            "MuteUntil": 1431069882, // Muting end time in seconds
            "AppMemberDefinedData": [ // Custom group member fields
                {
                    "Key": "MemberDefined2",
                    "Value": "ModifyDefined2"
                }
            ]
        },
        {
            "Member_Account": "peter",
            "Role": "Member",
            "JoinTime": 1425976500,
            "MsgSeq": 1233,
            "MsgFlag": "AcceptAndNotify",
            "LastSendMsgTime": 1425976500,
            "MuteUntil": 0, // `0`: the member is not muted; other values: the
time when the member will be unmuted
            "AppMemberDefinedData": [ // Custom group member fields
                {
                    "Key": "MemberDefined2",
                    "Value": "ModifyDefined2"
                }
            ]
        }
    ]
}

```

```
}

```

## Response to an ALL IN ONE request

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "144115265295492787", // This field is returned only for a community
group
  "MemberNum": 2, // Total number of members in the group
  "MemberList": [ // Group member list
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // Time when the member joined the group
      "MsgSeq": 1233,
      "MsgFlag": "AcceptAndNotify",
      "LastSendMsgTime": 1425976500, // Last time when the member sent a
message
      "MuteUntil": 1431069882, // Muting end time in seconds
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "MemberDefined1",
          "Value": "ModifyDefined1"
        },
        {
          "Key": "MemberDefined2",
          "Value": "ModifyDefined2"
        }
      ]
    },
    {
      "Member_Account": "peter",
      "Role": "Member",
      "JoinTime": 1425976500,
      "MsgSeq": 1233,
      "MsgFlag": "AcceptAndNotify",
      "LastSendMsgTime": 1425976500,
      "MuteUntil": 0, // `0`: the member is not muted; other values: the
time when the member will be unmuted
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "MemberDefined1",
          "Value": "ModifyDefined1"
        },
        {

```

```
        "Key": "MemberDefined2",
        "Value": "ModifyDefined2"
    }
  ]
}
]
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed
ErrorInfo	String	Error information
MemberNum	Integer	Total number of members in the group
MemberList	Array	Returned group member list, which contains information of all or specified group members. For details on group member information fields, see <a href="#">Group member profile</a> .
AppMemberDefinedData	Array	Returned custom group member fields
Next	String	Value of <code>Next</code> to be passed in in the next request. This field is returned only for a community group.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.



10007	No operation permissions. Check whether the operator is an app admin or whether the operator has the permission to read the fields in the request.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.
10018	The response exceeds the maximum size allowed (1 MB) because the group member data volume is too large. Try to use <code>Limit</code> and <code>Offset</code> to pull the group member data by page.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Modifying the Profile of a Group Member ([v4/group\\_open\\_http\\_svc/modify\\_group\\_member\\_info](#))

# Getting Specified Group Member Profiles

Last updated : 2025-03-10 10:30:48

## Overview

This API is used by the app admin to obtain specified group member profiles based on the group ID and UserId list of specified group members.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as the work group (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_specified_group_member_info?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table only describes the modified parameters when this API is called. For more information on other parameters, please see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located.

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_specified_group_member_info</code>	Request API.
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

A basic request is used to obtain detailed group member information, including group member profiles and custom group member fields. The request requires the group ID and `UserId` list of specified group members. The number of specified group members cannot exceed 50 at a time.

```
{
  "GroupId": "@TGS#2KIFZCIPQ", // Group ID (required)
  "Member_List_Account" : ["bob","peter"] // UserId list of specified group member (required)
}
```

### Specifying information to pull

You can use the `MemberInfoFilter` filter field to specify fields to pull. Fields that are not specified in it will not be pulled.

```
{
  "GroupId": "@TGS#2KIFZCIPQ", // Group ID (required)
  "Member_List_Account" : ["bob","peter"], // UserId list of specified group
member (required)
  "MemberInfoFilter": [ // Information to pull, where `Member_Account` is
included by default. If this field is not specified, all group member
information will be pulled.
    "Role",
    "JoinTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "MuteUntil",
    "NameCard",
    "OnlineStatus"
  ]
}
```

### Pulling the information of members in the specified role

You can use the `MemberRoleFilter` filter field to specify the role of members to pull information. If this field is not specified, the information of members in all roles will be pulled.

```
{
  "GroupId": "@TGS#2KIFZCIPQ", // Group ID (required)
  "Member_List_Account" : ["bob","peter","John"], // UserId list of specified
group member (required)
  "MemberRoleFilter": [ // Member role filter
    "Owner",
    "Member"
  ]
}
```

### Pulling custom group member fields

You can use the `AppDefinedDataFilter_GroupMember` filter field to specify the custom group member fields to pull. Fields that are not specified in it will not be pulled.

```
{
  "GroupId": "@TGS#2KIFZCIPQ", // Group ID (required)
  "Member_List_Account" : ["bob","peter"], // UserId list of specified group
member (required)
  "AppDefinedDataFilter_GroupMember": [ // Filter for custom group member
fields
    "group_member_p" // Key of a custom group member field
  ]
}
```

## Response to an ALL IN ONE request

```
{
  "GroupId": "@TGS#2KIFZCIPQ", // Group ID (required)
  "Member_List_Account" : ["bob","peter"], // UserId list of specified group
member (required)
  "MemberInfoFilter": [ // Information to pull. If this field is not specified,
all group member information will be pulled.
    "Role",
    "JoinTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "MuteUntil",
    "NameCard"
  ],
  "MemberRoleFilter":[ // Member role filter
    "Owner",
    "Member"
  ],
  "AppDefinedDataFilter_GroupMember": [ // Filter for custom group member fields
    "group_member_p", // Key of a custom group member field
    "group_member_p2"
  ]
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group to pull member information
Member_List_Account	Array	Yes	UserID list of specified group members that needs to obtain information. The number of UserID cannot exceed 50.
MemberInfoFilter	Array	No	Information to pull. If this field is not specified, all group member information will be pulled. For details on group member information fields, see <a href="#">Group member profile</a> .
MemberRoleFilter	Array	No	Role of group members to pull information. If this field is not specified, the information of members in all roles will be pulled. The member role can be <code>Owner</code> , <code>Admin</code> , or <code>Member</code> .

AppDefinedDataFilter_GroupMember	Array	No	This field is omitted by default. It specifies the custom group member fields to pull. For more information, see the <b>Custom Fields</b> section in <a href="#">Group System</a> .
----------------------------------	-------	----	---

Sample response

Response to a basic

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "GroupId": "@TGS#2KIFZCIPQ",
  "MemberList": [ //Group member list
    {
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        },
        {
          "Key": "group_member_p2",
          "Value": "the value2"
        }
      ],
      "JoinTime": 1728964923,          // Time when the member joined the
group
      "LastSendMsgTime": 1728973475,  // Last time when the member sent
a message
      "Member_Account": "bob",
      "MsgFlag": "AcceptAndNotify",
      "MsgSeq": 7,
      "MuteUntil": 1728977081,        // Muting end time in seconds
      "NameCard": "bob",
      "Role": "Member"
    },
    {
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        },
        {
          "Key": "group_member_p2",
          "Value": "the value2"
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "JoinTime": 1728964923,
  "LastSendMsgTime": 1728973184,
  "Member_Account": "peter",
  "MsgFlag": "AcceptAndNotify",
  "MsgSeq": 3,
  "MuteUntil": 0,    // `0`: the member is not muted; other values:
the time when the member will be unmuted
  "NameCard": "Peter",
  "Role": "Member"
}
]
}

```

### Response to a request pulling specified fields

```

{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "GroupId": "@TGS#2KIFZCIPQ",
  "MemberList": [ // Group member list
    {
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        },
        {
          "Key": "group_member_p2",
          "Value": "the value2"
        }
      ],
      "JoinTime": 1728964923, // Time when the member joined the group
      "LastSendMsgTime": 1728973475, // Last time when the member sent a
message
      "Member_Account": "bob",
      "MsgFlag": "AcceptAndNotify",
      "MsgSeq": 7,
      "MuteUntil": 1728977081, // Muting end time in seconds
      "NameCard": "bob",
      "OnlineStatus": "Online",
      "Role": "Member"
    },
    {
      "AppMemberDefinedData": [ // Custom group member fields

```

```

        {
            "Key": "group_member_p",
            "Value": "the value"
        },
        {
            "Key": "group_member_p2",
            "Value": "the value2"
        }
    ],
    "JoinTime": 1728964923,
    "LastSendMsgTime": 1728973184,
    "Member_Account": "peter",
    "MsgFlag": "AcceptAndNotify",
    "MsgSeq": 3,
    "MuteUntil": 0,    // `0`: the member is not muted; other values:
the time when the member will be unmuted
    "NameCard": "Peter",
    "OnlineStatus": "Offline",
    "Role": "Member"
}
]
}

```

### Pulling the information of members in the specified role

```

{
    "ActionStatus": "OK",    // The request succeeded.
    "ErrorCode": 0,        // Return code
    "ErrorInfo": "",
    "GroupId": "@TGS#2KIFZCIPQ",
    "MemberList": [
        {
            "AppMemberDefinedData": [    // Custom group member fields
                {
                    "Key": "group_member_p",
                    "Value": "the value"
                },
                {
                    "Key": "group_member_p2",
                    "Value": "the value2"
                }
            ],
            "JoinTime": 1728964923,    // Time when the member joined the group
            "LastSendMsgTime": 1728973184,    // Last time when the member sent
a message
            "Member_Account": "peter",    // Member account

```



```

        "MsgFlag": "AcceptAndNotify",    // Type of member messages being
blocked
        "MsgSeq": 7,    // Sequence number of the member's read message
        "MuteUntil": 0,    // `0`: the member is not muted; other values:
the time when the member will be unmuted
        "NameCard": "Peter",    // Member's contact card
        "Role": "Member"    // Member's role
    },
    {
        "AppMemberDefinedData": [    // Custom group member fields
            {
                "Key": "group_member_p",
                "Value": ""
            },
            {
                "Key": "group_member_p2",
                "Value": ""
            }
        ],
        "JoinTime": 1728964631,
        "LastSendMsgTime": 0,
        "Member_Account": "John",
        "MsgFlag": "AcceptAndNotify",
        "MsgSeq": 4,
        "MuteUntil": 0,
        "NameCard": "",
        "Role": "Owner"
    },
    {
        "AppMemberDefinedData": [    // Custom group member fields
            {
                "Key": "group_member_p",
                "Value": "the value"
            },
            {
                "Key": "group_member_p2",
                "Value": "the value2"
            }
        ],
        "JoinTime": 1728964923,
        "LastSendMsgTime": 1728973475,
        "Member_Account": "bob",
        "MsgFlag": "AcceptAndNotify",
        "MsgSeq": 7,
        "MuteUntil": 1728977081,    // Muting end time in seconds
        "NameCard": "bob",
        "Role": "Member"
    }

```

```

    }
  ]
}

```

### Pulling custom group member fields

```

{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "GroupId": "@TGS#2KIFZCIPQ",
  "MemberList": [    // Group member list
    {
      "AppMemberDefinedData": [    // Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        }
      ],
      "JoinTime": 1728964923,      // Time when the member joined the
group
      "LastSendMsgTime": 1728973475,    // Last time when the member sent
a message
      "Member_Account": "bob",
      "MsgFlag": "AcceptAndNotify",
      "MsgSeq": 7,
      "MuteUntil": 1728977081,    // Muting end time in seconds
      "NameCard": "bob",
      "Role": "Member"
    },
    {
      "AppMemberDefinedData": [// Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        }
      ],
      "JoinTime": 1728964923,
      "LastSendMsgTime": 1728973184,
      "Member_Account": "peter",
      "MsgFlag": "AcceptAndNotify",
      "MsgSeq": 7,
      "MuteUntil": 0,    // `0`: the member is not muted; other values:
the time when the member will be unmuted
      "NameCard": "Peter",
      "Role": "Member",
    }
  ]
}

```

```
]
}
```

### Response to an ALL IN ONE request

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "GroupId": "@TGS#2KIFZCIPQ",
  "MemberList": [ // Group member list
    {
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        },
        {
          "Key": "group_member_p2",
          "Value": "the value2"
        }
      ],
      "JoinTime": 1728964923, // Time when the member joined the
group
      "LastSendMsgTime": 1728973184, // Last time when the member sent a
message
      "Member_Account": "peter",
      "MsgFlag": "AcceptAndNotify",
      "MsgSeq": 7,
      "MuteUntil": 0, // `0`: the member is not muted; other values:
the time when the member will be unmuted
      "NameCard": "Peter",
      "Role": "Member"
    },
    {
      "AppMemberDefinedData": [ // Custom group member fields
        {
          "Key": "group_member_p",
          "Value": "the value"
        },
        {
          "Key": "group_member_p2",
          "Value": "the value2"
        }
      ],
      "JoinTime": 1728964923,
      "LastSendMsgTime": 1728973475,
```

```
        "Member_Account": "bob",
        "MsgFlag": "AcceptAndNotify",
        "MsgSeq": 7,
        "MuteUntil": 1728977081,    // Muting end time in seconds
        "NameCard": "bob",
        "Role": "Member",
    }
}
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed
ErrorInfo	String	Error information
GroupId	String	ID of the group to pull member information
MemberList	Array	Returned group member list, which contains information of all or specified group members fields. For details on group member information fields, see <a href="#">Group member profile</a> .
AppMemberDefinedData	Array	Returned custom group member fields

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10005	The number of group members queried exceeds 50. Please ensure that the number of group

	members requested is no more than 50.
10007	No operation permissions. Check whether the operator is an app admin or whether the operator has the permission to read the fields in the request.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Modifying the Profile of a Group Member ([v4/group\\_open\\_http\\_svc/modify\\_group\\_member\\_info](#))

# Modifying the Profile of a Group Member

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to modify the profile of a group member.

## API Call Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	No
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

Audio-video groups (AVChatRoom) do not store group member profiles and therefore do not allow member profile modification.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/modify_group_member_info?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_group_member_info	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Set an admin

Set a specified group member as an admin.

```
{
  "GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
  "Member_Account": "bob", // Target group member (required)
  "Role": "Admin" // Set as an admin
}
```

### Cancel an admin

Revoke a specified group member's admin role.

```
{
  "GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
  "Member_Account": "bob", // Target group member (required)
  "Role": "Member" //Cancel the admin role
}
```

```
}
```

### Set a member's message blocking type

Set a specified member's message blocking type, whose possible values are as follows. AcceptAndNotify: accept and notify. Discard: neither accept nor notify. AcceptNotNotify: accept but do not notify.

```
{
  "GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
  "Member_Account": "bob", // Target group member (required)
  "MsgFlag": "AcceptAndNotify" // Message blocking type, which can be
AcceptAndNotify, Discard, or AcceptNotNotify
}
```

### Set a member's group name card

Set a specified user's group name card, whose maximum length is 50 bytes.

```
{
  "GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
  "Member_Account": "bob", // Target group member (required)
  "NameCard": "bob" // Group name card (optional)
}
```

### Set member custom fields

Set group member custom fields. By default, AppMemberDefinedData is not available and needs to be enabled in the [Chat console](#) before use. For details, see the description table for request fields.

```
{
  "GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
  "Member_Account": "bob", // Target group member (required)
  "AppMemberDefinedData": [ // Target member custom field (optional)
    {
      "Key": "MemberDefined1", // Key of the target group member custom
field
      "Value": "ModifyData1" // Value of the key
    },
    {
      "Key": "MemberDefined3",
      "Value": "ModifyData3"
    }
  ]
}
```

### Set a group member's muting period

Set a specified group member's muting period.

```
{
```



```
"GroupId": "@TGS#2CLUZEAEJ", // Target group (required)
"Member_Account": "bob", // Target group member (required)
"MuteTime":86400 // Muting period for the specified user, in seconds
}
```

### Note

Private groups (same as Work groups in the new version) do not support muting group users.

### Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the Group whose historical messages will be modified
Member_Account	String	Required	Target group member.
Role	String	Optional	Role of the member. <code>Admin</code> : Set as an admin. <code>Member</code> : Cancel the admin role. (The group owner identity cannot be modified.)
MsgFlag	String	Optional	Message blocking type.
NameCard	String	Optional	Group name card (with a maximum length of 50 bytes).
AppMemberDefinedData	Array	Optional	Group member custom field. By default, this field is not available and needs to be enabled in the <a href="#">Chat console</a> . For details, see <a href="#">Group System</a> .
MuteTime	Integer	Optional	Muting period, in seconds. <code>0</code> : Unmute

### Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

### Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful. <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed

ErrorInfo	String	Error information
-----------	--------	-------------------

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. For example, a common member in a public group tries to remove other users from the group, but only the app admin can do so.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.

## API Debugging Tool

Use the [online debugging tool for RESTful APIs](#) to debug this API.

## References

Obtaining detailed information on group members ([v4/group\\_open\\_http\\_svc/get\\_group\\_member\\_info](#))

# Group Custom Attributes

## Getting Group Custom Attributes

Last updated : 2024-07-16 11:15:59

### Feature Overview

This API is used by the app admin to get custom group attributes.

### API Calling Description

#### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

#### Sample request URL

```
https://xxxxxx/v4/group_open_attr_http_svc/get_group_attr?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located.

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_attr_http_svc/get_group_attr</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Getting Custom Group Attributes

```
{
  "GroupId": "@TGS#aC5SZEAEF"
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the group for which you want to get custom attributes

## Sample response

```
{
```

```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0,
"GroupAttrAry": [
  {
    "key": "attr_key1",
    "value": "attr_val1"
  },
  {
    "key": "attr_key2",
    "value": "attr_val2"
  }
]
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. OK : successful; FAIL : failed
ErrorCode	Integer	Error code. 0 : Successful; other values: Failed
ErrorInfo	String	Error information
GroupAttrAry	Array	Key value pair for custom attributes

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. A group of this type does not support custom attributes.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.



# Modifying Group Custom Attributes

Last updated : 2024-07-16 11:16:00

## Feature Overview

This API is used by the app admin to modify custom group attributes.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/modify_group_attr?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_group_attr	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

Basic format

Modify custom attributes of a group.

```
{
  "GroupId": "@TGS#aC5SZEAEF",
  "GroupAttr": [
    {
      "key": "attr_key", //Attribute key
      "value": "attr_val" //Attribute value
    },
    {
      "key": "attr_key1",
      "value": "attr_val1"
    }
  ]
}
```

Request fields

--	--	--	--



Field	Type	Required	Description
GroupId	String	Yes	ID of the group for which you want to modify custom attributes
GroupAttr	Array	Yes	List of custom attributes. <code>key</code> : Key of the custom attribute, which can be up to 32 bytes in length. Up to 16 keys are supported. <code>value</code> : Value of the custom attribute, which can be up to 4,000 bytes in length.

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. This error occurs when, for example, a group of this type does not support custom attribute operations.

10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.
10045	The size of the custom attribute key exceeds the limit of 32 bytes.
10046	The size of the custom attribute value exceeds the limit of 4000 bytes.
10047	The number of custom attribute keys exceeds the limit of 16.
10048	The total size of custom attribute values exceeds the limit of 16000 bytes.
10049	The frequency of writing custom attributes to a group exceeds the limit. You can add, delete, or modify them for up to 5 times per second.

# Clearing Group Custom Attributes

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to clear custom group attributes.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/clear_group_attr?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/clear_group_attr</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Clear the custom attributes of a group:

```
{
  "GroupId": "@TGS#aC5SZEAEF"
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group whose custom attributes are to be cleared

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. This error occurs when, for example, a group of this type does not support custom attribute operations.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.
10045	The size of the custom attribute key exceeds the limit of 32 bytes.
10046	The size of the custom attribute value exceeds the limit of 4000 bytes.
10047	The number of custom attribute keys exceeds the limit of 16.
10048	The total size of custom attribute values exceeds the limit of 16000 bytes.
10049	The frequency of writing custom attributes to a group exceeds the limit. You can add, delete, or modify them for up to 5 times per second.

# Resetting Group Custom Attributes

Last updated : 2024-07-16 11:15:59

## Feature Overview

This API is used by the app admin to reset custom group attributes.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/set_group_attr?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code>

	Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/set_group_attr</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Reset the custom attributes of a group:

```
{
  "GroupId": "@TGS#aC5SZEAEF",
  "GroupAttr": [
    {
      "key": "attr_key", //Attribute key
      "value": "attr_val" //Attribute value
    },
    {
      "key": "attr_key1",
      "value": "attr_val1"
    }
  ]
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the group for which you want to reset custom attributes

GroupAttr	Array	Yes	List of custom attributes. <code>key</code> : Key of the custom attribute, which can be up to 32 bytes in length. Up to 16 keys are supported. <code>value</code> : Value of the custom attribute, which can be up to 4,000 bytes in length.
-----------	-------	-----	--

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	No operation permissions. This error occurs when, for example, a group of this type does not support custom attribute operations.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.



10045	The size of the custom attribute key exceeds the limit of 32 bytes.
10046	The size of the custom attribute value exceeds the limit of 4000 bytes.
10047	The number of custom attribute keys exceeds the limit of 16.
10048	The total size of custom attribute values exceeds the limit of 16000 bytes.
10049	The frequency of writing custom attributes to a group exceeds the limit. You can add, delete, or modify them for up to 5 times per second.

# Deleting Group Custom Attributes

Last updated : 2025-05-27 10:12:30

## Feature Overview

This API allows [app admins](#) to delete custom attributes of groups.

## API Call Description

### Applicable Group Types

Group Type ID	RESTful API Support
Private	Yes. Same as the Work group in the new version.
Public	Yes.
ChatRoom	Yes. Same as the Meeting group in the new version.
AVChatRoom	Yes.
Community	Yes.

These are built-in group types of Instant Messaging (IM). For more information, see [Group System](#).

### Sample Request URL

```
https://console.tim.qq.com/v4/group_open_http_svc/delete_group_attr?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The following table only lists parameters that can be modified when this API is called. For more information about the parameters, see [RESTful API Overview](#).

Parameter	Description
v4/group_open_http_svc/delete_group_attr	Request API.
sdkappid	SDKAppID assigned by the IM console when an app is created.

identifier	App admin account. For more information, see <a href="#">App Admin</a> .
usersig	Signature generated by the app admin account. For more information, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which is fixed at <code>json</code> .

## Maximum Call Frequency

200 calls per second

## Sample Request

### Basic Request

This example shows how to delete custom attributes of a group.

```
{
  "GroupId": "@TGS#aC5SZEAEF",
  "GroupAttr":[
    {
      "key":"attr_key" // Attribute key
    },
    {
      "key":"attr_key1"
    }
  ]
}
```

## Request Fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group of which custom attributes are to be deleted.
GroupAttr	Array	Yes	List of custom attributes to be deleted, with the keys of the custom attributes.

## Sample Response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

```
}

```

Response Fields

Field	Type	Description
ActionStatus	String	Request result.- OK: Successful- FAIL: Failed
ErrorCode	Integer	Error code.- 0: Successful- Other values: Failed
ErrorInfo	String	Error information.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and error information are indicated by ErrorCode and ErrorInfo respectively in the response.

For public error codes (60000 to 79999), see [Error Codes](#) .

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and rectify the fault.
10007	No operation permission. For example, a group of this type does not have the operation permission on custom attributes.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Use the correct group ID.
10045	The size of the custom attribute key exceeds the limit of 32 bytes.
10049	The frequency of writing custom attributes to a group exceeds the limit. You can add, delete, or modify the attributes for up to 5 times per second.
10050	Failed to delete a non-existent custom attribute.

# Live Group Management

## Setting Live Streaming Group Robots

Last updated : 2025-03-20 16:15:52

### Feature Overview

App administrators and group owners can add robot accounts to the live streaming group's online member list. The robot accounts can be fixed in the online member list without needing to call the group joining API, regardless of their online status. This API essentially sets a special tag (500) for robot accounts. For details, refer to the API [Setting Live Streaming Group Member Tags](#). This feature requires the Pro edition 、 Pro Plus edition or Enterprise edition, and the live streaming group online member list feature must have been enabled under group feature configuration in the Chat console. Calling this API will return a failure for other package editions.

### API Call Description

#### Applicable Group Types

Group Type ID	REST API Support
Private	No. It is similar to Work (friend work group) in the new version.
Public	No.
ChatRoom	No. It is similar to Meeting (temporary meeting group) in the new version.
AVChatRoom	Yes.
Community	No.

Chat incorporates the aforementioned group types. For details, see [Group System](#).

#### Note

This feature requires the [Pro edition 、 Pro Plus edition or Enterprise edition Package](#), and the live streaming group online member list feature must have been enabled (under group feature configuration in the console).

Calling this API will return a failure for other package editions.

This API essentially sets a special tag (500) for robot accounts. For details, refer to the API [Setting Live Streaming Group Member Tags](#).

## Sample Request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/add_robots?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request Parameters

The following table only lists the parameters involved when this API is called and their descriptions. For details on other parameters, see [REST API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_avchatroom_http_svc/add_robots	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	It must be an app admin account. For details, see <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The request format is fixed to <code>json</code> .

## Maximum Call Frequency

200 times/second

## Sample Request Packets

Add two robots, bot1 and bot2, to a live streaming group:

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
```

```
"Member_Account": ["bot1", "bot2"]
}
```

## Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the operated group.
Member_Account	Array	Required	List of robot accounts to be set.

## Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": ""
}
```

## Response Packet Fields

Field	Type	Description
ActionStatus	String	Processing results: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error message.

## Error Codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packets.

Common error codes (60000 to 79999) can be found in [Error Codes](#).

The private error codes for this API are as follows:

Error Code	Description

10001	The user does not log in. Please check whether the admin account and usersig are correct.
10002	System error. Please try again or contact technical support.
10003	Invalid request command. Please try again or contact technical support.
10004	Invalid parameters. Please check whether the required fields are filled in or whether the field fillings meet the protocol requirements based on the ErrorInfo field in the response packets.
10007	Insufficient permissions. This API must be called by an app administrator or group owner.
10010	The group does not exist, or it once existed but has now been dissolved.
10015	Invalid group ID. Please check whether the group ID is filled in correctly.



# Deleting Live Streaming Group Robots

Last updated : 2025-02-28 17:38:58

## Feature Overview

App administrators and group owners can delete previously added robot accounts from the live streaming group's online member list. This API essentially removes the special tag (500) for the robot account. For details, refer to the API [Setting Live Streaming Group Member Tags](#). This feature requires the Pro edition 、 Pro Plus edition or Enterprise edition, and the live streaming group online member list feature must have been enabled under group feature configuration in the Chat console. Calling this API will return a failure for other package editions.

## API Call Description

### Applicable Group Types

Group Type ID	REST API Support
Private	No. It is similar to Work (friend work group) in the new version.
Public	No.
ChatRoom	No. It is similar to Meeting (temporary meeting group) in the new version.
AVChatRoom	Yes.
Community	No.

Chat incorporates the aforementioned group types. For details, see [Group System](#).

### Note

This feature requires the [Pro edition 、 Pro Plus edition or Enterprise edition Package](#), and the **live streaming group online member list** feature must have been enabled (under **group feature configuration** in the console).

Calling this API will return a failure for other package editions.

This API essentially removes the special tag (500) for the robot account. For details, refer to the API [Setting Live Streaming Group Member Tags](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/del_robots?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request Parameters

The following table only lists the parameters involved when this API is called and their descriptions. For details on other parameters, see [REST API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_avchatroom_http_svc/del_robots	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	It must be an app admin account. For details, see <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The request format is fixed to <code>json</code> .

## Maximum Call Frequency

200 times/second

## Sample Request Packets

Delete bots bot1 and bot2 from a live streaming group:

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
  "Member_Account": ["bot1", "bot2"]
}
```

## Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the operated group.
Member_Account	Array	Required	List of robot accounts to be deleted.

## Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": ""
}
```

## Response Packet Fields

Field	Type	Description
ActionStatus	String	Processing results: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error message.

## Error Codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packets.

Common error codes (60000 to 79999) can be found in [Error Codes](#).

The private error codes for this API are as follows:

Error Code	Description
10001	The user does not log in. Please check whether the admin account and usersig are correct.
10002	System error. Please try again or contact technical support.

10003	Invalid request command. Please try again or contact technical support.
10004	Invalid parameters. Please check whether the required fields are filled in or whether the field fillings meet the protocol requirements based on the ErrorInfo field in the response packets.
10007	Insufficient permissions. This API must be called by an app administrator or group owner.
10010	The group does not exist, or it once existed but has now been dissolved.
10015	Invalid group ID. Please check whether the group ID is filled in correctly.

# Setting/Deleting Live Streaming Group Administrators

Last updated : 2025-03-11 11:36:39

## Feature Overview

App administrators can set and delete administrators for a live streaming group. When an account is designated as an administrator, it does not need to be in the live streaming group. Once set as an administrator, the account remains an administrator even if it leaves and re-enters the live streaming group. To cancel the administrator status, this API needs to be called. This feature requires the Pro edition 、 Pro Plus edition or Enterprise edition, and the live streaming group online member list feature must have been enabled under group feature configuration in the Chat console.

## API Call Description

### Applicable Group Types

Group Type ID	REST API Support
Private	No. It is similar to Work (friend work group) in the new version.
Public	No.
ChatRoom	No. It is similar to Meeting (temporary meeting group) in the new version.
AVChatRoom	Yes.
Community	No.

Chat incorporates the aforementioned group types. For details, see [Group System](#).

### Note

This feature requires the [Pro edition 、 Pro Plus edition or Enterprise edition Package](#), and the live streaming group online member list feature must have been enabled (under group feature configuration in the console).  
When an account is designated as an administrator, it does not need to be in the live streaming group.  
Once set as an administrator, the account remains an administrator even if it leaves and re-enters the live streaming group.  
To cancel the administrator status, this API needs to be called.

A live streaming group can have up to 5 administrators.

## Sample Request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/modify_admin?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request Parameters

The following table only lists the parameters involved when this API is called and their descriptions. For details on other parameters, see [REST API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_avchatroom_http_svc/modify_admin	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	It must be an app admin account. For details, see <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The request format is fixed to <code>json</code> .

## Maximum Call Frequency

200 times/second

## Sample Request Packets

Query whether a batch of users is in the live streaming group, and return the users who are in the group.

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
  "CommandType": 1,
  "Admin_Account": ["user1"]
}
```

Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the operated group.
CommandType	Integer	Required	1: Sets an administrator. 2: Cancels administrator settings.
Admin_Account	Array	Required	UserID list of administrators to be modified. A live streaming group can have up to 5 administrators.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": ""
}
```

Response Packet Fields

Field	Type	Description
ActionStatus	String	Processing results: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error message.

Error Codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packets.

Common error codes (60000 to 79999) can be found in [Error Codes](#).

The private error codes for this API are as follows:

Error Code	Description
10001	The user does not log in. Please check whether the admin account and usersig are correct.
10002	System error. Please try again or contact technical support.
10003	Invalid request command. Please try again or contact technical support.
10004	Invalid parameters. Please check whether the required fields are filled in or whether the field fillings meet the protocol requirements based on the ErrorInfo field in the response packets.
10007	Insufficient permissions. Non-AVChatRoom group types do not support obtaining the online user count.
10010	The group does not exist, or it once existed but has now been dissolved.
10008	The number of admin in the AVChatRoom has exceeded the limit. Or, the Pro edition, Pro Plus edition or Enterprise edition package has not been activated, and the live streaming group online member list feature has not been enabled in the group feature configuration on the console.
10015	Invalid group ID. Please check whether the group ID is filled in correctly.



# Obtaining the List of Live Streaming Group Administrators

Last updated : 2025-02-28 17:40:50

## Feature Overview

App administrators can obtain the list of live streaming group administrators by using the group ID. This feature is exclusively available for **Pro edition** 、 **Pro Plus edition** or **Enterprise edition** users who have enabled the live streaming group online member list feature under group feature configuration in the Chat console.

## API Call Description

### Applicable Group Types

Group Type ID	REST API Support
Private	No. It is similar to Work (friend work group) in the new version.
Public	No.
ChatRoom	No. It is similar to Meeting (temporary meeting group) in the new version.
AVChatRoom	Yes.
Community	No.

Chat incorporates the aforementioned group types. For details, see [Group System](#).

### Note

This feature requires the [Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition Package](#), and the live streaming group online member list feature must have been enabled (under group feature configuration in the console).

### Sample Request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/get_admin_list?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The following table only lists the parameters involved when this API is called and their descriptions. For details on other parameters, see [REST API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/group_open_avchatroom_http_svc/get_admin_list	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	It must be an app admin account. For details, see <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The request format is fixed to json .

### Maximum Call Frequency

200 times/second

### Sample Request Packets

Obtain the online user count of the AVChatRoom group, with the update granularity of the online count being 40 seconds.

```
{
  "GroupId": "@TGS#a6I4ZUUGO"
}
```

### Request Packet Fields

Field	Type	Attribute	Description
-------	------	-----------	-------------

GroupId	String	Required	Live streaming group ID.
---------	--------	----------	--------------------------

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "Admin_List": [
    {
      "Admin_Account": "user1",
      "Avatar": "https://example.com/abc.jpg",
      "NickName": "AA"
    },
    {
      "Admin_Account": "116400",
      "Avatar": "",
      "NickName": "Charles 2"
    }
  ]
}
```

Response Packet Fields

Field	Type	Description
ActionStatus	String	Processing results: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error message.
Admin_List	Object	The list of live streaming group administrators for this group.

Error Codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packets.

Common error codes (60000 to 79999) can be found in [Error Codes](#).

The private error codes for this API are as follows:

Error Code	Description
10001	The user does not log in. Please check whether the admin account and usersig are correct.
10002	System error. Please try again or contact technical support.
10003	Invalid request command. Please try again or contact technical support.
10004	Invalid parameters. Please check whether the required fields are filled in or whether the field fillings meet the protocol requirements based on the ErrorInfo field in the response packets.
10007	Insufficient permissions. Non-AVChatRoom group types do not support obtaining the online user count.
10010	The group does not exist, or it once existed but has now been dissolved.
10015	Invalid group ID. Please check whether the group ID is filled in correctly.

# Checking Whether Users Are in a Live Streaming Group

Last updated : 2025-02-28 17:40:51

## Feature Overview

App administrators can check whether a batch of users is in a live streaming group by using the group ID. This feature requires the Pro edition 、 Pro Plus edition or Enterprise edition, and the live streaming group online member list feature must have been enabled under group feature configuration in the Chat console.

## API Call Description

### Applicable Group Types

Group Type ID	REST API Support
Private	No. It is similar to Work (friend work group) in the new version.
Public	No.
ChatRoom	No. It is similar to Meeting (temporary meeting group) in the new version.
AVChatRoom	Yes.
Community	No.

Chat incorporates the aforementioned group types. For details, see [Group System](#).

### Note

This feature requires the [Pro edition 、 Pro Plus edition or Enterprise edition Package](#), and the live streaming group online member list feature must have been enabled (under group feature configuration in the console).

The overall update granularity of online members is 40s.

When there are more than 1,000 people in the live streaming group, the API can still query whether members outside of the 1,000 people are in the live streaming group.

### Sample Request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/check_members?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request Parameters

The following table only lists the parameters involved when this API is called and their descriptions. For details on other parameters, see [REST API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_avchatroom_http_svc/check_members	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	It must be an app admin account. For details, see <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The request format is fixed to <code>json</code> .

## Maximum Call Frequency

200 times/second

## Sample Request Packets

Query whether a batch of users is in the live streaming group, and return the users who are in the group.

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
  "Member_Account": ["245000", "202200", "239017", "361697"]
}
```

```
}

```

Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the operated group.
Member_Account	Array	Required	UserID list of queried users.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "Member_Account": ["245000", "239017"]
}

```

Response Packet Fields

Field	Type	Description
ActionStatus	String	Processing results: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error message.
Member_Account	Array	UserID list of users in the live streaming group.

Error Codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packets. Common error codes (60000 to 79999) can be found in [Error Codes](#).

The private error codes for this API are as follows:

Error	Description
-------	-------------

Code	
10001	The user does not log in. Please check whether the admin account and usersig are correct.
10002	System error. Please try again or contact technical support.
10003	Invalid request command. Please try again or contact technical support.
10004	Invalid parameters. Please check whether the required fields are filled in or whether the field fillings meet the protocol requirements based on the ErrorInfo field in the response packets.
10007	Insufficient permissions. Non-AVChatRoom group types do not support obtaining the online user count.
10010	The group does not exist, or it once existed but has now been dissolved.
10015	Invalid group ID. Please check whether the group ID is filled in correctly.



# Getting the Number of Online Users in an Audio-Video Group

Last updated : 2024-12-09 15:33:37

## Feature Description

The app admin can use the group ID to obtain the number of online users in a live stream group.

## API Description

### Applicable group types

Group Type ID	REST API Support
Private	Yes. Same as Work (work group) in the new version.
Public	Yes.
ChatRoom	Yes. Same as Meeting (temporary meeting group) in the new version.
AVChatRoom	Yes.
Community	Yes.

Above are the IM built-in groups. For more information, see [Group system](#).

### Note

The number of online users is updated every 10s.  
When there are web users, the number of online users will be updated 10s to 20s after a user joins or quits the group.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_online_member_num?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table only describes the modified parameters when this API is called. For more information on other parameters, please see [RESTful API Overview](#).

--	--

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/get_online_member_num	The API to which the request is sent.
sdkappid	The SDKAppID assigned by the IM console when an application is created.
identifier	Must be the app admin account. For more information, please see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	The signature generated by the app admin account. For more information on the operation, please see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is always <code>json</code> .

## Maximum call frequency

200 times/second

## Sample request packet

This API is used to obtain the number of online users in a live stream group (AVChatRoom). The number of online users is updated every 10s. We recommend that the service-end queries the number of online users every 10s and synchronizes it to all users in the group.

```
{
  "GroupId": "@TGS#a6I4ZUUGO"
}
```

## Request packet fields

Field	Type	Required	Description
GroupId	String	Yes	The group ID.

Sample response packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "OnlineMemberNum": 1000 // Number of online users
}
```

Response packet fields

Field	Type	Description
ActionStatus	String	The request result. <code>OK</code> : succeeded. <code>FAIL</code> : failed.
ErrorCode	Integer	The error code. <code>0</code> : succeeded. Other values: failed.
ErrorInfo	String	The error information.
OnlineMemberNum	Integer	The number of online users in the group.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes 60000 to 79999, please see [Error Codes](#). The following table describes the error codes specific to this API.

Error Code	Description
10002	A system error occurred. Try again or contact technical support.
10003	The request command is invalid. Try again or contact technical support.
10004	A parameter is invalid. Check the <code>ErrorInfo</code> field in the response packet to see whether the required fields have been specified or whether the fields are set according to protocol requirements.
10007	You do not have required permissions. You can only obtain the number of online users for live stream

	groups.
10010	The group does not exist or has been disbanded.
10015	The group ID is invalid. Please check the group ID.

## Debugging Tool

Use the [REST API online debugging tool](#) to debug this API.

# Getting the List of Online Members in Audio-Video Group

Last updated : 2025-02-28 17:40:51

## Feature Overview

This API is used to get the list of online members in an audio-video group based on the group ID. To use this feature, you need to purchase the Pro edition 、 Pro Plus edition or Enterprise edition, go to the Chat console, select **Group feature configuration**, and enable **List of online audio-video group members**. In other plan editions, the list of up to 30 latest group members will be returned after this API is called.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	No. Same as work groups (Work) in the new version.
Public	No
ChatRoom	No. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	Yes

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

To use this feature, you need to [purchase the Pro edition 、 Pro Plus edition or Enterprise edition](#) and enable the **List of online audio-video group members** feature on the **Group feature configuration** page in the console.

For other plan editions, the list of up to 30 latest group members will be returned after this API is called.

The list of group members is updated once every 10 seconds.

When there are more than 1,000 members in an audio-video group, only the 1,000 latest online members will be returned after this API is called.

When the group size is greater than or equal to 300, or when there are web users, the number of online users will be updated 10s to 20s after a user joins or quits the group.

When the group size is less than 300 and when there are no web users, the number of online users is updated immediately after a user joins or quits the group.

## Sample request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/get_members?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_avchatroom_http_svc/get_members	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

This API is used to get the number of online members in an audio-video group (AVChatRoom), which is collected once every 10 seconds.

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
  "Timestamp": 0
}
```

Only the group members marked as 1000 will be obtained.

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
  "Mark": 1000,
  "Timestamp": 0
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the target group
Timestamp	Integer	Yes	For the first request, the value is 0 . If the audio-video group has 500 or less members, the backend returns all members and NextTimestamp with the value 0 . If the audio-video group has more than 500 members, the backend returns NextTimestamp with a non-zero value, which is used as the value of Timestamp for the next request for pulling the remaining members.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "MemberList": [
    {
      "JoinTime": 1657773110,
      "Member_Account": "teacher2",
      "NickName": "Teacher t02",
      "Avatar": "http://www.example.com/teacher2.jpg"
    }
  ],
  "NextTimestamp": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MemberList	Object	Online accounts of the group. <code>Member_Account</code> is the <code>UserID</code> of a user, and <code>JoinTime</code> is the time when the user joined the group.
NextTimestamp	Integer	Pulling-by-page flag. If the value is not <code>0</code> , more members can be returned. You need to set the value in the <code>Timestamp</code> request parameter to pull more members. If it is <code>0</code> , all the members have been returned.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10001	The user is not logged in. Check whether the <code>admin</code> account and <code>usersig</code> are correct.
10002	A system error occurred. Try again or contact technical support.
10003	The request command is invalid. Try again or contact technical support.
10004	A parameter is invalid. Check the <code>ErrorInfo</code> field in the response packet to see whether the required fields have been specified or whether the fields are set according to protocol requirements.
10007	You do not have required permissions. You can only obtain the number of online users for audio-video groups.
10010	The group does not exist or has been deleted.
10015	The group ID is invalid. Please check the group ID.





# Setting Audio-Video Group Member Marks

Last updated : 2025-01-14 14:42:11

## Feature Overview

App admins and group owners can set different marks/labels for audio-video group members to distinguish different types of group members. To use this feature, you need to purchase the Pro edition 、 Pro Plus edition or Enterprise edition, enable **List of online audio-video group members** on the **Group configuration** page of the Chat console. For other plan editions, calling this API will fail.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	No. Same as work groups (Work) in the new version.
Public	No
ChatRoom	No. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

To use this feature, you need to [purchase the Pro edition 、 Pro Plus edition or Enterprise edition](#) and enable the **List of online audio-video group members** feature on the **Group configuration** page in the console.

For other plan editions, calling this API will fail.

For audio-video group, up to 10 different marks can be set.

Each mark can be set for up to 1,000 members. This limit is independent from the 1,000-member limit on the list of online members of an audio-video group.

Marks can be set only for online group members, except the special mark 500, which is described in the "Special marks" part. When a member leaves a group or goes offline, its mark information will be cleared. When the member joins the group or goes online again, you can reset a mark for the member through a webhook.

## Sample request URL

```
https://xxxxxx/v4/group_open_avchatroom_http_svc/modify_user_info?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_group_member_info	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

Set mark 1000 for member "teacher10" and mark 1001 for member "student9" of an audio-video group:

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
```

```
"CommandType": 1,
"MemberList": [
  {
    "Member_Account": "teacher10",
    "Marks": [1000]
  },
  {
    "Member_Account": "student9",
    "Marks": [1001]
  },
]
```

Cancel mark 1000 for member "teacher10" and mark 1001 for member "student9" of the audio-video group:

```
{
  "GroupId": "@TGS#a6I4ZUUGO",
  "CommandType": 2,
  "MemberList": [
    {
      "Member_Account": "teacher10",
      "Marks": [1000]
    },
    {
      "Member_Account": "student9",
      "Marks": [1001]
    },
  ]
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	The ID of the Group whose historical messages will be modified
CommandType	Integer	Yes	1 : Mark setting; 2 : Mark deletion
MemberList	Array	Yes	List of accounts for which marks are to be set. Up to 500 accounts are supported each time, and up to 10 different marks can be set for each group. A mark is a number greater than or equal to 1,000.

## Sample response

```
{
  "ActionStatus": "OK",
}
```

```
"ErrorCode": 0,
"ErrorInfo": "",
"CommandType": 1,
"MemberList": [
  {
    "Member_Account": "teacher10",
    "Marks": [1000]
  },
  {
    "Member_Account": "student9",
    "Marks": [1001]
  }
]
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
MemberList	Object	List of members whose marks are successfully set. In the case of mark setting for a batch of accounts, if certain accounts are offline, the backend does not return the offline accounts; if all accounts are offline or do not exist, the backend returns a failure result.

## Special marks

App admins and group owners can set marks within the range [1000, +) for group members. In addition, app admins can set the following special marks that are preset in Chat:

Mark	Description
500	When mark 500 is set for a member, the member is considered to be always online (disconnection events ignored) for a maximum of three days.
600	When a member is set with mark 600, the member is hidden when the list of online members is pulled, unless the list is explicitly specified to be pulled by the member's mark.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10001	The user is not logged in. Check whether the <code>admin</code> account and <code>usersig</code> are correct.
10002	A system error occurred. Try again or contact technical support.
10003	The request command is invalid. Try again or contact technical support.
10004	A parameter is invalid. Check the <code>ErrorInfo</code> field in the response packet to see whether the required fields have been specified or whether the fields are set according to protocol requirements.
10007	Insufficient permission. Only the app admin or group owner can call this API.
10010	The group does not exist or has been deleted.
10015	The group ID is invalid. Please check the group ID.

# Getting the List of Banned Group Members

Last updated : 2025-03-20 16:16:39

## Feature Overview

You can use this API to get the list of banned members of an audio-video group.

### Note

Audio-video group member banning is supported by SDK Enhanced edition v6.6 or later and web SDK v2.22.0 or later. To use it, you need to [purchase the Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#), go to the [console](#), choose **Feature configuration > Group configuration > Group feature configuration**, and enable **Audio-video group member banning**.

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	No. Same as work groups (Work) in the new version.
Public	No
ChatRoom	No. Same as meeting groups (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_ban_member?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/get_group_ban_member	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

It is used to deliver broadcast messages to all the audio-video groups.

```
{
  "GroupId": "@TGS#aJRGc4MH6", // Group ID
  "Limit":20, // Number of banned members to be pulled per page each time.
The maximum value allowed is 100.
  "Offset": 0 // Offset. For the first request, set `Offset` to 0. For
subsequent requests, set `Offset` to the value of `NextOffset` in the response
packet. If `NextOffset` is 0, the entire list of banned members of the audio-
video group is obtained.
}
```



## Request fields

Field	Type	Required	Description
GroupId	String	Yes	ID of the group whose members are to be unbanned
Limit	Integer	No	Number of unbanned users to be obtained each time. The maximum value allowed is 100.
Offset	Integer	Yes	Offset. For the first request, set <code>Offset</code> to 0. For subsequent requests, set <code>Offset</code> to the value of <code>NextOffset</code> in the response packet.

## Sample response

```
{
  "ActionStatus": "OK",
  "BannedAccountList": [
    {
      "BannedUntil": 1660875916,
      "Member_Account": "brennanli1"
    },
    {
      "BannedUntil": 1660875916,
      "Member_Account": "brennanli12"
    }
  ],
  "ErrorCode": 0,
  "ErrorInfo": "",
  "NextOffset": 2
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
BannedAccountList	Array	Information of banned members. <code>BannedUntil</code> indicates the banning end time of banned members. <code>Member_Account</code> indicates the account IDs of banned members.
NextOffset	Integer	Offset value of the next request. If this field is <code>0</code> , the list of banned

members of the AVChatRoom group is fully pulled.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10005	The number of members to be unbanned exceeds the maximum limit of 20 members per request.
10007	Insufficient operation permissions. For example, the switch is not enabled in the console, or the operating account is not the root account.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Community Management

## Creating Topic

Last updated : 2025-01-14 14:46:03

### Feature Overview

This API is used by the app admin to create a topic.

### API Calling Description

#### Applicable group types

Group Type ID	Whether This RESTful API Is Supported
Private	No
Public	No
ChatRoom	No
AVChatRoom	No
Community	This API applies only to topic-enabled communities.

These are the preset group types in Chat. For more information, see [Group System](#).

#### Note:

To use the topic feature, you need to go to the [console](#), choose **Feature Configuration > Group configuration > Group feature configuration > Community**, enable the community feature and then enable the topic feature.

#### Sample request URL

```
https://xxxxxx/v4/million_group_open_http_svc/create_topic?sdkappid=88888888&identi
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/million_group_open_http_svc/create_topic	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Create a topic. Here, `GroupId` is required, which indicates the group of the topic. Before creating a topic, a topic-enabled group must exist.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group ID of the topic, which is required
  "TopicName": "TestTopic"              // Topic name, which is required
}
```

### Customize the topic ID

To simplify the topic ID, Tencent Cloud allows customizing it in the format of "GroupId+@TOPIC#\_+Custom part" during topic creation through the RESTful API in the application. For more information, see [Group System](#).

```
{
```

```
"GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group ID of the topic, which is required
"TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic", // Custom topic ID,
"TopicName": "TestTopic" // Topic name, which is required
}
```

## ALL IN ONE

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group ID of the topic, which is required
  "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic", // Custom topic ID,
  "TopicName": "TestTopic", // Topic name, which is required
  "From_Account": "1400187352", // Member creating the topic
  "CustomString": "This is a custom string", // Custom string
  "FaceUrl": "http://this.is.face.url", // (Optional) Topic profile photo URL
  "Notification": "This is topic Notification", // (Optional) Topic notice
  "Introduction": "This is topic Introduction" // (Optional) Topic introduction
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Required	Group ID of the topic to be created
TopicId	String	No	To simplify topic IDs and make them easy to remember, Tencent Cloud allows apps to customize topic IDs during topic creation through RESTful APIs. For details, see <a href="#">here</a> .
TopicName	String	Yes	Topic name, whose maximum length is 30 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
From_Account	uint64	No	User account that wants to create the topic.
CustomString	String	No	Custom string, which can contain up to 3,000 bytes, encoded in UTF-8.
FaceUrl	String	No	URL of the topic profile photo, whose maximum length is 100 bytes.
Notification	String	No	Topic notice, whose maximum length is 300 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
Introduction	String	No	Topic introduction, whose maximum length is 240 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.

## Sample response

### Basic form, containing only basic topic information

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#cETE3HIM62CQ"
}
```

### Custom ID and **ALL IN ONE** of the topic

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic"
}
```

### Response fields

Field	Type	Description
ActionStatus	String	Request result. <b>OK</b> : successful; <b>FAIL</b> : failed
ErrorCode	Integer	Error code. <b>0</b> : Successful; other values: Failed
ErrorInfo	String	Error information
TopicId	String	ID of the successfully created topic, which can be customized or assigned by the Chat backend.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields **ErrorCode** and **ErrorInfo** respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.

10004	Invalid parameter. Check the error description and troubleshoot the issue.
10006	The operation exceeds the frequency limit. Please reduce the call frequency. This error is usually caused by too much net group increase in a single day or too frequent calls to obtain all groups in the app.
10007	Insufficient operation permissions. Check the request parameters based on the error message.
10008	The request is invalid, probably because verification of the signing information in the request failed. Please try again or <a href="#">submit a ticket</a> .
10010	The current group does not exist or has been deleted.
10015	The requested group ID is invalid. Check the request parameter based on the error message.
10016	The application backend rejected this operation through a wehook. Check the returned value of the webhook before topic creation.
10025	You have already used this group ID. Disband the existing group first or select another group ID.
10036	The number of audio-video groups (AVChatRoom) created exceeds the limit. Delete some of the groups or purchase an upgrade by referring to <a href="#">Pricing</a> .
10037	The number of prepaid topics created exceeds the limit. Delete some of the topics or purchase an upgrade by referring to <a href="#">Pricing</a> .
11000	The current group does not support the community topic feature. To use this feature, you need to purchase the <a href="#">Pro edition</a> 、 <a href="#">Pro Plus edition</a> or <a href="#">Enterprise edition</a> and <a href="#">enable it in the console</a> .
80001	Failed to pass the security check. Check the request parameters based on the error message.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Possible Webhooks

[Before a Topic Is Created](#)

[After a Topic Is Created](#)

# Deleting Topic

Last updated : 2025-01-14 14:46:02

## Feature Overview

This API is used to delete a topic.

## API Calling Description

### Applicable group types

Group Type ID	Whether This RESTful API Is Supported
Private	No
Public	No
ChatRoom	No
AVChatRoom	No
Community	This API applies only to topic-enabled communities.

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

To use the topic feature, you need to go to the [console](#), choose **Feature Configuration > Group configuration > Group feature configuration > Community**, enable the community feature and then enable the topic feature.

### Sample request URL

```
https://xxxxxx/v4/million_group_open_http_svc/destroy_topic?sdkappid=88888888&ident
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your



	SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/million_group_open_http_svc/destroy_topic	Request API
sdkappid	The SDKAppID assigned by the Chat console when the application is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. Fixed value: json .

Maximum call frequency

200 calls per second

Sample request

Note that a topic cannot be recovered once deleted through this API.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group ID of the topic to be deleted
  "TopicIdList": [ // List of IDs of the topics to be deleted
    "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic",
    "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic_1"
  ]
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Required	Group ID of the topic to be deleted

TopicIdList	Array	Required	List of IDs of the topics to be deleted
-------------	-------	----------	---

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "ok",
  "ErrorCode": 0,
  "DestroyResultItem": [
    {
      "ErrorCode": 0,
      "ErrorInfo": "ok",
      "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic"
    },
    {
      "ErrorCode": 0,
      "ErrorInfo": "ok",
      "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_TestTopic_1"
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
DestroyResultItem	Array	Returned result of topic deletion. Each item indicates the deletion result of a topic.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#). The following table describes the error codes specific to this API:

Error	Description
-------	-------------

Code	
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10006	The operation exceeds the frequency limit. Please reduce the call frequency. This error is usually caused by too much net group increase in a single day or too frequent calls to obtain all groups in the app.
10007	Insufficient operation permissions. Check the request parameters based on the error message.
10008	The request is invalid, probably because verification of the signing information in the request failed. Please try again or <a href="#">submit a ticket</a> .
10015	The requested group ID is invalid. Check the request parameter based on the error message.
10021	The group ID has already been used by another user. Select another group ID.
10025	You have already used this group ID. Disband the existing group first or select another group ID.
11000	The current group does not support the community topic feature. To use this feature, you need to purchase the <a href="#">Pro edition</a> 、 <a href="#">Pro Plus edition or Enterprise edition</a> and <a href="#">enable it in the console</a> .
110002	Failed to delete the topic. Check based on the error message.
110003	Failed to delete the topic. The topic has been deleted.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Possible Webhooks

[After a Topic Is Deleted](#)

# Getting Topic Profile

Last updated : 2025-01-14 14:46:02

## Feature Overview

This API is used to get the topic profile.

## API Calling Description

### Applicable group types

Group Type ID	Whether This RESTful API Is Supported
Private	No
Public	No
ChatRoom	No
AVChatRoom	No
Community	This API applies only to topic-enabled communities.

These are the preset group types in Chat. For more information, see [Group System](#).

### Note

To use the topic feature, you need to go to the [console](#), choose **Feature Configuration > Group configuration > Group feature configuration > Community**, enable the community feature and then enable the topic feature.

### Sample request URL

```
https://xxxxxx/v4/million_group_open_http_svc/get_topic?sdkappid=88888888&identifie
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your

	SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/million_group_open_http_svc/get_topic</code>	Request API
<code>sdkappid</code>	The <code>SDKAppID</code> assigned by the Chat console when the application is created
<code>identifier</code>	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
<code>usersig</code>	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295.
<code>contenttype</code>	Request format. Fixed value: <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Used to get the topic details. The request contains the group ID and the list of all the topics to be pulled.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ",          // You need to get the group ID of
  "From_Account": "1400187352" // Get the information of the specified user in the
}
```

### Specify the form of topic list to be pulled

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // You need to get the group ID of the topic
  "From_Account": "1400187352", // Get the information of the specified user in the
  "TopicIdList": [              // Specify the topic list to be pulled, which is optional
    "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#cRTE3HIM62C5"
  ]
}
```

## Use the filter to get the specified information form

You can specify the topic field to be pulled in `ResponseFilter` . If a filter is set, fields not specified in the filter will not be pulled.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // You need to get the group ID of the topic
  "From_Account": "1400187352", // Get the information of the specified user in the topic
  "TopicIdList": [ // Specify the topic list to be pulled, which is optional
    "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#cRTE3HIM62C5"
  ],
  "ResponseFilter": [ // Response filter. Specify the topic information to be pulled
    "TopicName", // Topic name
    "FaceUrl", // Profile photo URL of the topic
    "Notification", // Topic notice
    "Introduction" // Topic introduction
  ]
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Required	You need to get the group ID of the topic.
From_Account	String	Required	Get the information of the specified user in the topic.
TopicIdList	Array	Optional	You need to get the topic list. If it is left empty, all the topic information of the group will be obtained.
ResponseFilter	Array	Optional	Response filter. Specify the topic information to be obtained.

## Sample response

### Basic form; specify the form of topic list to be obtained

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "ok", // The `ErrorInfo` parameter here is meaningless. You need to check the actual result.
  "ErrorCode": 0, // The `ErrorCode` parameter here is meaningless. You need to check the actual result.
  "TopicAndSelfInfo": [ // The returned result is a topic information array. Only one topic is returned.
    {
      "ErrorCode": 0,
      "ErrorInfo": "ok", // Return result specific to this topic
      "SelfInfo": { // Information of the specified user in the topic
        "ShutUpTime": 0, // Muting period of the user
      }
    }
  ]
}
```

```

        "MsgSeq": 0, // Sequence number of the member's read message
        "MsgFlag": "AcceptAndNotify"
    },
    "TopicInfo": { // Returned topic information
        "TopicName": "TestTopicName", // Topic name
        "GroupID": "@TGS#_@TGS#cQVLVHIM62CJ", // Group ID of the topic
        "FaceUrl": "http://this.is.new.face.url", // URL of the topic profile
        "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#cRTE3HIM62C5", //
        "NextMsgSeq": 1,
        "Notification": "Notification", // Topic notice
        "Introduction": "Introduction", // Topic introduction
        "CreateTime": 1658839448, // Topic creation time in UTC
        "LastMsgTime": 0, // UTC time when the last message in the topic was s
        "MuteAllMember": "Off", // Flag indicating whether to mute all in th
        "CustomString": "This is a customs string." // Custom string in
    }
}
]
}

```

### Use the filter to get the specified topic information form

```

{
    "ActionStatus": "OK",
    "ErrorInfo": "ok", // The `ErrorInfo` parameter here is meaningless. You need to
    "ErrorCode": 0, // The `ErrorCode` parameter here is meaningless. You need to c
    "TopicAndSelfInfo": [ // The returned result is a topic information array
        {
            "ErrorCode": 0,
            "ErrorInfo": "ok", // Return result specific to this topic
            "SelfInfo": { // Information of the specified user in the topic
                "ShutUpTime": 0, // Muting period of the user
                "MsgSeq": 0,
                "MsgFlag": "AcceptAndNotify"
            },
            "TopicInfo": { // Returned topic information
                "TopicName": "TestTopicName", // Topic name
                "GroupID": "@TGS#_@TGS#cQVLVHIM62CJ", // Group ID of the topic
                "FaceUrl": "http://this.is.new.face.url", // URL of the topic profil
                "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#cRTE3HIM62C5",
                "Notification": "Notification", // Topic notice
                "Introduction": "Introduction" // Topic introduction
            }
        }
    ]
}

```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed
ErrorInfo	String	Error information
TopicAndSelfInfo	Array	The returned result is a topic information array, which contains the basic topic information and the information of the specified user in the topic.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10006	The operation exceeds the frequency limit. Please reduce the call frequency. This error is usually caused by too much net group increase in a single day or too frequent calls to obtain all groups in the app.
10007	Insufficient operation permissions. Check the request parameters based on the error message.
10008	The request is invalid, probably because verification of the signing information in the request failed. Please try again or <a href="#">submit a ticket</a> .
10010	Invalid request. The group or topic has been deleted.
11000	The current group does not support the community topic feature. To use this feature, you need to purchase the <a href="#">Pro edition</a> 、 <a href="#">Pro Plus edition</a> or <a href="#">Enterprise edition</a> and <a href="#">enable it in the console</a> .
110003	Request failed. The topic has been deleted.



# API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Modifying Topic Profile

Last updated : 2025-01-14 14:46:02

## Feature Overview

This API is used to modify a topic profile.

## API Calling Description

### Applicable group types

Group Type ID	Whether This RESTful API Is Supported
Private	No
Public	No
ChatRoom	No
AVChatRoom	No
Community	This API applies only to topic-enabled communities.

These are the preset group types in Chat. For more information, see [Group System](#).

### Note:

To use the topic feature, you need to go to the [console](#), choose **Feature Configuration > Group configuration > Group feature configuration > Community**, enable the community feature and then enable the topic feature.

### Sample request URL

```
https://xxxxxx/v4/million_group_open_http_svc/modify_topic?sdkappid=88888888&identi
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your

	SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/million_group_open_http_svc/modify_topic	Request API
sdkappid	The SDKAppID assigned by the Chat console when the application is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. Fixed value: json .

## Maximum call frequency

200 calls per second

## Sample request

### Modify the basic topic information

Modify the basic topic information, such as topic name and topic notice.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group of the topic to be modified, which i
  "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#crTE3HIM62C5", // ID of the topic
  "TopicName": "TestTopicName", // Topic name, which is optional
  "Notification": "Notification" // Topic notice, which is optional
}
```

### ALL IN ONE

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group of the topic to be modified, which
  "TopicId": "@TGS#_@TGS#cQVLVHIM62CJ@TOPIC#_@TOPIC#crTE3HIM62C5", // ID of th
  "From_Account": "1400187352", // Member modifying the topic
  "TopicName": "TestTopicName", // Topic name, which is optional
}
```

```
"FaceUrl": "http://this.is.new.face.url", // Topic profile photo, which is option
"Notification": "Notification", // Topic notice, which is optional
"Introduction": "Introduction", // Topic introduction, which is optional
"MuteAllMember": "On", // Mutes all members, which is optional. `On`: Enable. `Of
"CustomString": "This is a customs string." // Custom string, which is optional
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Required	Group ID of the topic to be modified
TopicId	String	Required	ID of the topic to be modified
TopicName	String	Optional	Topic name, whose maximum length is 30 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
From_Account	uint64	Optional	User account that wants to modify the topic
CustomString	String	Optional	A custom string, which can contain up to 3,000 bytes. encoded in UTF-8.
FaceUrl	String	Optional	Profile photo URL of the topic, which can contain up to 100 bytes.
Notification	String	No	Topic notice, whose maximum length is 300 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
Introduction	String	No	Topic introduction, whose maximum length is 240 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
MuteAllMember	String	Optional	Mutes all members in a topic and allows only the group admin, group owner, and system admin to speak.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "ok",
  "ErrorCode": 0
}
```

Response fields

--	--	--

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10006	The operation exceeds the frequency limit. Please reduce the call frequency. This error is usually caused by too much net group increase in a single day or too frequent calls to obtain all groups in the app.
10007	Insufficient operation permissions. Check the request parameters based on the error message.
10008	The request is invalid, probably because verification of the signing information in the request failed. Please try again or <a href="#">submit a ticket</a> .
10010	Invalid request. The topic has been deleted.
10015	The requested group ID is invalid. Check the request parameter based on the error message.
10016	The app backend rejected this operation through a third-party callback. Check the returned value of your webhook "Topic Profile Change Webhook".
11000	The current group does not support the community topic feature. To use this feature, you need to purchase the <a href="#">Pro edition</a> 、 <a href="#">Pro Plus edition</a> or <a href="#">Enterprise edition</a> and <a href="#">enable it in the console</a> .
80001	Failed to pass the security check. Check the request parameters based on the error message.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Possible Webhooks

[Topic Profile Change Webhook](#)

# Importing Topic Profiles

Last updated : 2025-01-14 14:46:02

## Feature Overview

This API allows the app admin to import topic data without triggering callbacks or delivering notifications. When your app needs to be migrated to Chat from another instant messaging system, you can use this API to import existing topic data.

## API Calling Description

### Applicable group types

Group Type	Support for This RESTful API
Private	No
Public	No
ChatRoom	No
AVChatRoom	No
Community	This API applies only to topic-enabled communities.

These are the preset group types in Chat. For more information, see [Group System](#).

### Note:

To use the topic feature, you need to go to the [console](#), choose **Feature Configuration > Group configuration > Group feature configuration > Community**, enable the community feature and then enable the topic feature.

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/import_topic?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

--	--

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/import_topic	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Specify the group to which the topic to import belongs. You can use `CreateTime` to specify the topic creation time.

```
{
  "GroupId": "@TGS#_@TGS#CBZXAIIM62CV",      // (Required) ID of the group to
  which the topic to import belongs
  "TopicName": "test_topic3",                // (Optional) Topic name
  "CreateTime": 1448357837                    // (Optional) Topic creation time
}
```

### Specifying other optional fields

Specify optional fields such as `Introduction` and `Notification`.

```
{
  "Type": "Community", // (Optional) Type of the group to which the topic
  belongs, which should be `Community`
}
```



```
"GroupId": "@TGS#_@TGS#cBZXAIIM62CV", // (Required) ID of the group to
which the topic to import belongs
"TopicName": "test_topic3", // (Optional) Topic name
"Introduction": "This is topic Introduction", // (Optional) Topic
introduction
"Notification": "This is topic Notification", // (Optional) Topic
notification
"FaceUrl": "http://this.is.face.url",
"CreateTime": 1448357837 // (Optional) Topic creation time. If this field is
not specified, the default creation time is the request time.
}
```

## Request fields

Field	Type	Required	Description
Type	String	No	Type of the group to which the topic belongs. Currently, only a Community group is supported.
GroupId	String	Yes	ID of group to which the topic to import belongs.
TopicId	String	No	To simplify topic IDs and make them easy to remember, Tencent Cloud allows apps to customize topic IDs during topic creation through RESTful APIs. For details, see <a href="#">here</a> .
TopicName	String	Yes	Topic name, whose maximum length is 30 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
From_Account	uint64	No	User account that wants to create the topic.
CustomString	String	No	Custom string, which can contain up to 3,000 bytes, encoded in UTF-8.
FaceUrl	String	No	URL of the topic profile photo, whose maximum length is 100 bytes.
Notification	String	No	Topic notice, whose maximum length is 300 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
Introduction	String	No	Topic introduction, whose maximum length is 240 bytes. This field is UTF-8-encoded, and one Chinese character occupies three bytes.
CreateTime	Integer	No	Topic creation time

## Sample response

### Basic format

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicId": "@TGS#_@TGS#cBZXAIIM62CV@TOPIC#_@TOPIC#cTCCCIIM62CW"
}
```

### Specifying other optional fields

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicId": "@TGS#_@TGS#cBZXAIIM62CV@TOPIC#_@TOPIC#c5CCCIIM62CW"
}
```

### Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
TopicId	String	Topic ID after successful creation, which is assigned by the Chat backend or specified by users.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10003	Invalid command word.
10004	Invalid parameter. Check the error description and troubleshoot the issue.

10007	Insufficient operation permissions. Check the request parameters based on the error message.
10010	The current group does not exist or has been deleted.
10015	The requested group ID is invalid. Check the request parameter based on the error message.
10021	The topic ID is already in use. Specify another topic ID.
10037	The number of prepaid topics created exceeds the limit. Delete some of the topics or upgrade your service. For more information, see <a href="#">Pricing</a> .
11000	The current group does not support the community topic feature. To use this feature, you need to purchase the <a href="#">Pro edition</a> 、 <a href="#">Pro Plus edition</a> or <a href="#">Enterprise edition</a> and <a href="#">enable it in the console</a> .
80001	Failed to pass the security check. Check the request parameters based on the error message.
80005	Failed to pass the security check: Security check timed out.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Creating a topic ([v4/million\\_group\\_open\\_http\\_svc/create\\_topic](#))

Deleting a topic ([v4/million\\_group\\_open\\_http\\_svc/destroy\\_topic](#))

Importing a group profile ([v4/group\\_open\\_http\\_svc/import\\_group](#))

# Permission Group Management

## Creating Permission Groups

Last updated : 2024-07-16 11:20:06

### Feature Overview

App admins can call this API to create a new permission group.

### API Calling Description

#### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

#### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/create_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description

xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/group_open_http_svc/create_permission_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: json

Maximum Calling Frequency

200 queries/sec.

Sample Request Packet

Basic Form

```
{
  "GroupId": "@TGS#_@TGS#cV6IHIM62C4",    // Group ID belonging to the
permission group
  "PermissionGroupId": "@PMG#_test_permission_group", // Permission group ID
  "PermissionGroupName": "test_permission_group", // Permission group name
  "Permission": 123, // Specific permission
  "CustomString": "test_custom_string" // Custom field
}
```

Description of Request Packet Fields

Field	Type	Attribute	Description

GroupId	String	Required	Group ID
PermissionGroupId	String	Optional	Permission group ID. To simplify the permission group ID and make it easier to remember and disseminate, Tencent Cloud allows you to customize permission group IDs in the app when creating a permission group through the RESTful API <a href="#">Customizing Permission Group IDs</a> . If this parameter is not specified, the system will generate a default permission group ID and return it.
PermissionGroupName	String	Required	The name of the permission group. It can contain up to 150 UTF-8 encoded bytes, and 1 Chinese character occupies 3 bytes.
Permission	Integer	Required	The specific permissions associated with the permission group, where each bit represents a type of permission. For details on the permission bits, refer to <a href="#">Permission Bits</a> .
CustomString	String	Optional	The custom fields of the permission group. It can contain up to 3000 bytes. The business layer can use this field to meet the needs of special scenarios.

Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "PermissionGroupId": "@PMG#_test_permission_group"
}
```

Response packet field description

Field	Type	Description
ActionStatus	String	Result of the request processing: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
PermissionGroupId	String	Returned permission group ID

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by `ErrorCode` and `ErrorInfo` in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to create a permission group
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
10019	The requested UserID does not exist. Check if all Member_Accounts in the MemberList are correct.
10021	The created permission group ID has been used, indicating a possible duplication in creating permission groups.
10037	The number of permission groups created exceeds the limit.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Terminating Permission Groups ([Terminating Permission Groups](#))

Modifying Permission Group Information ([Modifying Permission Group Information](#))

Obtaining Permission Group Information ([Obtaining Permission Group Information](#))



# Terminating Permission Groups

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to terminate created permission groups.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/destroy_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/destroy_permission_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

### Basic Form

```
{
  "GroupId": "@TGS#_@TGS#cv6IHIIM62C4", // Group ID
  "PermissionGroupIdList": [ // List of permission group IDs to be
    terminated. Up to 20 groups can be terminated at a time.
    "@PMG#_@PMG#cM2",
    "@PMG#_@PMG#cN3"
  ]
}
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the group to operate on
PermissionGroupIdList	Array	Required	Permission group ID list. Up to 20 permission groups can be

			terminated at a time.
--	--	--	-----------------------

Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "PermissionGroupResultList": [    // Returned results
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "PermissionGroupId": "@PMG#_@PMG#cM2"
    },
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "PermissionGroupId": "@PMG#_@PMG#cN3"
    }
  ]
}
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
PermissionGroupResultList	Array	Returned results of permission group termination. Each item represents the execution result of terminating a permission group, with only ErrorCode 0 indicating successful termination.

Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by `ErrorCode` and `ErrorInfo` in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to create a permission group
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
11000	The requested SDKAppID does not support communities or communities with topics. Apply for a qualified SDKAppID, configure it, and then make a request again.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110007	Failed to dissolve the permission group.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Creating Permission Groups ([Creating Permission Groups](#))

Modifying Permission Group Information ([Modifying Permission Group Information](#))

Obtaining Permission Group Information ([Obtaining Permission Group Information](#))



# Modifying Permission Group Information

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to modify information related to the permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/modify_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_permission_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

```
{
  "GroupId": "@TGS#_@TGS#cv6IHIIM62C4",
  "PermissionGroupId": "@PMG#_test_permission_group",
  "Permission": 123,
  "PermissionGroupName": "test_modify_group_name",
  "CustomString": "test_modify_custom_string",
}
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the group to operate on
PermissionGroupId	String	Required	Permission group ID
PermissionGroupName	String	Optional	The name of the permission group. It can contain up to 150

			UTF-8 encoded bytes, and 1 Chinese character occupies 3 bytes.
Permission	Integer	Optional	Specific permissions associated with the permission group
CustomString	String	Optional	The custom fields of the permission group. It can contain up to 3000 bytes. The business layer can use this field to meet the needs of special scenarios.

Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message

Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.



10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to create a permission group
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
11000	The requested SDKAppID does not support communities or communities with topics. Apply for a qualified SDKAppID, configure it, and then make a request again.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Creating Permission Groups ([Creating Permission Groups](#))

Terminating Permission Groups ([Terminating Permission Groups](#))

Obtaining Permission Group Information ([Obtaining Permission Group Information](#))

# Obtaining Permission Group Information

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to obtain relevant information about the permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/get_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_permission_group</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
<code>usersig</code>	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295
<code>contenttype</code>	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

### Obtaining All Permission Groups

```
{
  "GroupId": "@TGS#_@TGS#cV6IHIIM62C4",
  "Limit": 20,
  "Next": ""
}
```

### Obtaining Specified Permission Group

```
{
  "GroupId": "@TGS#_@TGS#cV6IHIIM62C4",
  "PermissionGroupIdList": [
    "@PMG#_test_permission_group_with_topic",
    "@PMG#_@PMG#cA7VPIIM62CJ"
  ]
}
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the group to operate on
PermissionGroupIdList	String	Optional	The list of permission groups to be obtained. If this field is empty, it means all permission groups are obtained.
Limit	Integer	Optional	How many permission groups' information can be obtained at one time. It cannot exceed 20 and is used for pagination to obtain all permission groups.
Next	Integer	Optional	The position of the last permission group obtained. Fill in "" for the first call, and fill in the Next Value returned in the response for subsequent pulling.

## Sample Response Packet

### Obtaining All Permission Groups

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "PermissionGroupInfoList": [    // The permission group information
    returned. Each item represents one permission group.
    {
      "ErrorCode": 0,    // The result of obtaining this permission group
      "ErrorInfo": "",
      "PermissionGroupId": "@PMG#_@PMG#c04",
      "PermissionGroupName": "test_permission_group",
      "CustomString": "test_custom_string",
      "Permission": 123,
      "MemberCount": 0    // Number of members in the permission group
    },
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "PermissionGroupId": "@PMG#_@PMG#cP5",
      "PermissionGroupName": "test_permission_group",
      "CustomString": "test_custom_string",
      "Permission": 123,
      "MemberCount": 0
    },
    {
      "ErrorCode": 0,
```

```
        "ErrorInfo": "",
        "PermissionGroupId": "@PMG#_@PMG#cQ6",
        "PermissionGroupName": "test_permission_group",
        "CustomString": "test_custom_string",
        "Permission": 123,
        "MemberCount": 0
    },
    // The following omits some permission group information.
],
"Next": ""
}
```

Obtaining Specified Permission Group

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "PermissionGroupInfoList": [
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "PermissionGroupId": "@PMG#_test_permission_group_with_topic",
      "PermissionGroupName": "test_permission_group",
      "CustomString": "test_custom_string",
      "Permission": 123,
      "MemberCount": 1
    },
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "PermissionGroupId": "@PMG#_@PMG#cP5",
      "PermissionGroupName": "test_permission_group",
      "CustomString": "test_custom_string",
      "Permission": 123,
      "MemberCount": 0
    }
  ],
  "Next": ""
}
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing

		OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
PermissionGroupInfoList	Array	The obtained permission group data list (note that each permission group data includes ErrorCode and ErrorInfo. Only the data with ErrorCode as 0 is meaningful). For field details, refer to <a href="#">Permission Group Information</a> .

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to create a permission group
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
11000	The requested SDKAppID does not support communities or communities with topics. Apply for a qualified SDKAppID, configure it, and then make a request again.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.

# API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Creating Permission Groups ([Creating Permission Groups](#))

Terminating Permission Groups ([Terminating Permission Groups](#))

Modifying Permission Group Information ([Modifying Permission Group Information](#))

# Adding Topic Permissions

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to add topic permissions to a permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/add_topic_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:



	China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/group_open_http_svc/add_topic_permission_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: json

### Maximum Calling Frequency

200 queries/sec.

### Sample Request Packet

#### Basic Form

It is used to add topic permissions to a permission group.

```
{
  "GroupId": "@TGS#_@TGS#cv6IHIIM62C4",
  "PermissionGroupId": "@PMG#_test_permission_group_with_topic",
  "TopicPermissionList": [
    {
      "Permission": 1234,
      "TopicId": "@TGS#_@TGS#cv6IHIIM62C4@TOPIC#_@TOPIC#cKXPQIIM62CD"
    }
  ]
}
```

### Description of Request Packet Fields

Field	Type	Attribute	Description
-------	------	-----------	-------------

GroupId	String	Required	Group ID
PermissionGroupId	String	Required	Permission group ID
TopicPermissionList	Array	Required	The topic permissions added cannot exceed 50 in one request.
TopicId	String	Required	Added topic ID
Permission	Integer	Required	Similar to the permission bits of a permission group, each bit of topic permissions represents a specific permission. For detailed descriptions, refer to <a href="#">Topic Permission Bits</a> .

Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicPermissionResultList": [
    {
      "ErrorCode": 0, // Returned results
      "ErrorInfo": "",
      "TopicId": "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cKXPQIIM62CD"
    }
  ]
}
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
TopicPermissionResultList	Array	Addition results of topic permissions. Each item represents a result. Only an ErrorCode of 0 indicates a successful addition.

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by `ErrorCode` and `ErrorInfo` in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to create a permission group
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.
110010	The permission group or topic permission already exists.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Creating Permission Groups ([Creating Permission Groups](#))

# Modifying Topic Permissions

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to modify the topic permissions in a permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/set_topic_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/set_topic_permission_group	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

Modify topic permissions in a permission group

```
{
  "GroupId": "@TGS#_@TGS#cv6IHIIM62C4",
  "PermissionGroupId": "@PMG#_test_permission_group_with_topic",
  "TopicPermissionList": [
    {
      "Permission": 123456,
      "TopicId": "@TGS#_@TGS#cv6IHIIM62C4@TOPIC#_@TOPIC#cTEJHIIM62CJ"
    }
  ]
}
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	Group ID

PermissionGroupId	String	Required	Permission group ID
TopicPermissionList	Array	Required	The topic permissions added cannot exceed 50 in one request.
TopicId	String	Required	Added topic ID
Permission	Integer	Required	Similar to the permission bits of a permission group, each bit of topic permissions represents a specific permission. For specific permission bit descriptions, refer to <a href="#">Introduction to Topic Permission Bits</a> .

Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicPermissionResultList": [
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "TopicId": "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cTEJHIIM62CJ"
    }
  ]
}
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
TopicPermissionResultList	Array	The result of the added topic permissions. Each item represents a result. Only when the ErrorCode is 0, it indicates a successful modification.

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by `ErrorCode` and `ErrorInfo` in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to delete permission group members
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.
110011	Topic permissions do not exist.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Adding Topic Permissions ([Adding Topic Permissions](#))

Deleting Topic Permissions ([Deleting Topic Permissions](#))

Obtaining Topic Permissions ([Obtaining Topic Permissions](#))

# Deleting Topic Permissions

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to delete topic permissions in a permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/delete_topic_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:



	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/delete_topic_permission_group</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
<code>usersig</code>	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295
<code>contenttype</code>	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

Delete topic permissions in a permission group

```
{
  "GroupId": "@TGS#_@TGS#cV6IHIIM62C4",
  "PermissionGroupId": "@PMG#_test_permission_group_with_topic",
  "TopicIdList": [
    "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cKXPQIIM62CD"
  ]
}
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	Group ID
PermissionGroupId	String	Required	Permission group ID

TopicIdList	Array	Required	List of topic IDs to be deleted. A maximum of 50 topics can be deleted per request.
-------------	-------	----------	---

Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicPermissionResultList": [
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "TopicId": "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cKXPQIIM62CD"
    }
  ]
}
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
TopicPermissionResultList	Array	Result of the deleted topic permissions. Each item represents a result. Only when the ErrorCode is 0, it indicates a successful modification.

Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

--	--

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to delete permission group members
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.
110011	Topic permissions do not exist.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Adding Topic Permissions ([Adding Topic Permissions](#))

Modifying Topic Permissions ([Modifying Topic Permissions](#))

# Obtaining Topic Permissions

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to obtain topic permissions in a permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/get_topic_permission_group?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_topic_permission_group</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
<code>usersig</code>	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295
<code>contenttype</code>	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

### Obtaining All Topic Permissions

```
{
  "GroupId": "@TGS#_@TGS#cV6IHIIM62C4",
  "PermissionGroupId": "@PMG#_test_permission_group_with_topic"
}
```

### Obtaining Specified Permission Group

```
{
  "GroupId": "@TGS#_@TGS#cV6IHIIM62C4",
  "PermissionGroupId": "@PMG#_test_permission_group_with_topic",
  "TopicIdList": [
    "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cTEJHIIM62CJ"
  ]
}
```

Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	Group ID
PermissionGroupId	String	Required	Permission group ID
TopicIdList	Array	Required	The list of topic permissions to be retrieved. If this parameter is not specified, all topic permissions will be returned.

Sample Response Packet

Obtaining All Permission Groups

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicPermissionResultList": [
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "TopicId": "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cKXPQIIM62CD",
      "Permission": 1234
    },
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "TopicId": "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cTEJHIIM62CJ",
      "Permission": 123456
    }
  ]
}
```

Obtaining Specified Permission Group

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TopicPermissionResultList": [
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "TopicId": "@TGS#_@TGS#cV6IHIIM62C4@TOPIC#_@TOPIC#cTEJHIIM62CJ",
      "Permission": 123456
    }
  ]
}
```

```
    }
  ]
}
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
TopicPermissionResultList	Array	The results of obtained topic permissions. Each item represents a result. Only the topic permissions with an ErrorCode of 0 are meaningful.

Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to create a permission group
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.

110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.
110014	Topic permissions deleted

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Adding Topic Permissions ([Adding Topic Permissions](#))

Modifying Topic Permissions ([Modifying Topic Permissions](#))

Deleting Topic Permissions ([Deleting Topic Permissions](#))



# Adding Members to a Permission Group

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to add new members to a specified permission group.

## API Calling Description

### Applicable Group Types

Group Type	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned four types of groups. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/add_permission_group_member?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/add_permission_group_member</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an application is created
<code>identifier</code>	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
<code>usersig</code>	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295
<code>contenttype</code>	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

### Basic Form

It is used to add members to the permission group. A maximum of 100 members can be added per request.

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group to operate on (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group to operate on
  (required)
  "MemberList": [ // A maximum of 100 members can be added at once.
    {
      "Member_Account": "tommy" // Permission group member ID to be added
    },
    {
      "Member_Account": "jared"
    }
  ]
}
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the group to operate on
PermissionGroupId	String	Required	ID of the permission group to operate on
MemberList	Array	Required	Array of permission group members to be added
Member_Account	String	Required	UserID of the permission group member to be added

## Sample Response Packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "MemberList": [
    {
      "Member_Account": "tommy",
      "Result": 0 // Addition result: 0 indicates success, and 10013
indicates it is already a group member. For other results, refer to the error
code description.
    },
    {
      "Member_Account": "jared",
      "Result": 10013
    }
  ]
}
```

## Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
MemberList	Array	Returned result of added group members

Member_Account	String	Returned group member UserID
Result	Integer	Addition result: 0: success 10013: Already a group member For other results, refer to the error codes.

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error, Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to add members to the permission group
10013	Already a permission group member
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
10019	The requested UserID does not exist. Check if all Member_Accounts in the MemberList are correct.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.
110012	The number of permission group members exceeds the limit.

## API Debugging Tool

---

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Deleting Permission Group Members ([Deleting Permission Group Members](#))

# Deleting Permission Group Members

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to delete members from permission groups.

## API Calling Description

### Applicable Group Types

Group Type ID	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned group types. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/delete_permission_group_member?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/delete_permission_group_member</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an application is created
<code>identifier</code>	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
<code>usersig</code>	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295
<code>contenttype</code>	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

### Basic Form

It is used to delete members from the permission group. A maximum of 100 members can be deleted per request. By default, a system notification for the deletion of group members is sent to all members in the group.

If the user to be deleted is not in the permission group, the API still returns success.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ", // Group to operate on (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group to operate on (required)
  "MemberList": [ // Up to 100 members can be deleted at once.
    {
      "Member_Account": "tommy" // ID of the permission group member to be deleted
    },
    {
      "Member_Account": "jared"
    }
  ]
}
```

```
    }]  
  }
```

## Description of Request Packet Fields

Field	Type	Attribute	Description
GroupId	String	Required	ID of the group to operate on
PermissionGroupId	String	Required	ID of the permission group to operate on
MemberList	Array	Required	Array of permission group members to be deleted
Member_Account	String	Required	UserID of the permission group member to be deleted

## Sample Response Packet

```
{  
  "ActionStatus": "OK",  
  "ErrorInfo": "",  
  "ErrorCode": 0,  
  "MemberList": [  
    {  
      "Member_Account": "tommy",  
      "Result": 0 // Deletion result: 0 indicates success. For other  
results, refer to the error code description.  
    },  
    {  
      "Member_Account": "jared",  
      "Result": 10013  
    }  
  ]  
}
```

## Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.



ErrorInfo	String	Error message
MemberList	Array	Returned result of deleted group members
Member_Account	String	Returned group member UserID
Result	Integer	Addition result: 0: success For other results, refer to the error codes.

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions, such as lacking the permission to delete permission group members
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
10016	The developer backend has denied this operation through third-party callback.
10019	The requested UserID does not exist. Check if all Member_Accounts in the MemberList are correct.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

## References

Adding Members to a Permission Group ([Adding Members to a Permission Group](#))

# Obtaining Permission Group Member List

Last updated : 2024-07-16 11:20:06

## Feature Overview

App admins can call this API to obtain group member information based on the group ID.

## API Calling Description

### Applicable Group Types

Group Type ID	Whether This RESTful API Is Supported
Private	Not supported, same as Work (Work Friend Group) in the new version
Public	Not supported
ChatRoom	Not supported, same as Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Not supported
Community	Supported

Chat provides the aforementioned group types. For details, refer to [Group System](#).

### Sample Request URL

```
https://xxxxxx/v4/group_open_http_svc/get_permission_group_member_list?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The table below only lists the parameters modified when calling this API and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name for the country/region where your SDKAppID is located:

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
<code>v4/group_open_http_svc/get_permission_group_member_list</code>	Request API
<code>sdkappid</code>	SDKAppID assigned by the Chat console when an app is created
<code>identifier</code>	You must be an app admin account. For details, refer to <a href="#">App Admins</a> .
<code>usersig</code>	The signature generated by the app admin account. For details, refer to <a href="#">Generating UserSig</a> .
<code>random</code>	A random 32-bit unsigned integer ranging from 0 to 4294967295
<code>contenttype</code>	Request format fixed value: <code>json</code>

## Maximum Calling Frequency

200 queries/sec.

## Sample Request Packet

### Basic Form

It is used to obtain detailed information of permission group members (including group member information and group member dimensions from the Definition field). The request only contains the group ID and permission group ID.

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group ID (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group ID (required)
}
```

### Batch Acquisition

Use Limit and Next values to control pagination retrieval:

Limit restricts the number of members in the MemberList array in the response packet, which cannot exceed 50.

Next controls the retrieval of subsequent information from a certain member's position. In the first request, the client's request parameter for Next shall be ""; when the server's return for Next is "" after the last request, it indicates the end of retrieval.

For requests in between, the client's Next shall use the Next returned by the server from the previous request. It is similar to the scan cursor query of Redis.

For example: Suppose you need to batch pull, the request parameters for the first time shall be: `{"Limit": 20, "Next": ""}`, and the server returns the following content.

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "TGSMTQ0MTE1MjM1MTUyNDIOMzY1",
  "MemberList": [
    ....
  ]
}
```

The request parameters for the second time shall be `{"Limit": 20, "Next":`

`"TGSMTQ0MTE1MjM1MTUyNDIOMzY1"}`.

And so on, until the server's response packet returns Next as "", indicating there is no subsequent member information, and the client shall end the query.

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group ID (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group ID (required)
  "Limit": 50, // The maximum number of members whose information to be
  obtained
  "Next": "" // Start pulling from the end of the last pull
}
```

### Specifying the Information to Be Pulled

Select the fields to be pulled through the MemberInfoFilter filter. Fields not specified in the filter will not be pulled.

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group ID (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group ID (required)
  "MemberInfoFilter": [ // Which information is needed (Member_Account is
  included by default). If this field is not specified, then information of all
  members is included.
    "Role",
    "JoinTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "JoinPermissionGroupTime",
    "MuteUntil",
  ]
}
```

```
        "NameCard"
    ]
}
```

### Pulling Custom Field for Group Members

Select the custom field for group member to be pulled through the AppDefinedDataFilter\_GroupMember filter. Fields not specified in the filter will not be pulled.

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group ID (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group ID (required)
  "AppDefinedDataFilter_GroupMember": [ // Filter for custom field of group
members
    "MemberDefined2" // Key of the custom field for group members
  ]
}
```

### ALL IN ONE

```
{
  "GroupId": "@TGS#_@TGS#cAVQXXXXXX", // Group ID (required)
  "PermissionGroupId": "@PMG#_@PMG#cDR", // Permission group ID (required)
  "MemberInfoFilter": [ // Which information is needed. If this field is not
specified, then information of all members is included.
    "Role",
    "JoinTime",
    "JoinPermissionGroupTime",
    "MsgSeq",
    "MsgFlag",
    "LastSendMsgTime",
    "MuteUntil",
    "NameCard"
  ],
  "AppDefinedDataFilter_GroupMember": [ // Filter for custom field of group
members
    "MemberDefined2", // Key of the custom field for group members
    "MemberDefined1"
  ],
  "Limit": 50, // The maximum number of members whose information to be
obtained
  "Offset": 0 // The starting point to obtain member information
}
```

### Description of Request Packet Fields

Field	Type	Attribute	Description
-------	------	-----------	-------------

GroupId	String	Required	ID of the group from which member information needs to be obtained
PermissionGroupId	String	Required	ID of the permission group to operate on
MemberInfoFilter	Array	Optional	Which information is needed. If this field is not specified, then information of all members is included. For details on member information fields, refer to <a href="#">Group Member Information</a> .
AppDefinedDataFilter_GroupMember	Array	Optional	By default, this is not available. This field is used as a filter for custom fields in the group member dimension, specifying the custom fields of the group member dimension that need to be obtained For details, refer to <a href="#">Custom Fields</a> .
Next	String	Optional	The member location obtained last time. It is required for communities, as communities do not support the Offset parameter. Use the Next parameter instead. Fill in "" for the first call, and use the Next value returned for subsequent retrievals.

## Sample response packets

### Basic Form and Pagination Retrieval

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "TGSMTQ0MTE1MjM1MTUyNDI0MzY1",
  "MemberNum": 2, // Total number of members in this permission group
  "MemberList": [ // Permission group member list
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // Join time
      "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
      "MsgSeq": 1233,
      "MsgFlag": "AcceptAndNotify",
      "LastSendMsgTime": 1425976500, // Last message sent time
      "MuteUntil": 1431069882, // Mute until (seconds)
      "AppMemberDefinedData": [ // Group member's custom definition field
```

```

        {
            "Key": "MemberDefined1",
            "Value": "ModifyDefined1"
        },
        {
            "Key": "MemberDefined2",
            "Value": "ModifyDefined2"
        }
    ]
},
{
    "Member_Account": "peter",
    "Role": "Member ",
    "JoinTime": 1425976500, // Join time
    "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
    "MsgSeq": 1233,
    "MsgFlag": "AcceptAndNotify",
    "LastSendMsgTime": 1425976500,
    "MuteUntil": 0, // 0 indicates not muted; otherwise, it's the mute
expiry timestamp.
    "AppMemberDefinedData": [ // Group member's custom definition field
        {
            "Key": "MemberDefined1",
            "Value": "ModifyDefined1"
        },
        {
            "Key": "MemberDefined2",
            "Value": "ModifyDefined2"
        }
    ]
}
]
}

```

### Pulling Specified Fields

```

{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "Next": "TGSMTQ0MTE1MjM1MTUyNDI0MzY1",
    "MemberNum": 2, // Total number of members in this permission group
    "MemberList": [ // Permission group member list
        {
            "Member_Account": "bob",
            "Role": "Owner",

```



```

    "JoinTime": 1425976500, // Join time
    "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
    "MsgSeq": 1233,
    "MsgFlag": "AcceptAndNotify",
    "LastSendMsgTime": 1425976500, // Last message sent time
    "MuteUntil": 1431069882, // Mute until (seconds)
  },
  {
    "Member_Account": "peter",
    "Role": "Member ",
    "JoinTime": 1425976500, // Join time
    "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
    "MsgSeq": 1233,
    "MsgFlag": "AcceptAndNotify",
    "LastSendMsgTime": 1425976500,
    "MuteUntil": 0, // 0 indicates not muted; otherwise, it's the mute
expiry timestamp.
  }
]
}

```

### Pulling Custom Field for Group Members

```

{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "TGSMTQ0MTE1MjM1MTUyNDI0MzY1",
  "MemberNum": 2, // Total number of members in this permission group
  "MemberList": [ // Permission group member list
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // Join time
      "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
      "MsgSeq": 1233,
      "MsgFlag": "AcceptAndNotify",
      "LastSendMsgTime": 1425976500, // Last message sent time
      "MuteUntil": 1431069882, // Mute until (seconds)
      "AppMemberDefinedData": [ // Custom fields for group members
        {
          "Key": "MemberDefined2",
          "Value": "ModifyDefined2"
        }
      ]
    }
  ]
}

```

```

    ]
  },
  {
    "Member_Account": "peter",
    "Role": "Member",
    "JoinTime": 1425976500, // Join time
    "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
    "MsgSeq": 1233,
    "MsgFlag": "AcceptAndNotify",
    "LastSendMsgTime": 1425976500,
    "MuteUntil": 0, // 0 indicates not muted; otherwise, it's the mute
expiry timestamp.
    "AppMemberDefinedData": [ // Group member's custom definition field
      {
        "Key": "MemberDefined2",
        "Value": "ModifyDefined2"
      }
    ]
  }
]
}

```

## ALL IN ONE

```

{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "TGSMTQ0MTE1MjM1MTUyNDI0MzY1",
  "MemberNum": 2, // Total number of members in this permission group
  "MemberList": [ // Permission group member list
    {
      "Member_Account": "bob",
      "Role": "Owner",
      "JoinTime": 1425976500, // Join time
      "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
      "MsgSeq": 1233,
      "MsgFlag": "AcceptAndNotify",
      "LastSendMsgTime": 1425976500, // Last message sent time
      "MuteUntil": 1431069882, // Mute until (seconds)
      "AppMemberDefinedData": [ // Custom fields for group members
        {
          "Key": "MemberDefined1",
          "Value": "ModifyDefined1"
        }
      ],

```

```
        {
            "Key": "MemberDefined2",
            "Value": "ModifyDefined2"
        }
    ]
},
{
    "Member_Account": "peter",
    "Role": "Member",
    "JoinTime": 1425976500, // Join time
    "JoinPermissionGroupTime": 1704804868, // Time joined the permission
group
    "MsgSeq": 1233,
    "MsgFlag": "AcceptAndNotify",
    "LastSendMsgTime": 1425976500,
    "MuteUntil": 0, // 0 indicates not muted; otherwise, it's the mute
expiry timestamp.
    "AppMemberDefinedData": [ // Group member's custom definition field
        {
            "Key": "MemberDefined1",
            "Value": "ModifyDefined1"
        },
        {
            "Key": "MemberDefined2",
            "Value": "ModifyDefined2"
        }
    ]
}
]
```

Description of Response Packet Fields

Field	Type	Description
ActionStatus	String	Result of the request processing: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code 0: Indicates success. Non-zero: Indicates failure.
ErrorInfo	String	Error message
MemberNum	Integer	Total number of members in this permission group

MemberList	Array	Obtained permission group members list, which includes all or specified group member information. For details on member information fields, refer to <a href="#">Group Member Information</a> .
AppMemberDefinedData	Array	Returned group member custom field information
Next	String	Next value that shall be passed in the next request

## Error Codes

Unless a network error (such as a 502 error) occurs, the HTTP return code for this API is always 200. The real error code and error message are indicated by ErrorCode and ErrorInfo in the response packet body.

Common error codes (60000 to 79999) can be found in the [Error Codes](#) document.

The private error codes for this API are as follows:

Error Code	Description
10002	Internal server error. Try again.
10003	Illegal request command word
10004	Invalid parameter. Check if the request is correct based on the error description.
10007	Insufficient operation permissions. Confirm whether the operator is an app admin or has permission to read the fields in the request.
10010	The group does not exist, or it existed in the past but has now been dissolved.
10015	Invalid group ID. Check if the group ID is filled in correctly.
110006	The permission group does not exist, or it existed in the past but has now been dissolved.
110008	Invalid permission group ID. Check if the permission group ID is filled in correctly.

## API Debugging Tool

Use the [RESTful API Online Debugging Tool](#) to debug this API.

# Getting the List of Online Members in Community Group

Last updated : 2024-12-09 15:30:31

## Feature Overview

The app admin can retrieve the number of online users in the community according to the Group ID.

## API Calling Description

### Applicable Group Types

Group Type ID	Is this REST API supported
Private	Supported, same as the Work (Friends Work Group) in the new version
Public	Supported
ChatRoom	Supported, same as the Meeting (Temporary Meeting Group) in the new version
AVChatRoom	Supported
Community (Community)	Supported

Chat has built-in support for the above group type. For details, please refer to [Group System](#).

### Note

The overall update granularity for the number of online users in a Community is 4 minutes.  
Due to the large number of people in a Community, the overall granularity for updating the number of online users is 4 minutes. When group members go online or offline, their status will be updated in the next 4-minute cycle because their online status has already been counted in the current cycle.

### Sample request URL

```
https://xxxxxx/v4/million_group_open_http_svc/get_online_member_num?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

## Request parameters

The table below only lists the parameters that are modified when calling this interface and their descriptions. For more details on other parameters, please refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain for the country/region where your SDKAppID is located: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley : <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/million_group_open_http_svc/get_online_member_num	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	Must be an App Administrator Account. For more details, see <a href="#">App Administrator</a>
usersig	The signature generated by the App Administrator Account. For details on the procedure, see <a href="#">Generate UserSig</a>
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	The request format is a fixed value of <code>json</code>

## Maximum calling frequency

200 queries/sec.

## Sample request packets

This API is used to get the number of online users in a Community.

```
{
  "GroupId": "@TGS#_@TGS#cQVLVHIM62CJ"
}
```

## Request packet fields

Field	Type	Attribute	Description
GroupId	String	Required	Group ID to operate

Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "OnlineMemberNum":1000 // Number of online members
}
```

Response packet fields description

Field	Type	Description
ActionStatus	String	Result of request processing: OK: Indicates successful processing FAIL: Indicates failure
ErrorCode	Integer	Error codes: 0: Indicates success Non-zero: Indicates failure
ErrorInfo	String	Error message
OnlineMemberNum	Integer	The number of online members in this group

Error codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this interface is always 200. The real error code and error message are represented by ErrorCode and ErrorInfo in the response body.

For common error codes (60000 to 79999), see the [Error Code](#) documentation.

The private error codes for this API are as follows:

Error code	Description of Meaning
10002	System error, please try again or contact technical support
10003	The request command is illegal, please try again or contact technical support

10004	Invalid parameter. Please check whether the mandatory fields are filled or if the filled fields meet the protocol requirements based on the ErrorInfo field in the response packet
10007	Insufficient permissions, non-Community types do not support obtaining the number of people online
10010	The group does not exist, or it once existed but has now been dissolved
10015	Group ID is invalid, please check if the Group ID is filled correctly

## API Debugging Tool

Debug this API with the [RESTful API Online Debugging Tool](#).



# Group Counter

## Getting Group Counters

Last updated : 2025-03-03 10:40:35

### Feature Overview

The app admin can use this API to get group counters.

#### Note:

To use this feature, you need to purchase the [Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#).

### API Calling Description

#### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

#### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/get_group_counter?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
-----------	-------------

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_group_attr	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Get group counters:

```
{
  "GroupId": "@TGS#aC5SZEAEF",
  "GroupCounterKeys":[    // List of group counter keys to get. If no value
is passed in, all group counter keys will be returned.
    "like",
    "unlike"
  ]
}
```

## Request fields

Field	Type	Required	Description
-------	------	----------	-------------

GroupId	String	Yes	Group ID
GroupCounterKeys	Array	No	List of group counter keys to get. If no value is passed in, all group counter keys will be returned.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupCounter": [
    {
      "Key": "like", // Group counter key
      "Value": 5 // Latest group counter value
    },
    {
      "Key": "unlike",
      "Value": 1
    }
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
GroupCounter	Array	List of the latest group counter key-value pairs to get

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
------------	-------------

10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	Insufficient operation permission. For example, the current user is not a group member.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.
10059	To use this feature, you need to purchase the Pro edition 、 Pro Plus edition or Enterprise edition.

# Updating Group Counters

Last updated : 2025-03-03 10:40:35

## Feature Overview

The app admin can use this API to update (set/increase/decrease) group counters.

### Note:

To use this feature, you need to purchase the [Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#).

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/update_group_counter?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located.

	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/group_open_http_svc/modify_group_attr	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

Update group counters:

```
{
  "GroupId": "@TGS#aC5SZEAEF",
  "GroupCounter": [
    {
      "Key": "like", // Group counter key
      "Value": 5 // Group counter value
    },
    {
      "Key": "unlike",
      "Value": 1
    }
  ],
  "Mode": "Set" // `Set`, `Increase`, `Decrease`
}
```

## Request fields

Field	Type	Required	Description
GroupId	String	Yes	Group ID
GroupCounter	Array	Yes	List of group counters. <code>Key</code> indicates the key of the group counter. The list can contain up to 20 keys, with each key containing no more than 128 bytes. <code>Value</code> indicates the value of the group counter, which is a 64-bit signed integer.
Mode	String	No	<code>Increase</code> and <code>Decrease</code> indicate increasing or decreasing the original value respectively. If the <code>Key</code> does not exist, create one with <code>Value</code> being 0 and then increase or decrease the value. <code>Set</code> indicates setting <code>Value</code> corresponding to the <code>Key</code> .

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "GroupCounter":[ // List of the latest group counter key-value pairs
updated
    {
      "Key":"like", // Group counter key
      "Value":5 // Latest group counter value
    },
    {
      "Key":"unlike",
      "Value":1
    }
  ]
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

GroupCounter	Array	List of the latest group counter key-value pairs updated
--------------	-------	--

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	Insufficient operation permission. For example, the current user is not a group member.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.
10059	To use this feature, you need to purchase the Pro edition 、 Pro Plus edition or Enterprise edition.
10063	The number of keys in the group counter exceeds the limit of 20.



# Deleting Group Counters

Last updated : 2025-03-03 10:40:35

## Feature Overview

The app admin can use this API to delete group counters.

### Note:

To use this feature, you need to purchase the [Pro edition](#) 、 [Pro Plus edition](#) or [Enterprise edition](#).

## API Calling Description

### Applicable group types

Group Type ID	RESTful API Support
Private	Yes. Same as work groups (Work) in the new version.
Public	Yes
ChatRoom	Yes. Same as the meeting group (Meeting) in the new version.
AVChatRoom	Yes
Community	No

These are the preset group types in Chat. For more information, see [Group System](#).

### Sample request URL

```
https://xxxxxx/v4/group_open_http_svc/delete_group_counter?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes only the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located.

	China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/group_open_http_svc/modify_group_attr	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be json .

Maximum call frequency

200 calls per second

Sample request

Basic format

Deleting group counters:

```
{
  "GroupId": "@TGS#aC5SZEAEF",
  "GroupCounterKeys":[ // List of group counter keys to delete. If no
value is passed in, all group counter keys will be deleted.
    "like",
    "unlike"
  ]
}
```

Request fields

Field	Type	Required	Description
GroupId	String	Yes	Group ID

GroupCounterKeys	Array	No	List of group counter keys to delete. If no value is passed in, all group counter keys will be deleted.
------------------	-------	----	---

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal server error. Try again.
10004	Invalid parameter. Check the error description and troubleshoot the issue.
10007	Insufficient operation permission. For example, the current user is not a group member.
10010	The group does not exist or has been deleted.
10015	Invalid group ID. Check whether the group ID is entered correctly and whether the group exists.
10059	To use this feature, you need to purchase the Pro edition 、 Pro Plus edition or Enterprise edition.

# User Management

## Account Management

### Importing a Single Account

Last updated : 2025-01-14 15:44:02

## Feature Overview

This API is used to import an app-owned account to Chat and create an internal ID for the account to ensure that the account can use the Chat service.

#### Note:

If the same account is imported to Chat repeatedly, only one internal ID will be created for the account.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/im_open_login_svc/account_import?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/im_open_login_svc/account_import	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created

identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

```
{
  "UserID": "test",
  "Nick": "test",
  "FaceUrl": "http://www.qq.com"
}
```

## Request fields

Field	Type	Required	Description
UserID	String	Yes	Username, which is a string of up to 32 bytes in length
Nick	String	No	User nickname
FaceUrl	String	No	URL of the user's profile photo

## Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful. <code>FAIL</code> : failed

ErrorCode	Integer	Error code. 0 : Successful. Other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
40006	An internal server error occurred during profile setting. Try again later.
40601	The value of the profile field exceeds the length limit of 500 bytes.
70169	Server timeout. Try again later.
70398	The number of accounts exceeds the limit allowed. To create more than 100 accounts, upgrade your app to the Premium edition. For specific steps, see <a href="#">Purchase Guide</a> .
70402	Invalid parameters. Check whether the required fields have been specified or whether the fields are set according to protocol requirements.
70403	The request failed. It requires the app admin's permissions.
70500	Internal server error. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing Multiple Accounts ([v4/im\\_open\\_login\\_svc/multiaccount\\_import](#))

Deleting an account ([v4/im\\_open\\_login\\_svc/account\\_delete](#))

Querying Accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Setting Profiles ([v4/profile/portrait\\_set](#))

Invalidating Account Login States ([v4/im\\_open\\_login\\_svc/kick](#))

Querying Account Online Status ([v4/openim/query\\_online\\_status](#))

# Importing Multiple Accounts

Last updated : 2024-12-25 10:04:06

## Feature Overview

This API is used to batch import app-owned accounts into the Chat system, creating a corresponding internal ID for the account, enabling it to use Chat services.

**Note:**

This API supports importing up to 100 accounts at a time. Repeated import of the same account will only create one internal ID.

When batch importing accounts, if an account already exists, it will be updated with the new parameters provided (user nickname, avatar URL, etc.).

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/im_open_login_svc/multiaccount_import?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/im_open_login_svc/multiaccount_import	The request API that is used.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b>



	section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

100 times/second

Sample Request Packets for the Old Version

```
{
  "Accounts": [
    "UserID_001",
    "UserID_002",
    "UserID_003"
  ]
}
```

Request Packet Fields for the Old Version

Field	Type	Attribute	Description
Accounts	Array	Required	Username list, each username length not exceeding 32 bytes, up to 100 usernames can be imported at a time

Sample Request Packets for the New Version

```
{
  "AccountList": [
    {
      "UserID": "UserID_001",
      "Nick": "Nick_001",
      "FaceUrl": "http://www.qq.com"
    },
    {
      "UserID": "UserID_002",
      "Nick": "Nick_002",
      "FaceUrl": "http://www.qq.com"
    },
    {
```

```
    "UserID": "UserID_002",
    "Nick": "Nick_001",
    "FaceUrl": "http://www.qq.com"
  }
]
```

Request Packet Fields for the New Version

Field	Type	Attribute	Description
AccountList	Array	Required	List of users to be imported, up to 100 users can be imported at a time
UserID	String	Required	Username, which is a string of up to 32 bytes in length
Nick	String	Optional	User nickname
FaceUrl	String	Optional	URL of the user's profile photo

Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "FailAccounts": [
    "UserID_002",
    "UserID_003"
  ]
}
```

Response packet field description

Field	Type	Description
ActionStatus	String	Request result. OK : successful. FAIL : failed
ErrorCode	Integer	Error code. 0 : Successful. Other values: Failed
ErrorInfo	String	Error Message
FailAccounts	Array	List of failed account

## Error Code Description

Unless a network error occurs (e.g., 502 error), the HTTP status code for this API is always 200. The actual error code and error message are indicated by `ErrorCode` and `ErrorInfo` in the response body.

Common error codes (60000 to 79999) can be found in the [Error Code](#) document.

Private error codes for this API are as follows:

Error Code	Description
40006	Internal server error when setting data, please try again later.
40601	The length of the Value in the data field exceeds 500 bytes.
70169	Server-side internal timeout, please try again later
70202	Server-side internal timeout, please try again later
70398	Account number exceeds limit. To create more than 100 accounts, please upgrade the application to the professional version. For specific operation guide, see <a href="#">Purchase Guide</a> .
70399	For non-IM trial applications, re-import is not allowed within three months after the account is deleted.
70402	Invalid parameters. Please check whether the required fields are filled or if the fields meet protocol requirements.
70403	Request failed. App administrator permissions are required.
70500	Internal server error, please try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing a Single Account ([v4/im\\_open\\_login\\_svc/account\\_import](#))

Deleting an account ([v4/im\\_open\\_login\\_svc/account\\_delete](#))

Querying Accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Invalidating Account Login States ([v4/im\\_open\\_login\\_svc/kick](#))

Querying Account Online Status ([v4/openim/query\\_online\\_status](#))

# Deleting Accounts

Last updated : 2025-02-28 18:02:34

## Feature Description

The Chat trial edition and Chat Free Trial applications support account deletion by default. For other types of applications (such as: TRTC, whiteboard, Pro edition 、 Pro Plus edition、 Enterprise edition), it needs to be enabled in the console.

### Note:

For specific package types, you can enter the application page in the [Console](#), and click **Application > Application Details > Plan** to view.

### Note:

After deleting an account, **the user's data will be irrecoverable**, please use this interface with caution.

When deleting an account, that user's contacts, profile, groups, sessions, and other data will also be deleted.

The Chat trial edition and Chat free trial edition support importing deleted UserIDs. For other types of applications (such as: TRTC, whiteboard, Pro edition 、 Pro Plus edition、 Enterprise edition), **deleted UserIDs are not allowed to be re-imported within 3 months**.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/im_open_login_svc/account_delete?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table only describes the modified parameters when this API is called. For more information on other parameters, please see [RESTful API Overview](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code>

	Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/im_open_login_svc/account_delete	Request API
sdkappid	<code>SDKAppID</code> assigned by the console when the app is created
identifier	App admin account. For more information, please see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated in the app admin account. For details on how to generate the signature, please see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format. The value is always <code>json</code> .

Maximum calling frequency

100 calls per second

Sample request packet

```
{
  "DeleteItem":
  [
    {
      "UserID": "UserID_1"
    },
    {
      "UserID": "UserID_2"
    }
  ]
}
```

Request packet fields

Field	Type	Required	Description
Deleteltem	Array	Yes	Account array to delete. A single request can contain up to 100 accounts.
UserID	String	Yes	<code>UserID</code> of the account to delete

Sample response packet

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ResultItem": [
    {
      "ResultCode": 0,
      "ResultInfo": "",
      "UserID": "UserID_1"
    },
    {
      "ResultCode": 70107,
      "ResultInfo": "Err_TLS_PT_Open_Login_Account_Not_Exist",
      "UserID": "UserID_2"
    }
  ]
}
```

Response packet fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed
ErrorInfo	String	Error information about the request failure
ResultItem	Array	Array of results for different accounts
ResultCode	Integer	Error code for the account. <code>0</code> : successful; other values: failed
ResultInfo	String	Error information about the failure to delete the account
UserID	String	<code>UserID</code> of the account to delete

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), please see [Error Codes](#). The following table describes the error codes specific to this API:

Error	Description
-------	-------------

Code	
30006	An internal server error occurred while clearing contacts data. Try again later.
30007	Internal server timeout occurred while clearing contacts data. Try again later.
30008	A write conflict occurred while writing contacts data. You are advised to use the batch mode.
40006	An internal server error occurred while clearing profiles. Try again later.
70107	The <code>UserID</code> to delete does not exist. Make sure that the <code>UserID</code> is valid.
70169	Server timeout. Try again later.
70202	Server timeout. Try again later.
70402	Invalid parameters. Make sure that the required fields are all entered and the parameter settings meet the protocol requirements.
70403	Request failed. App admin permissions are required to perform this operation.
70500	Internal server error. Try again later.
71000	Failed to delete accounts. Only accounts of apps using the Chat Trial plan can be deleted. Your current app is using the Pro plan and therefore does not support deleting accounts.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing a Single Account ([v4/im\\_open\\_login\\_svc/account\\_import](#))

Importing Multiple Accounts ([v4/im\\_open\\_login\\_svc/multiaccount\\_import](#))

Querying Accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Invalidating Account Login States ([v4/im\\_open\\_login\\_svc/kick](#))

Querying Account Online Status ([v4/openim/query\\_online\\_status](#))

# Querying Accounts

Last updated : 2024-07-16 11:25:50

## Feature Overview

This API is used to query whether app-owned accounts have been imported to Chat. Batch query is supported.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/im_open_login_svc/account_check?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/im_open_login_svc/account_check	Request API
sdkappid	SDKAppID assigned by the console when the app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295



contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

100 times/second

Sample request

```
{
  "CheckItem":
  [
    {
      "UserID": "UserID_1"
    },
    {
      "UserID": "UserID_2"
    }
  ]
}
```

Request fields

Field	Type	Required	Description
CheckItem	Array	Yes	Account object array to check. A single request can contain up to 100 accounts.
UserID	String	Yes	UserID of the account to check

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ResultItem": [
    {
      "UserID": "UserID_1",
      "ResultCode": 0,
      "ResultInfo": "",
      "AccountStatus": "Imported"
    },
    {
      "UserID": "UserID_2",
      "ResultCode": 0,
```

```
        "ResultInfo": "",
        "AccountStatus": "NotImported"
    }
]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information about the request failure
ResultItem	Array	Array of results for different accounts
UserID	String	UserID of the account to check
ResultCode	Integer	Check result for the account. <code>0</code> : Successful. Other values: Failed
ResultInfo	String	Error information about the failure to check the account
AccountStatus	String	Import status of an account. Valid values: <code>Imported</code> and <code>NotImported</code>

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
70169	Server timeout. Try again later.
70202	Server timeout. Try again later.
70402	Invalid parameters. Make sure that the required fields are all entered and the parameter settings meet the protocol requirements.
70403	Request failed. App admin permissions are required to perform this operation.

70500

Internal server error. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing a Single Account ([v4/im\\_open\\_login\\_svc/account\\_import](#))

Importing Multiple Accounts ([v4/im\\_open\\_login\\_svc/multiaccount\\_import](#))

Deleting an account ([v4/im\\_open\\_login\\_svc/account\\_delete](#))

Invalidating Account Login States ([v4/im\\_open\\_login\\_svc/kick](#))

Querying Account Online Status ([v4/openim/query\\_online\\_status](#))

# User Information

## Setting Profiles

Last updated : 2024-10-10 10:33:28

### Feature Overview

This API is used to set [standard profile fields](#) and [custom profile fields](#).

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/profile/portrait_set?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/profile/portrait_set	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "id",
  "ProfileItem": [
    {
      "Tag": "Tag_Profile_IM_Nick",
      "Value": "MyNickName"
    }
  ]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UserID of the user whose profile needs to be set
ProfileItem	Array	Yes	Profile object array of the target user. Each object in the array contains a tag and a value.
Tag	String	Yes	Profile field to be set. Profile fields that can be set include: 1. <a href="#">Standard Profile Fields</a> 2. <a href="#">Custom Profile Fields</a>
Value	uint64_t/string/bytes	Yes	Value of the profile field to be set. For details, see <a href="#">Profile Fields</a> .

Sample response

```
{
  "ActionStatus": "FAIL",
  "ErrorCode": 40001,
  "ErrorInfo": "Err_Profile_Comm_Decode_Fail",
  "ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
ActionStatus	String	The result of the request. <code>OK</code> means the request was successful. <code>FAIL</code> means the request failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed information displayed on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
40001	Incorrect request parameter. Check your request according to the error description.
40003	The requested account does not exist.
40004	The request requires app admin permission.
40006	Internal server error. Try again later.
40008	No permission to write profile fields.
40009	The tag of a profile field does not exist.
40601	The value of the profile field exceeds the length limit of 500 bytes.
40605	Incorrect value of the standard profile field. For more information, see <a href="#">Standard Profile Fields</a> .
40610	Incorrect value type of the standard profile field. For more information, see <a href="#">Standard Profile Fields</a> .

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Pulling profiles ([v4/profile/portrait\\_get](#))

# Pulling Profiles

Last updated : 2024-10-10 10:33:28

## Feature Overview

This API is used to pull profile fields of friends and other users.

This API can pull [standard profile fields](#) and [custom profile fields](#).

We recommend that you pull the profile fields of no more than 100 users at a time to avoid response packet failure due to excessive data volume.

Ensure that all accounts in the request are imported into the Chat console. Otherwise, an error will be displayed at the Chat backend.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/profile/portrait_get?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/profile/portrait_get	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .



usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

Read a profile field of a user

```
{
  "To_Account":["id1"],
  "TagList":
  [
    "Tag_Profile_IM_Nick"
  ]
}
```

Read multiple profile fields of a user

```
{
  "To_Account":["id1"],
  "TagList":
  [
    "Tag_Profile_IM_Nick",
    "Tag_Profile_IM_AllowType",
    "Tag_Profile_IM_SelfSignature",
    "Tag_Profile_Custom_Test"
  ]
}
```

Read a profile field of multiple users

```
{
  "To_Account":["id1","id2","id3"],
  "TagList":
  [
    "Tag_Profile_IM_Nick"
  ]
}
```

Read multiple profile fields of multiple users

```
{
  "To_Account":["id1","id2","id3","id4"],
  "TagList":
  [
    "Tag_Profile_IM_Nick",
    "Tag_Profile_IM_AllowType",
    "Tag_Profile_IM_SelfSignature",
    "Tag_Profile_Custom_Test"
  ]
}
```

Request fields

Field	Type	Required	Description
To_Account	Array	Required	The UserID of users whose profile fields need to be pulled. Note: profile fields of no more than 100 users can be pulled each time to prevent a response packet failure due to excessive data.
TagList	Array	Required	The tag of a profile field to be pulled. Supported fields include: 1. Standard profile fields. For more information, see <a href="#">Standard Profile Fields</a> . 2. Custom profile fields. For more information, see <a href="#">Custom profile fields</a> .

Sample response

Read a profile field of a user

```
{
  "UserProfileItem":
  [
    {
      "To_Account":"id1",
      "ProfileItem":
      [
        {
          "Tag":"Tag_Profile_IM_Nick",
          "Value":"NickNameTest1"
        }
      ],
      "ResultCode":0,
      "ResultInfo":""
    }
  ],
  "ActionStatus":"OK",
}
```

```
"ErrorCode":0,
"ErrorInfo":"",
"ErrorDisplay":""
}
```

### Read multiple profile fields of a user

```
{
  "UserProfileItem":
  [
    {
      "To_Account":"id1",
      "ProfileItem":
      [
        {
          "Tag":"Tag_Profile_IM_Nick",
          "Value":"NickNameTest1"
        },
        {
          "Tag":"Tag_Profile_IM_AllowType",
          "Value":"AllowType_Type_NeedConfirm"
        },
        {
          "Tag":"Tag_Profile_IM_SelfSignature",
          "Value":"I'm Test1"
        },
        {
          "Tag":"Tag_Profile_Custom_Test",
          "Value":"Custom Data1"
        }
      ],
      "ResultCode":0,
      "ResultInfo":""
    }
  ],
  "ActionStatus":"OK",
  "ErrorCode":0,
  "ErrorInfo":"",
  "ErrorDisplay":""
}
```

### Read a profile field of multiple users

```
{
  "UserProfileItem":
  [
    {
```

```

        "To_Account": "id1",
        "ProfileItem":
        [
            {
                "Tag": "Tag_Profile_IM_Nick",
                "Value": "NickNameTest1"
            }
        ],
        "ResultCode": 0,
        "ResultInfo": ""
    },
    {
        "To_Account": "id2",
        "ProfileItem":
        [
            {
                "Tag": "Tag_Profile_IM_Nick",
                "Value": "NickNameTest2"
            }
        ],
        "ResultCode": 0,
        "ResultInfo": ""
    },
    {
        "To_Account": "id3",
        "ProfileItem":
        [
            {
                "Tag": "Tag_Profile_IM_Nick",
                "Value": "NickNameTest3"
            }
        ],
        "ResultCode": 0,
        "ResultInfo": ""
    }
],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}

```

### Read multiple profile fields of multiple users

```

{
    "UserProfileItem":
    [

```

```
{
  "To_Account":"id1",
  "ProfileItem":
  [
    {
      "Tag":"Tag_Profile_IM_Nick",
      "Value":"NickNameTest1"
    },
    {
      "Tag":"Tag_Profile_IM_AllowType",
      "Value":"AllowType_Type_NeedConfirm"
    },
    {
      "Tag":"Tag_Profile_IM_SelfSignature",
      "Value":"I'm Test1"
    },
    {
      "Tag":"Tag_Profile_Custom_Test",
      "Value":"Custom Data1"
    }
  ],
  "ResultCode":0,
  "ResultInfo":""
},
{
  "To_Account":"id2",
  "ProfileItem":
  [
    {
      "Tag":"Tag_Profile_IM_Nick",
      "Value":"NickNameTest2"
    },
    {
      "Tag":"Tag_Profile_IM_AllowType",
      "Value":"AllowType_Type_DenyAny"
    },
    {
      "Tag":"Tag_Profile_IM_SelfSignature",
      "Value":"I'm Test2"
    },
    {
      "Tag":"Tag_Profile_Custom_Test",
      "Value":"Custom Data2"
    }
  ],
  "ResultCode":0,
  "ResultInfo":""
}
```

```
    },
    {
      "To_Account": "id3",
      "ProfileItem": [
        {
          "Tag": "Tag_Profile_IM_Nick",
          "Value": "NickNameTest3"
        },
        {
          "Tag": "Tag_Profile_IM_AllowType",
          "Value": "AllowType_Type_AllowAny"
        },
        {
          "Tag": "Tag_Profile_IM_SelfSignature",
          "Value": "I'm Test3"
        },
        {
          "Tag": "Tag_Profile_Custom_Test",
          "Value": "Custom Data3"
        }
      ],
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id4",
      "ResultCode": 40006,
      "ResultInfo": "Err_Profile_PortraitGet_Read_Custom_Data_Fail"
    }
  ],
  "Fail_Account": ["id4"],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
UserProfileItem	Array	Structured information of the returned user profile.
To_Account	String	UserID of a user.
ProfileItem	Array	The returned user profile object array. Each object in the array

		contains a tag and a value.
Tag	String	The returned profile object name. The profile objects include: 1. Standard profile fields. For more information, see <a href="#">Standard Profile Fields</a> 2. Custom profile fields. For more information, see <a href="#">Custom Profile Fields</a> .
Value	uint64_t/string/bytes	The value of a pulled profile object. For more information, see <a href="#">Profile Fields</a> .
ResultCode	Integer	The processing result of To_Account. 0: Succeeded. Other values: Failed.
ResultInfo	String	Error description of <code>To_Account</code> . This field is empty if the request is successful.
Fail_Account	Array	List of users whose processing failed. This field is only returned when at least one user fails.
ActionStatus	String	The result of the request. <code>OK</code> means the request was successful. <code>FAIL</code> means the request failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed information displayed on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
40001	Incorrect request parameter. Check your request according to the error description.
40002	Request parameter error. To_Account is not specified.
40003	The requested account does not exist.
40004	The request requires app admin permission.
40006	Internal server error. Try again later.

40007	No permission to read profile fields.
40009	The tag of a profile field does not exist.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Configuring Profiles ([v4/profile/portrait\\_set](#)).



# User Status

## Invalidating Account Login States

Last updated : 2025-03-10 10:30:48

### Feature Description

This API is used to invalidate the login state (for example, UserSig) of an app account.

For example, when a developer finds a malicious account, the developer can call this API to invalidate the user's login state. When the user uses the original UserSig state to log in to Chat, the login fails.

#### Note

After a user's login state is invalidated using this API, the user can use a newly generated UserSig state to log in to Chat. This API can invalidate only one account at a time.

### Call Description

#### Example request URL

```
https://xxxxxx/v4/im_open_login_svc/kick?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table only describes the parameters that are modified when this API is called. For more information on other parameters, see [RESTful API Overview](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/im_open_login_svc/kick	The request API.

sdkappid	The SDKAppID assigned by the Chat console when an app is created.
identifier	The value of this parameter must be an app admin account. For more information, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details on the operation, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is always <code>json</code> .

## Maximum call frequency

200 times/second

## Example request packet

```
{
  "UserID": "test"
}
```

## Request packet fields

Field	Type	Attribute	Description
UserID	String	Required	Username

## Example response packet

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response packet fields

Field	Type	Description
ActionStatus	String	The request processing result. OK: succeeded. FAIL: failed.
ErrorCode	Integer	The error code. 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packet represent the actual error code and error information.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API.

Error Code	Description
70107	The requested user account does not exist.
70169	An internal server timeout has occurred. Please try again later.
70402	The parameters are invalid. Check whether required fields are specified and specified fields meet protocol requirements.
70403	The request requires app admin permissions.
70500	An internal server error has occurred. Please try again later.

## Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing an account ([v4/im\\_open\\_login\\_svc/account\\_import](#))

Importing multiple accounts ([v4/im\\_open\\_login\\_svc/multiaccount\\_import](#))

Deleting accounts ([v4/im\\_open\\_login\\_svc/account\\_delete](#))

Querying accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Querying the online status of an account ([v4/openim/query\\_online\\_status](#))

# Querying Account Online Status

Last updated : 2025-03-10 10:30:48

## Feature Description

This API is used to query the current login status of a user.

## API Call Description

### Sample request URL

```
https://xxxxxx/v4/openim/query_online_status?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following is a list of the parameters commonly used when calling this API and their descriptions. For more parameters, see the [RESTful API Overview](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/openim/query_online_status	The request API that is used.
sdkappid	The SDKAppID assigned via the Chat console when the application is created.
identifier	The value must be the app admin account. For more information, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For more information on the operation, see <a href="#">Generating UserSig</a> .

random	The value must be a random 32-bit unsigned integer. Value range: 0 to 4294967295.
contenttype	Request format. The value is always <code>json</code> .

### Maximum call frequency

This API can be called up to 200 times per second.

### Sample request packets

#### When detailed login platform information is not needed

```
{
  "To_Account": ["id1", "id2", "id3", "id4"],
}
```

#### When detailed login platform information is needed

```
{
  "IsNeedDetail": 1,
  "To_Account": ["id1", "id2", "id4"]
}
```

### Request packet fields

Field	Type	Property	Description
To_Account	Array	Required	The one or more UserIDs whose login statuses are to be queried. This API can be used to query the login statuses of up to 500 UserIDs at a time.
IsNeedDetail	Integer	Optional	Specifies whether detailed login platform information is needed in the response. 0: not needed. 1: needed.

### Sample response packet body

#### When detailed login platform information is not needed

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

```
"QueryResult": [
  {
    "To_Account": "id1",
    "State": "Offline"
  },
  {
    "To_Account": "id2",
    "State": "Online"
  },
  {
    "To_Account": "id3",
    "State": "PushOnline"
  }
],
"ErrorList": [
  {
    "To_Account": "id4",
    "ErrorCode": 70107
  }
]
}
```

#### When detailed login platform information is needed

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
  "QueryResult": [
    {
      "To_Account": "id1",
      "State": "Online",
      "Detail": [
        {
          "Platform": "IPhone",
          "Status": "PushOnline",
          "IsBackground": 1,
          "Instid": 74224656,
          "CustomIdentifier": "device-1"
        },
        {
          "Platform": "Web",
          "Status": "Online",
          "IsBackground": 1,
          "Instid": 74224656,

```

```
        "CustomIdentifier": "device-2"
      }
    ],
    },
    {
      "To_Account": "id2",
      "State": "Offline",
    }
  ],
  "ErrorList": [
    {
      "To_Account": "id4",
      "ErrorCode": 70107
    }
  ]
}
```

Request error

```
{
  "ActionStatus": "FAIL",
  "ErrorInfo": "Fail to Parse json data of body, Please check it",
  "ErrorCode": 90001
}
```

Response packet fields

Field	Type	Description
ActionStatus	String	The processing result of the request. OK: succeeded. FAIL: failed.
ErrorInfo	String	Detailed information on the error.
ErrorCode	Integer	The error code returned for the request. If the status query for any account succeed, the value is 0. If the query for all the accounts failed, the return value is not 0.
QueryResult	Array	The returned structured information of the login status of the user.
QueryResult.To_Account	String	The UserID of the user whose status is returned.
QueryResult.Status	String	The returned login status. Valid values:

		<p>Online: after the user logs in to the client, the client remains in a persistent connection with the Chat backend.</p> <p>PushOnline: the client enters the PushOnline state when the iOS or Android process is disconnected due to a network error or is killed by the operating system. In this state, the client still can receive offline messages. However, if the client's process is not terminated by the operating system after the client is switched to the background, the client is in Online state.</p> <p>Offline: the user has logged out of the client properly or has not logged in to the client for at least 7 days since the last login.</p> <p>If the user logs in to the client on multiple devices, the value is Online provided that the client is in the Online state on any device.</p>
QueryResult.Detail	Object	The detailed information on the login platform.
QueryResult.Detail.Platform	String	The type of the login platform. Valid values: "iPhone", "Android", "Web", "PC", "iPad", and "Mac".
QueryResult.Detail.Instid	Integer	The unique identifier of this device.
QueryResult.Detail.CustomIdentifier	String	The custom identifier of this device.
ErrorList	Array	<p>The list of accounts whose statuses failed to be queried.</p> <p>The target accounts in this list were not found or their statuses failed to be queried. If the status query for all accounts succeeded, the value of the ErrorList field is blank.</p>
ErrorList.To_Account	String	The target account whose status failed to be queried.
ErrorList.ErrorCode	Integer	The error code indicating that the status query failed. If the error code for a target account is 70107, the account was not found.

### Caution

The Chat backend stores the PushOnline state for only 7 days. If a user has not logged in to the client within 7 days since the previous login, the user enters the Offline state.

## Error Codes

The HTTP return code for this API is 200 unless a network error such as error 502 occurs. The actual error code and error information are indicated by ErrorCode and ErrorInfo respectively in the response packet body.



For public error codes 60000 to 79999, see [Error Codes](#).

The private error codes for this API are as follows:

Error Code	Description
70107	The requested UserID does not exist.
70169	The server timed out. Try again later.
90001	JSON format parsing failed. Check whether the request packet meets the JSON specification, or whether To_Account is a null array.
90003	The value of the To_Account field in the JSON format request packet does not meet the message format requirements. Please check whether the type of the To_Account field is String.
90009	The request requires the app admin's permissions.
90011	The number of target accounts to which the message is to be sent exceeds 500. Reduce the number of target accounts in To_Account.
90992	The backend service timed out. Please try again.
90994	An internal service error occurred. Please try again.
90995	An internal service error occurred. Please try again.
91000	An internal service error occurred. Please try again.

## API Debugging Tool

To debug this API, you can use the [Online RESTful API Debugging Tool](#).

## References

Importing an Account ([v4/im\\_open\\_login\\_svc/account\\_import](#))

Batch Importing Multiple Accounts ([v4/im\\_open\\_login\\_svc/multiaccount\\_import](#))

Deleting an Account ([v4/im\\_open\\_login\\_svc/account\\_delete](#))

Querying an Account ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Invalidating the Login Status of an Account ([v4/im\\_open\\_login\\_svc/kick](#))

# Friend Management

## Adding Friends

Last updated : 2024-09-25 15:13:30

### Feature Overview

This API is used to add one or multiple users to the friend list.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/sns/friend_add?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/sns/friend_add	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype

Request format, which should always be `json` .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

```
{
  "From_Account": "id",
  "AddFriendItem":
  [
    {
      "To_Account": "id1",
      "AddSource": "AddSource_Type_XXXXXXX"
    }
  ]
}
```

### Response to a complete request

```
{
  "From_Account": "id",
  "AddFriendItem":
  [
    {
      "To_Account": "id1",
      "Remark": "remark1",
      "GroupName": "Classmates", // Each user can only be assigned to one
friend list when the user is added as a friend. Therefore, we can use `String`
as the data type.
      "AddSource": "AddSource_Type_XXXXXXX",
      "AddWording": "I'm Test1"
    }
  ],
  "AddType": "Add_Type_Both",
  "ForceAddFlags": 1
}
```

### Response to a batch request

```
{
  "From_Account": "id",
  "AddFriendItem":
  [
```

```

{
  "To_Account": "id1",
  "AddSource": "AddSource_Type_XXXXXXX",
},
{
  "To_Account": "id2",
  "Remark": "remark2",
  "GroupName": "Classmates", // Each user can only be assigned to one
  friend list when the user is added as a friend. Therefore, we can use `String`
  as the data type.
  "AddSource": "AddSource_Type_XXXXXXX",
  "AddWording": "I'm Test2"
},
{
  "To_Account": "id3",
  "Remark": "remark3",
  "GroupName": "Colleagues", // Each user can only be assigned to one
  friend list when the user is added as a friend. Therefore, we can use `String`
  as the data type.
  "AddSource": "AddSource_Type_XXXXXXX",
  "AddWording": "I'm Test3"
}
],
"AddType": "Add_Type_Both",
"ForceAddFlags": 1
}

```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID that initiates the request, the number of friends in a single request must not exceed 1000.
AddFriendItem	Array	Yes	An array of friend objects.
To_Account	String	Yes	The UserID to add as a friend.
Remark	String	No	Friend remarks set by <code>From_Account</code> for <code>To_Account</code> . For more information, see the <b>Standard friend fields</b> section in <a href="#">Contacts Management</a> .
GroupName	String	No	The friend list that the user who initiates the friend request assigns to the user to be added. Each user can only be assigned to one friend list. Therefore, we can use the <code>String</code> data type. For details, see the standard friend fields section in <a href="#">Contacts Management</a> .

AddSource	String	No	Source from which a friend is added. For more information, see the <code>Standard friend fields</code> section in <a href="#">Contacts Management</a> .
AddWording	String	No	Remarks that the user who initiates the friend request writes about the user to be added. For details, see the standard friend fields section in <a href="#">Contacts Management</a> .
AddType	String	No	Friend adding type: Add_Type_Single: one-sided Add_Type_Both (default): mutual
ForceAddFlags	Integer	No	Flag denoting the friend is force added by an administrator: 1 means force added while 0 means the friend is added normally.

Sample response

Response to a basic or complete request

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response to a batch request

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "ResultCode": 30006,

```

```
      "ResultInfo": "Err_SNS_FriendAdd_Unpack_Profile_Data_Fail"
    },
    {
      "To_Account": "id3",
      "ResultCode": 30002,
      "ResultInfo": "Err_SNS_FriendAdd_SdkAppId_Illegal"
    }
  ],
  "Fail_Account": ["id2", "id3"],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response fields

Field	Type	Description
ResultItem	Array	The result of adding friends in bulk, which is an array of UserIDs and corresponding results.
To_Account	String	The UserID that you requested to add as a friend.
ResultCode	Integer	The result. <code>0</code> means success and other values mean failure. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	Error description of <code>To_Account</code> . This field is empty if the request is successful.
Fail_Account	Array	List of users whose processing failed. This field is only returned when at least one user fails.
ActionStatus	String	The result of the request. <code>OK</code> means the request was successful. <code>FAIL</code> means the request failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed information displayed on the client.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30002	The SDKAppID does not match.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.
30009	You have been prohibited from adding friends.
30010	Your friend list is full.
30011	The maximum number of friend lists has been reached.
30012	You have too many pending friend requests.
30014	The user you are trying to add has too many friends.
30515	The user you are trying to add is on your blocklist. You cannot add this user.
30516	The user you are trying to add has disabled friend requests.
30525	You have been blocklisted by the user you are trying to add. You cannot add this user.
30539	The user you are trying to add has selected <code>AllowType_Type_NeedConfirm</code> as their friend request authentication method. Your friend request is pending approval. This code is used to differentiate a successful friend request, meaning the friend is added, and a pending friend request, so more helpful messages can be displayed.
30540	You have sent too many friend requests in a short amount of time. Request filtered for security

reasons.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Importing Friends ([v4/sns/friend\\_import](#))

Updating Friends ([v4/sns/friend\\_update](#))

Deleting Friends ([v4/sns/friend\\_delete](#))

Deleting All Friends ([v4/sns/friend\\_delete\\_all](#))

Verifying Friends ([v4/sns/friend\\_check](#))

Pulling Friends ([v4/sns/friend\\_get](#))

Pulling Specified Friends ([v4/sns/friend\\_get\\_list](#))



# Importing Friends

Last updated : 2024-09-25 15:13:30

## Feature Overview

This API is used to import one-way friends in batches.  
We recommend that you import friends for the same user in batches to avoid write conflicts due to concurrent writes.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/friend_import?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/friend_import	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

```
{
  "From_Account": "id",
  "AddFriendItem":
  [
    {
      "To_Account": "id1",
      "AddSource": "AddSource_Type_XXXXXXX"
    }
  ]
}
```

### Response to a complete request

```
{
  "From_Account": "id",
  "AddFriendItem":
  [
    {
      "To_Account": "id1",
      "Remark": "remark1",
      "RemarkTime": 1420000001,
      "GroupName": ["Friends"],
      "AddSource": "AddSource_Type_XXXXXXX",
      "AddWording": "I'm Test1",
      "AddTime": 1420000001,
      "CustomItem":
      [
        {
          "Tag": "Tag_SNS_Custom_XXXX",
          "Value": "Test"
        },
        {
          "Tag": "Tag_SNS_Custom_YYYY",
          "Value": 0
        }
      ]
    }
  ]
}
```

```
]
}
```

### Response to a batch request

```
{
  "From_Account": "id",
  "AddFriendItem": [
    {
      "To_Account": "id1",
      "AddSource": "AddSource_Type_XXXXXXX"
    },
    {
      "To_Account": "id2",
      "Remark": "remark2",
      "RemarkTime": 1420000001,
      "GroupName": ["Friends"],
      "AddSource": "AddSource_Type_XXXXXXX",
      "AddWording": "I'm Test2",
      "AddTime": 1420000001
    },
    {
      "To_Account": "id3",
      "Remark": "remark3",
      "RemarkTime": 1420000001,
      "GroupName": ["Colleagues", "Friends"],
      "AddSource": "AddSource_Type_XXXXXXX",
      "AddWording": "I'm Test3",
      "AddTime": 1420000001,
      "CustomItem": [
        {
          "Tag": "Tag_SNS_Custom_XXXX",
          "Value": "Test"
        },
        {
          "Tag": "Tag_SNS_Custom_YYYY",
          "Value": 0
        }
      ]
    }
  ]
}
```

### Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID that initiates the request.
AddFriendItem	Array	Yes	Structural object of the friend, the number of friends in a single request must not exceed 1000.
To_Account	String	Yes	<code>UserID</code> of a friend.
Remark	String	No	Friend remarks set by <code>From_Account</code> for <code>To_Account</code> . For more information, see <a href="#">Standard friend fields</a> .
RemarkTime	Integer	No	Time of remarking <code>To_Account</code> by <code>From_Account</code> .
GroupName	Array	No	Group information of <code>To_Account</code> added by <code>From_Account</code> . For more information, see <a href="#">Standard friend fields</a> .
AddSource	String	Yes	Source from which a friend is added. For more information, see <a href="#">Standard friend fields</a> .
AddWording	String	No	Remarks that the user who initiates the friend request writes about the user to be added. For more information, see <a href="#">Standard friend fields</a> .
AddTime	Integer	No	Time of establishing a friend relationship between <code>From_Account</code> and <code>To_Account</code> .
CustomItem	Array	No	Custom friend data of <code>To_Account</code> set by <code>From_Account</code> . Each member has a Tag field and a Value field. For more information, see <a href="#">Custom friend fields</a> .
Tag	String	No	Name of the custom friend field. To use it, first apply for a custom friend field in the <a href="#">Chat console</a> by navigating to <b>App Configuration &gt; eature Configuration</b> .
Value	String/Integer	No	Value of the custom friend field.

## Sample response

### Response to a basic or complete request

```
{
  "ResultItem":
  [
    {
```

```
        "To_Account": "id1",
        "ResultCode": 0,
        "ResultInfo": ""
    },
    ],
    "ActionStatus": "OK",
    "ErrorCode": 0,
    "ErrorInfo": "",
    "ErrorDisplay": ""
}
```

Response to a batch request

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "ResultCode": 30010,
      "ResultInfo": "Err_SNS_FriendImport_My_Friend_Num_Exceed_Threshold"
    },
    {
      "To_Account": "id3",
      "ResultCode": 30002,
      "ResultInfo": "Err_SNS_FriendImport_SdkAppId_Illegal"
    }
  ],
  "Fail_Account": ["id2", "id3"],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response fields

Field	Type	Description
ResultItem	Array	Result of adding friends in bulk, which is an array of UserIDs and corresponding results.
To_Account	String	UserID that you requested to add as a friend.

ResultCode	Integer	Result of <code>To_Account</code> . <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	Error description of <code>To_Account</code> . This field is empty if the request is successful.
Fail_Account	Array	List of users whose processing failed. This field is only returned when at least one user fails.
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30002	The SDKAppID does not match.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.

30010	The number of friends has reached the upper limit of the system.
30011	The maximum number of friend lists has been reached.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

- [Adding Friends](#)
- [Updating Friends](#)
- [Deleting Friends](#)
- [Deleting All Friends](#)
- [Verifying Friends](#)
- [Pulling Friends](#)
- [Pulling Specified Friends](#)

# Updating Friends

Last updated : 2024-09-25 15:13:30

## Feature Overview

This API is used to update the contacts data of multiple friends of a user at a time.  
You are advised to update multiple friends of a user at a time to avoid write conflicts caused by concurrent writes.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/friend_update?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/friend_update	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.



contenttype	Request format, which should always be json .
-------------	---

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

```
{
  "From_Account": "id",
  "UpdateItem":
  [
    {
      "To_Account": "id1",
      "SnsItem":
      [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark1"
        }
      ]
    }
  ]
}
```

### Response to a complete request

```
{
  "From_Account": "id",
  "UpdateItem":
  [
    {
      "To_Account": "id1",
      "SnsItem":
      [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark1"
        },
        {
          "Tag": "Tag_SNS_IM_Group",
          "Value":
          [
            "group1",
            "group2"
          ]
        }
      ]
    }
  ]
}
```

```

    ],
    },
    {
        "Tag": "Tag_SNS_Custom_Test",
        "Value": "test"
    }
]
}
]
}

```

### Response to a batch request

```

{
  "From_Account": "id",
  "UpdateItem": [
    {
      "To_Account": "id1",
      "SnsItem": [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark1"
        }
      ]
    },
    {
      "To_Account": "id2",
      "SnsItem": [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark2"
        },
        {
          "Tag": "Tag_SNS_IM_Group",
          "Value": [
            "group1",
            "group2"
          ]
        }
      ]
    },
    {
      "To_Account": "id3",

```

```
      "SnsItem":
      [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark3"
        },
        {
          "Tag": "Tag_SNS_IM_Group",
          "Value":
          [
            "group3"
          ]
        },
        {
          "Tag": "Tag_SNS_Custom_Test",
          "Value": "test"
        }
      ]
    }
  ]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	<code>UserID</code> of the account for which to update contacts data.
UpdateItem	Array	Yes	An object array of friends to be updated.The number of friends per request should not exceed 1000.
To_Account	String	Yes	<code>UserID</code> of a friend.
SnsItem	Array	Yes	An object array of the contacts data to be updated.
Tag	String	Yes	The name of a contacts field to be updated. Users are only allowed to update the remarks, group, and custom fields of contacts. For more information on contacts fields, see the <b>Friend Lists</b> in <a href="#">Relationship Chain Management</a> .
Value	Array/String/Integer	Yes	The value of a contacts field. For information on value types, see the <b>Friend Lists</b> section in <a href="#">Relationship Chain Management</a> .

## Sample response

### Response to a basic or complete request

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

### Response to a batch request

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "ResultCode": 30011,
      "ResultInfo": "Err_SNS_FriendUpdate_Group_Num_Exceed_Threshold"
    },
    {
      "To_Account": "id3",
      "ResultCode": 30002,
      "ResultInfo": "Err_SNS_FriendImport_SdkAppId_Illegal"
    }
  ],
  "Fail_Account": ["id2", "id3"],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
ResultItem	Array	The result of updating friends, which is an array of UserIDs and corresponding results.
To_Account	String	<code>UserID</code> of the friend that you requested to update.
ResultCode	Integer	The result of <code>To_Account</code> . <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	The error description of <code>To_Account</code> . This field is empty if the request succeeds.
Fail_Account	Array	The users that failed to be verified. This field is only returned when at least one user fails.
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed.
ErrorCode	Integer	Error code. <code>0</code> : successful. Other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30002	The SDKAppID does not match.
30003	The requested account does not exist.

30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.
30011	The maximum number of friend lists has been reached.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Adding Friends](#)

[Importing Friends](#)

[Deleting Friends](#)

[Deleting All Friends](#)

[Verifying Friends](#)

[Pulling Friends](#)

[Pulling Specified Friends](#)

# Deleting Friends

Last updated : 2024-07-16 11:25:50

## Feature Overview

This API is used to delete friends. Both one-way deletion and two-way deletion are supported.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/friend_delete?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/sns/friend_delete	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### One-way deletion

```
{
  "From_Account": "id",
  "To_Account": ["id1", "id2", "id3"],
  "DeleteType": "Delete_Type_Single"
}
```

### Two-way deletion

```
{
  "From_Account": "id",
  "To_Account": ["id1", "id2", "id3"],
  "DeleteType": "Delete_Type_Both"
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the account that requests to delete friends
To_Account	Array	Yes	The list of UserIDs of the friends to be deleted. Each request cannot contain more than 1000 UserIDs.
DeleteType	String	No	Deletion mode. For details, see <a href="#">Deleting Friends</a> .

## Sample response

```
{
  "ResultItem": [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "ResultCode": 0,

```



```
        "ResultInfo":""
    },
    {
        "To_Account":"id3",
        "ResultCode":0,
        "ResultInfo":""
    }
],
"ActionStatus":"OK",
"ErrorCode":0,
"ErrorInfo":"0",
"ErrorDisplay":""
}
```

Response fields

Field	Type	Description
ResultItem	Array	The result object array of deleting friends in batch
To_Account	String	The UserID you requested to delete
ResultCode	Integer	The process result of <code>To_Account</code> . <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	The error description of <code>To_Account</code> . This field is empty if the request succeeds.
ActionStatus	String	The request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed.
ErrorCode	Integer	Error code. <code>0</code> : successful. Other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30002	The SDKAppID does not match.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.
31704	The account you requested to delete is not your friend.
31707	The friend deletion request was filtered by the security policy. Do not initiate friend deletion requests too frequently.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Adding Friends](#)

[Importing Friends](#)

[Updating Friends](#)

[Deleting All Friends](#)

[Verifying Friends](#)

[Pulling Friends](#)

[Pulling Specified Friends](#)

## Possible Webhooks

After a Friend Is Deleted

# Deleting All Friends

Last updated : 2024-07-16 11:25:50

## Feature Overview

This API is used to delete the standard and custom friend data of a specified user.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/friend_delete_all?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/sns/friend_delete_all	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

### One-way deletion

```
{
  "From_Account": "id"
}
```

### Two-way deletion

```
{
  "From_Account": "id",
  "DeleteType": "Delete_Type_Both"
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the account that requests to delete friends
DeleteType	String	No	Deletion mode. One-way deletion is the default mode. For details, see <a href="#">Deleting Friends</a> .

## Sample responses

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
ActionStatus	String	The request result. <code>OK</code> : successful. <code>FAIL</code> : failed.
ErrorCode	Integer	Error code. <code>0</code> : successful. Other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .

ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Adding Friends](#)

[Importing Friends](#)

[Updating Friends](#)

[Deleting Friends](#)

[Verifying Friends](#)

Pulling Friends

Pulling Specified Friends

# Verifying Friends

Last updated : 2024-07-16 11:25:50

## Feature Overview

This API is used to check friendship in bulk.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/friend_check?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/sns/friend_check	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be json .



## Maximum call frequency

200 calls per second

## Sample request

```
{
  "From_Account": "id",
  "To_Account": ["id1", "id2", "id3", "id4", "id5"],
  "CheckType": "CheckResult_Type_Both"
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the account that requests to check friendship
To_Account	Array	Yes	The UserIDs of the friends to be checked. Each request cannot contain more than 1,000 UserIDs.
CheckType	String	Yes	Verification mode. For more information, see <a href="#">Verifying friends</a> .

## Sample responses

```
{
  "InfoItem": [
    {
      "To_Account": "id1",
      "Relation": "CheckResult_Type_BothWay",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "Relation": "CheckResult_Type_AWithB",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id3",
      "Relation": "CheckResult_Type_BWithA",
      "ResultCode": 0,
      "ResultInfo": ""
    }
  ]
}
```

```

        "To_Account": "id4",
        "Relation": "CheckResult_Type_NoRelation",
        "ResultCode": 0,
        "ResultInfo": ""
    },
    {
        "To_Account": "id5",
        "Relation": "CheckResult_Type_NoRelation",
        "ResultCode": 30006,
        "ResultInfo": "Err_SNS_FriendCheck_Check_Relation_Exec_Task_Fail"
    }
],
"Fail_Account": ["id5"],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}

```

## Response fields

Field	Type	Description
InfoItem	Array	The object array of verification results
To_Account	String	The UserID of the account that you requested to check
Relation	String	The friend relationship between <code>To_Account</code> and <code>From_Account</code> upon successful verification. For details, see <a href="#">Verifying friends</a> .
ResultCode	Integer	The process result of <code>To_Account</code> . <code>0</code> : successful. Other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	The error description of <code>To_Account</code> . This field is empty if the request succeeds.
Fail_Account	Array	The users that failed to be verified. This field is only returned when at least one user fails.
ActionStatus	String	The request result. <code>OK</code> : successful. <code>FAIL</code> : failed.
ErrorCode	Integer	Error code. <code>0</code> : successful. Other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Adding Friends](#)

[Importing Friends](#)

[Updating Friends](#)

[Deleting Friends](#)

[Deleting All Friends](#)

[Pulling Friends](#)

[Pulling Specified Friends](#)

# Pulling Friends

Last updated : 2024-07-16 11:25:50

## Feature Overview

This API is used to pull the data of all friends by page.  
It cannot pull profile data.  
You do not need to specify the fields to pull. By default, all standard and custom friend data will be returned.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/friend_get?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/friend_get	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

200 calls per second

Sample requests

```
{
  "From_Account": "id",
  "StartIndex": 0,
  "StandardSequence": 0,
  "CustomSequence": 0
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the account that requests to pull friend data
StartIndex	Integer	Yes	The starting point of the page to be pulled
StandardSequence	Integer	No	The <code>StandardSequence</code> returned for the previous friend data pull. If the value of the <code>StandardSequence</code> field is the same as that in the backend, the backend will not return standard friend data.
CustomSequence	Integer	No	The <code>CustomSequence</code> returned for the previous friend data pull. If the value of the <code>CustomSequence</code> field is the same as that in the backend, the backend will not return custom friend data.

Sample responses

```
{
  "UserDataItem": [
    {
      "To_Account": "id1",
      "ValueItem": [
        {
          "Tag": "Tag_SNS_IM_AddSource",
          "Value": "AddSource_Type_Android"
        }
      ]
    }
  ]
}
```

```
        {
            "Tag": "Tag_SNS_IM_Remark",
            "Value": "Remark1"
        },
        {
            "Tag": "Tag_SNS_IM_Group",
            "Value": ["Group1", "Group2"]
        },
        {
            "Tag": "Tag_SNS_IM_AddTime",
            "Value": 1563867420
        },
        {
            "Tag": "Tag_SNS_Custom_Test",
            "Value": "CustomData1"
        }
    ]
},
{
    "To_Account": "id2",
    "ValueItem": [
        {
            "Tag": "Tag_SNS_IM_AddSource",
            "Value": "AddSource_Type_IOS"
        },
        {
            "Tag": "Tag_SNS_IM_Group",
            "Value": ["Group1"]
        },
        {
            "Tag": "Tag_SNS_IM_AddTime",
            "Value": 1563867425
        }
    ]
}
],
"StandardSequence": 88,
"CustomSequence": 46,
"FriendNum": 20,
"CompleteFlag": 1,
"NextStartIndex": 0,
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

## Response fields

Field	Type	Description
UserDataItem	Array	The friend object array. Each friend object contains a <code>To_Account</code> field and a <code>ValueItem</code> array.
To_Account	String	The UserID of a friend
ValueItem	Array	The array for storing friend data. Each array element contains a <code>Tag</code> field and a <code>Value</code> field.
Tag	String	The name of a friend field
Value	String/Integer/Array	The value of the friend field. For more information, see <a href="#">Relationship Chain Fields</a> .
StandardSequence	Integer	The sequence for standard friend data. The client can save this sequence and return it to the backend via the <code>StandardSequence</code> field in the next request.
CustomSequence	Integer	The sequence for custom friend data. The client can save this sequence and return it to the backend via the <code>CustomSequence</code> field in the next request.
FriendNum	Integer	The total number of friends
CompleteFlag	Integer	The ending tag of the page. A non-zero value indicates that all friend data is pulled.
NextStartIndex	Integer	The starting point of the next page
ActionStatus	String	The request result. <code>OK</code> : successful. <code>FAIL</code> : failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[Adding Friends](#)

[Importing Friends](#)

[Updating Friends](#)

[Deleting Friends](#)

[Deleting All Friends](#)

[Verifying Friends](#)

[Pulling Specified Friends](#)



# Pulling Specified Friends

Last updated : 2024-07-16 11:25:50

## Feature Overview

This API allows you to pull the data and profiles of specified friends.

We recommend that you pull no more than 100 friends at a time to avoid response packet failure due to excessive data volume.

## API Calling Description

### Request URL

```
https://xxxxxx/v4/sns/friend_get_list?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/friend_get_list	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype

Request format, which should always be `json` .

## Maximum call frequency

200 calls per second

## Sample request

### Basic format

```
{
  "From_Account": "UserID_1",
  "To_Account":
  [
    "UserID_2"
  ],
  "TagList":
  [
    "Tag_Profile_Custom_Test",
    "Tag_Profile_IM_Image",
    "Tag_Profile_IM_Nick",
    "Tag_SNS_Custom_Test",
    "Tag_SNS_IM_Remark",
    "Tag_SNS_IM_Group"
  ]
}
```

### Response to a batch request

```
{
  "From_Account": "UserID_1",
  "To_Account":
  [
    "UserID_2",
    "UserID_3",
    "UserID_4"
  ],
  "TagList":
  [
    "Tag_Profile_Custom_Test",
    "Tag_Profile_IM_Image",
    "Tag_Profile_IM_Nick",
    "Tag_SNS_Custom_Test",
    "Tag_SNS_IM_Remark",
    "Tag_SNS_IM_Group"
  ]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID of the account that requests to pull friend data.
To_Account	Array	Yes	The list of friend UserIDs. We recommend that you pull no more than 100 friends at a time to avoid response packet failure due to excessive data volume.
TagList	Array	Yes	This specifies the following profile fields and friend fields to be pulled: <a href="#">Standard profile fields</a> <a href="#">Custom profile fields</a> <a href="#">Standard friend fields</a> <a href="#">Custom friend fields</a>

Sample response

Response to a basic or complete request

```
{
  "InfoItem": [
    {
      "To_Account": "UserID_2",
      "SnsProfileItem": [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark_2"
        },
        {
          "Tag": "Tag_SNS_IM_Group",
          "Value": ["group1", "group2"]
        },
        {
          "Tag": "Tag_Profile_IM_Nick",
          "Value": "nick_2"
        },
        {
          "Tag": "Tag_SNS_Custom_Test",
          "Value": "custom_sns_2"
        },
        {
          "Tag": "Tag_Profile_Custom_Test",
          "Value": "custom_profile_2"
        }
      ]
    }
  ]
}
```

```

    ],
    "ResultCode": 0,
    "ResultInfo": ""
  }
],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}

```

### Pulling specified friends in batches

```

{
  "InfoItem": [
    {
      "To_Account": "UserID_2",
      "SnsProfileItem": [
        {
          "Tag": "Tag_SNS_IM_Remark",
          "Value": "remark_2"
        },
        {
          "Tag": "Tag_SNS_IM_Group",
          "Value": ["group1", "group2"]
        },
        {
          "Tag": "Tag_Profile_IM_Nick",
          "Value": "nick_2"
        },
        {
          "Tag": "Tag_SNS_Custom_Test",
          "Value": "custom_sns_2"
        },
        {
          "Tag": "Tag_Profile_Custom_Test",
          "Value": "custom_profile_2"
        }
      ],
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "UserID_3",
      "SnsProfileItem": [
        {
          "Tag": "Tag_SNS_IM_Remark",

```

```
        "Value": "remark_3"
      },
      {
        "Tag": "Tag_Profile_IM_Nick",
        "Value": "nick_3"
      },
      {
        "Tag": "Tag_Profile_Custom_Test",
        "Value": "custom_profile_3"
      }
    ],
    "ResultCode": 0,
    "ResultInfo": ""
  },
  {
    "To_Account": "UserID_4",
    "ResultCode": 30001,
    "ResultInfo": "Err_SNS_FriendGetList_Friend_Not_Exist"
  }
],
"Fail_Account": [
  "UserID_4"
],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

Response fields

Field	Type	Description
Infotem	Array	Friend object array. Each friend object contains a <code>To_Account</code> field and a <code>SnsProfileItem</code> array.
To_Account	String	UserID of a friend
SnsProfileItem	Array	Array for storing friend data. Each array element contains a <code>Tag</code> field and a <code>Value</code> field.
Tag	String	Field name
Value	String/Integer/Array	Field value
ResultCode	Integer	The result of <code>To_Account</code> . <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .

ResultInfo	String	The error description of <code>To_Account</code> . This field is empty if the request succeeds.
Fail_Account	Array	List of UserIDs that failed to be verified. This field is only returned when at least one user fails.
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[Adding Friends](#)

[Importing Friends](#)

[Updating Friends](#)

[Deleting Friends](#)

[Deleting All Friends](#)

[Verifying Friends](#)

[Pulling Friends](#)

# Friend Lists

## Adding Lists

Last updated : 2024-09-25 15:13:30

### Feature Overview

This API is used to add a friend list. You can add friend lists in batches and add a specific friend to a friend list.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/sns/group_add?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/sns/group_add	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.



contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum call frequency

200 calls per second

Sample request

Basic format

```
{
  "From_Account": "id",
  "GroupName": ["group1", "group2", "group3"]
}
```

Complete format

```
{
  "From_Account": "id",
  "GroupName": ["group1", "group2", "group3"],
  "To_Account": ["id1", "id2", "id3"]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	UserID for which a new friend list needs to be added.
GroupName	Array	Yes	List of new group lists. <b>Note :</b> Since the maximum number of groups is 32, a single request must not exceed 32 groups.
To_Account	Array	No	UserID list of friends that need to be added to the new group list; a single request must not exceed 1000 friends.

Sample response

Basic format

```
{
  "CurrentSequence": 2,
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
}
```

```
"ErrorDisplay": ""
}
```

Complete format

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "ResultCode": 32216,
      "ResultInfo": "Err_SNS_GroupAdd_ToTinyId_Not_Friend"
    },
    {
      "To_Account": "id3",
      "ResultCode": 30002,
      "ResultInfo": "ERR_SDKAPPID_ILLEGAL"
    }
  ],
  "Fail_Account":["id2","id3"],
  "CurrentSequence": 3,
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response fields

Field	Type	Description
ResultItem	Array	Result object array for adding a friend to a new friend list.
To_Account	String	UserID of the friend to be added to the new friend list.
ResultCode	Integer	Result of <code>To_Account</code> . <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	Error description of <code>To_Account</code> . This field is empty if the request is successful.
Fail_Account	Array	List of users whose processing failed. This field is only returned when at least

		one user fails.
CurrentSequence	Integer	Current sequence of the lists.
ActionStatus	String	Request result. <code>OK</code> : Successful; <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30002	The SDKAppID does not match.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.
30011	The maximum number of friend lists has been reached.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Deleting Lists ([v4/sns/group\\_delete](#))

# Deleting Lists

Last updated : 2024-09-25 15:13:30

## Feature Overview

This API is used to delete specified friend lists.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/group_delete?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code>
v4/sns/group_delete	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

### Maximum call frequency

200 calls per second

### Sample request

```
{
  "From_Account": "id",
  "GroupName": ["group1", "group2", "group3"]
}
```

### Request fields

Field	Type	Required	Description
From_Account	String	Yes	<code>UserID</code> of the account for which to delete lists.
GroupName	Array	Yes	The list of groups to be deleted. <b>Note :</b> Since the maximum number of groups is 32, the number of groups in a single request must not exceed 32.

### Sample response

```
{
  "CurrentSequence": 4,
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "0",
  "ErrorDisplay": ""
}
```

### Response fields

Field	Type	Description
CurrentSequence	Integer	The current sequence of the lists.
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed.
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information.

ErrorDisplay	String	Detailed information displayed on the client
--------------	--------	--

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.
30008	A write conflict occurred due to concurrent write operations. You are advised to use bulk processing.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

Adding Lists ([v4/sns/group\\_add](#))

# Pulling Lists

Last updated : 2024-09-25 15:13:30

## Feature Description

This API is used to pull friend lists. You can specify the lists to pull and pull the friends in the lists.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/group_get?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table only describes the modified parameters when this API is called. For more information on other parameters, please see [RESTful API Overview](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com India: adminapiind.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/group_get	Request API
sdkappid	SDKAppID assigned by the IM console when an app is created
identifier	App admin account. For more information, please see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .



usersig	Signature generated in the app admin account. For details on how to generate the signature, please see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format. The value is always <code>json</code> .

Maximum calling frequency

200 calls per second

Sample request packets

Basic request

```
{
  "From_Account": "id"
}
```

Complete request

```
{

  "From_Account": "id",

  "NeedFriend": "Need_Friend_Type_Yes",

  "GroupName": [

    "group1"

  ]

}
```

Request packet fields

Field	Type	Required	Description
From_Account	String	Yes	<code>UserID</code> of the account for which to pull friend lists
NeedFriend	String	No	Whether to pull the users in the list. <code>Need_Friend_Type_Yes</code> :

			pulls users. If this field is left empty, users will not be pulled. It is valid only when <code>GroupName</code> is left empty.
GroupName	Array	No	Name of the list to pull

Sample response packets

Response to a basic request

```
{
  "ResultItem": [
    {
      "GroupName": "group1",
      "FriendNumber": 1
    },
    {
      "GroupName": "group2",
      "FriendNumber": 2
    },
    {
      "GroupName": "group3",
      "FriendNumber": 3
    }
  ],
  "CurrentSequence": 2,
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
}
```

```
"ErrorDisplay": ""

}
```

Response to a complete request

```
{

  "ResultItem": [

    {

      "GroupName": "group1",

      "FriendNumber": 1,

      "To_Account": ["friend1"]

    }

  ],

  "CurrentSequence": 2,

  "ActionStatus": "OK",

  "ErrorCode": 0,

  "ErrorInfo": "",

  "ErrorDisplay": ""

}
```

Response packet fields

Field	Type	Description
ResultItem	Array	Result object array of pulling lists
GroupName	String	List name
FriendNumber	Integer	Number of friends in the list

To_Account	Array	<code>UserID</code> of friends in the list
CurrentSequence	Integer	Current sequence of the lists
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful; other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[Adding Lists](#)

[Deleting Lists](#)

# Blocklist

## Blocklisting Users

Last updated : 2025-03-10 10:30:49

### API Description

This API is used to add one or multiple people to your blocklist.

#### Note

If users A and B are friends, either one adding the other to the blocklist removes them from each other's friend list.

If user A blocks user B, or vice versa, then neither of them can send a friend request to the other person.

If user B is on user A's blocklist and user A is also on user B's blocklist, then user A and B cannot start a conversation with each other.

-> If user B is on user A's blocklist, but user A is not on user B's blocklist, then user A can message user B but not the other way around.

### API Call Description

#### Sample request URL

```
https://xxxxxx/v4/sns/black_list_add?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The list below contains only the parameters commonly used when calling this API and their descriptions. For more parameters, see the [RESTful API Overview](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	<p>The country/region where your SDKAppID is located.</p> <p>China: <code>console.tim.qq.com</code></p> <p>Singapore: <code>adminapisgp.im.qcloud.com</code></p> <p>Seoul: <code>adminapikr.im.qcloud.com</code></p> <p>Frankfurt: <code>adminapiger.im.qcloud.com</code></p> <p>Silicon Valley: <code>adminapiusa.im.qcloud.com</code></p> <p>Jakarta: <code>adminapiidn.im.qcloud.com</code></p>

v4/sns/black_list_add	Request API
sdkappid	The SDKAppID assigned by the Chat console when the application is created.
identifier	The administrator account of the app. For more information, refer to <a href="#">App Administrator</a> .
usersig	The signature generated in the app administrator account. For details on how to generate the signature, refer to <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is always <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

```
{
  "From_Account": "id",
  "To_Account": ["id1", "id2", "id3"]
}
```

Request fields

Field	Type	Required	Description
From_Account	String	Yes	The UserID that initiates the blocklist request.
To_Account	Array	Yes	A list of UserIDs to be added to the blocklist. This array should not contain more than 1000 UserIDs.

Sample response

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
```

```
      "ResultCode":30001,
      "ResultInfo":"Err_SNS_BlackListAdd_Already_Exist"
    },
    {
      "To_Account":"id3",
      "ResultCode":30002,
      "ResultInfo":"Err_SNS_BlackListAdd_SdkAppId_Illegal"
    }
  ],
  "Fail_Account":["id2","id3"],
  "ActionStatus":"OK",
  "ErrorCode":0,
  "ErrorInfo":"",
  "ErrorDisplay":""
}
```

Response fields

Field	Type	Description
ResultItem	Array	The result of batch blocking, which is an array of UserIDs and corresponding results.
To_Account	String	The UserID that you requested to add to blocklist.
ResultCode	Integer	The result. <code>0</code> means success and other values mean failure. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	Error description. This field is empty when the request succeeds.
Fail_Account	Array	A list of users that failed to be added to the blocklist. This field is only returned when at least one user fails.
ActionStatus	String	The result of the request. <code>OK</code> means the request is successfully handled. <code>FAIL</code> means the request failed.
ErrorCode	Integer	Error code. <code>0</code> means success and other values mean failure. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed information displayed on the client.

Error Codes



Unless a network error (such as error 502) occurs, the returned HTTP status code for this API is always 200. The specific error code and details can be found in the response fields such as `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), see [Error Codes](#).

The list below contains only error codes specific to this API:

Error Code	Description
30001	Wrong request parameter. Check your request according to the error description.
30002	The SDKAppID does not match.
30003	The requested user account does not exist.
30004	The request requires app administrator permissions.
30006	Internal server error. Please try again.
30007	Request timed out. Please try again later.
30008	A write conflict has occurred due to concurrent write operations. We recommend that you use batch processing.
30013	The maximum number of blocked users has been reached.

## API Debugging Tool

Use the [online RESTful API debugging tool](#) to commission this API.

## See Also

[Deleting a user from the blocklist](#)

[Querying a blocklist](#)

[Verifying a blocklist](#)

## Possible Callback

[Callback after adding users to the blocklist](#)



# Unblocklisting Users

Last updated : 2025-03-10 10:30:48

## Feature Description

This API is used to remove one or multiple users from the blocklist.

## API Invocation Description

### Request URL example

```
https://xxxxxx/v4/sns/black_list_delete?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table lists and describes only the parameters to be modified when this API is invoked. For details on other parameters, see [RESTful API Overview](#).

Parameter	Description
https	The request protocol is HTTPS, and the request method is POST.
xxxxxx	The country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/black_list_delete	The Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	This must be the app admin account. For details, see <a href="#">App Admins</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	Enter a random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format. The value is always <code>json</code> .
-------------	---

Maximum invocation frequency

The maximum invocation frequency is 200 times per second.

Request packet example

```
{
  "From_Account": "id",
  "To_Account": ["id1", "id2", "id3"]
}
```

Request packet fields

Field	Type	Attribute	Description
From_Account	String	Required	The identifier for which specific users are to be removed from their blocklist.
To_Account	Array	Required	The list of identifiers to be removed from the blocklist. The number of To_Accounts in a single request cannot exceed 1,000.

Response packet example

```
{
  "ResultItem":
  [
    {
      "To_Account": "id1",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id3",
      "ResultCode": 30006,
      "ResultInfo": "Err_SNS_BlackListCheck_Check_Reverse_BlackList_Fail"
    }
  ],
  "Fail_Account": ["id3"],
}
```

```
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

## Response packet fields

Field	Type	Description
ResultItem	Array	The result object array for batch deletion from blocklist.
To_Account	String	The identifier you request to be removed from the blocklist.
ResultCode	Integer	The processing result of To_Account. 0: succeeded. Others: failed.
ResultInfo	String	Error description for To_Account. If the processing is successful, the field is empty.
Fail_Account	Array	The list of users that fails to be removed. This field is returned only when someone fails to be removed.
ActionStatus	String	The request packet processing result. OK: succeeded. FAIL: failed.
ErrorCode	Integer	The error code. 0: succeeded. Others: failed.
ErrorInfo	String	Detailed error information.
ErrorDisplay	String	Detailed error information to be displayed on the client.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packet represent the actual error code and error information, respectively.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API.

Error Code	Description
30001	A request parameter is incorrect. In this case, check request parameters based on the error description.
30003	The requested user account does not exist.
30004	The request requires app admin permissions.

30006	An internal server error occurred. In this case, try again.
30007	The network connection timed out. In this case, try again later.
30008	A write conflict occurred due to concurrent writes. In this case, we recommend that you use the batch mode.

## API Commissioning Tool

Use the [online commissioning tool for RESTful APIs](#) to commission this API.

## References

[Removing users from blocklist](#)

[Pulling a blocklist](#)

[Verifying a blocklist](#)

## Callback That May Be Triggered

[Callback after removing users from blocklist](#)

# Pulling a Blacklist

Last updated : 2024-09-25 15:13:30

## Feature Overview

This API is used to pull a full blacklist by page.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/black_list_get?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/sns/black_list_get	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be json .

### Maximum call frequency

200 calls per second

### Sample request

```
{
  "From_Account": "id",
  "StartIndex": 0,
  "MaxLimited": 30,
  "LastSequence": 12
}
```

### Request fields

Field	Type	Required	Description
From_Account	String	Yes	UserID whose blocklist is to be pulled
StartIndex	Integer	Yes	Starting point for the blocklist pull
MaxLimited	Integer	Yes	Maximum number of blocklisted users that can be pulled per page <b>Note :</b> Since the total number of blocklists is up to 1000, the number of blocklists per request must not exceed 1000.
LastSequence	Integer	Yes	<code>Seq</code> that the backend returns to the client in the last blocklist pull. The value is <code>0</code> for the first pull. (For a RESTful API, enter <code>0</code> .)

### Sample response

```
{
  "BlackListItem": [
    {
      "To_Account": "id1",
      "AddBlackTimeStamp": 1430000001
    },
    {
      "To_Account": "id2",
      "AddBlackTimeStamp": 1430000002
    }
  ],
  "StartIndex": 0,
  "CurruentSequence": 13,
  "ActionStatus": "OK",
}
```



```
"ErrorCode": 0,  
"ErrorInfo": "",  
"ErrorDisplay": ""  
}
```

## Response fields

Field	Type	Description
BlackListItem	Array	Blocklist object array. Each blocklist object contains a <code>To_Account</code> and an <code>AddBlackTimeStamp</code> .
To_Account	String	UserID in the blocklist
AddBlackTimeStamp	Integer	Blocklisted time
StartIndex	Integer	Starting position for the next page pull. <code>0</code> indicates that the entire blocklist has been pulled.
CurruentSequence	Integer	Latest <code>Seq</code> of the blocklist
ActionStatus	String	Request result. <code>OK</code> : successful; <code>FAIL</code> : failed
ErrorCode	Integer	Error code. <code>0</code> : successful. Other values: failed. For details on non-zero results, please see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed information about the error.
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.

30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

- [Blocklisting users \(v4/sns/black\\_list\\_add\)](#)
- [Unblocklisting users](#)
- [Verifying users on a blocklist](#)

# Verifying Users on a Blocklist

Last updated : 2024-07-16 11:25:50

## Feature Overview

Batch verifies users in the blocklist

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/sns/black_list_check?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/sns/black_list_check	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

```
{
  "From_Account": "id",
  "To_Account": ["id1", "id2", "id3", "id4", "id5"],
  "CheckType": "BlackCheckResult_Type_Both"
}
```

## Request fields

Field	Type	Required	Description
From_Account	String	Yes	UserID that requests blocklist verification
To_Account	Array	Yes	List of UserIDs for blocklist verification. This field supports up to 1,000 UserIDs per request.
CheckType	String	Yes	Verification mode. For more information, see <a href="#">Verifying a blocklist</a> .

## Sample response

```
{
  "BlackListCheckItem": [
    {
      "To_Account": "id1",
      "Relation": "BlackCheckResult_Type_BothWay",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id2",
      "Relation": "BlackCheckResult_Type_AWithB",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "id3",
      "Relation": "BlackCheckResult_Type_BWithA",
      "ResultCode": 0,
      "ResultInfo": ""
    }
  ]
}
```

```

        "To_Account": "id4",
        "Relation": "BlackCheckResult_Type_NO",
        "ResultCode": 0,
        "ResultInfo": ""
    },
    {
        "To_Account": "id5",
        "Relation": "BlackCheckResult_Type_NO",
        "ResultCode": 30007,
        "ResultInfo": "Err_SNS_BlackListCheck_Check_Reverse_BlackList_Fail"
    }
],
"Fail_Account": ["id5"],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}

```

## Response fields

Field	Type	Description
BlackListCheckItem	Array	Verification result object array
To_Account	String	UserID of the account that requests blocklist verification
Relation	String	The blocklist relationship between <code>To_Account</code> and <code>From_Account</code> upon successful verification. For details, see <a href="#">Verifying a blocklist</a> .
ResultCode	Integer	Result of <code>To_Account</code> . <code>0</code> : Successful. Other values: Failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ResultInfo	String	The error description of <code>To_Account</code> . This field is empty if the request succeeds.
Fail_Account	Array	The users that failed to be verified. This field is only returned when at least one user fails.
ActionStatus	String	The request result. <code>OK</code> : successful. <code>FAIL</code> : failed.
ErrorCode	Integer	Error code. <code>0</code> : successful. Other values: failed. For details on non-zero results, see <a href="#">Error Codes</a> .
ErrorInfo	String	Detailed error information
ErrorDisplay	String	Detailed information displayed on the client

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ResultCode` , `ResultInfo` , `ErrorCode` , and `ErrorInfo` .

For public error codes (60000 to 79999), please see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Incorrect request parameter. Check your request according to the error description.
30003	The requested account does not exist.
30004	The request requires app admin permissions.
30006	Internal server error. Try again.
30007	Network timeout. Try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

[Unblocklisting users](#)

[Pulling a blocklist](#)

[Verifying users on a blocklist](#)

# Following and Fans

## Following Users

Last updated : 2025-06-09 17:25:18

### Feature Description

Follow user Support batch following users

### API Call Instructions

#### Sample Request URL

```
https://xxxxxx/v4/follow/follow_add?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request Parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Dedicated domain name corresponding to the country/region where the SDKAppID is located: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Mumbai: adminapiind.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/follow/follow_add	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created.
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum Calling Frequency

200 times/second.

Sample Requests

```
{
  "From_Account": "UserID_001",
  "FollowItem": [
    {
      "To_Account": "UserID_002"
    },
    {
      "To_Account": "UserID_003"
    }
  ]
}
```

Request Fields

Field	Type	Attribute	Description
From_Account	String	Required	UserID of the user who initiated the follow operation.
FollowItem	Array	Required	Focus on struct objects.
To_Account	String	Required	UserID of the user to be followed, the number of To_Accounts in a single request shall not exceed 20.

Sample Response

```
{
  "ResultItem": [
    {
      "To_Account": "UserID_002",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "UserID_003",
```



```
        "ResultCode": 0,
        "ResultInfo": ""
    }
],
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

Response Fields

Field	Type	Description
ResultItem	Array	Array of batch follow result objects.
To_Account	String	UserID of the user who requested the follow operation.
ResultCode	Integer	Processing result of To_Account 0: Processing is successful. Non-zero: Processing failed.
ResultInfo	String	Error description information of To_Account, this field is empty when successful.
ActionStatus	String	Request processing result: OK: Processing is successful. FAIL: Processing failed.
ErrorCode	Integer	Error Code: 0: Processing is successful. Non-zero: Processing failed.
ErrorInfo	String	Detailed error information.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description

30001	Request parameter error. Please check the request parameters according to the error description.
30002	SDKAppID does not match.
30003	Requested user account does not exist.
30004	Request requires App administrator permissions.
30006	Server internal error, please try again.
30007	Network timeout. Please try again later.
32100	The number of follows for From_Account has reached the system upper limit.
32101	The fan count for To_Account has reached the system upper limit.
32102	The mutual follow count for From_Account has reached the system upper limit.
32103	The mutual follow count for To_Account has reached the system upper limit.

## Debugging Tool for API

Debug this interface using the [REST API Online Debugging Tool](#).

## References

Unfollow user ([v4/follow/follow\\_delete](#))

Check follow relationship ([v4/follow/follow\\_check](#))

Pull follow, fans, and mutual followers list ([v4/follow/follow\\_get](#))

Get user's follow count, fan count, and mutual follow count ([v4/follow/follow\\_get\\_info](#))

## Callback Triggering S

# Unfollow Users

Last updated : 2025-06-09 17:24:06

## Feature Description

Unfollow a user, support batch unfollowing.

## API Call Description

### Sample Request URL

```
https://xxxxxx/v4/follow/follow_delete?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The following table lists only the parameters involved in modification when calling this API and their descriptions. For more details about the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	Dedicated domain name corresponding to the country/region where the SDKAppID is located: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Mumbai: adminapiind.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/follow/follow_delete	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created.
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

Maximum Calling Frequency

200 times/second.

Sample Requests

```
{
  "From_Account": "UserID_001",
  "To_Account": ["UserID_002", "UserID_003"]
}
```

Request Fields

Field	Type	Attribute	Description
From_Account	String	Required	The UserID of the operating user who initiated the unfollow operation.
To_Account	Array	Required	The UserID of the user to be unfollowed. The number of To_Accounts per request should not exceed 20.

Sample response

```
{
  "ResultItem": [
    {
      "To_Account": "UserID_002",
      "ResultCode": 0,
      "ResultInfo": ""
    },
    {
      "To_Account": "UserID_003",
      "ResultCode": 0,
      "ResultInfo": ""
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

## Response Fields

Field	Type	Description
ResultItem	Array	Result object array of batch unfollowing
To_Account	String	The UserID of the user who requested to unfollow.
ResultCode	Integer	Processing result of To_Account: 0: Processing succeeded. Non-zero: Processing failed.
ResultInfo	String	Error description information of To_Account, this field is empty when successful.
ActionStatus	String	Request processing result: OK: Processing succeeded. FAIL: Processing failed.
ErrorCode	Integer	Error Code: 0: Processing succeeded. Non-zero: Processing failed.
ErrorInfo	String	Detailed error information.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Request parameter error. Please check the request parameters according to the error description.
30002	SDKAppID mismatch
30003	Requested user account does not exist
30004	request requires App administrator permissions
30006	Server internal error, please try again.
30007	Network timeout, please try again.

## Debugging Tool for API

Debug this interface using the [REST API Online Debugging Tool](#).

## References

Follow user ([v4/follow/follow\\_add](#))

Check follow relationship ([v4/follow/follow\\_check](#))

Pull follow, fans, and mutual followers list ([v4/follow/follow\\_get](#))

Get user's follow count, fan count, and mutual follow count ([v4/follow/follow\\_get\\_info](#))

## Callback Triggering Scenarios

[Unfollow callback](#)

# Get The Fans, Follows, And Mutual Followers Count

Last updated : 2025-06-09 17:26:27

## Feature Description

Paged pull of fans/follows/mutual followers list for designated users.

## API Call Description

### Sample Request URL

```
https://xxxxxx/v4/follow/follow_get?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The following table lists only the parameters involved in modification when calling this API and their descriptions. For more details about the parameters, please refer to [REST API Introduction](#).

Parameter	Description
xxxxxx	Dedicated domain name corresponding to the country/region where the SDKAppID is located: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Mumbai: adminapiind.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/follow/follow_get	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created.
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum Calling Frequency

200 times/second.

Sample Requests

```
{
  "From_Account": "UserID_001",
  "FollowType": 1,
  "StartCursor": "",
  "WantNum": 100,
}
```

Request Fields

Field	Type	Attribute	Description
From_Account	String	Required	User's UserID.
FollowType	Integer	Required	Type to be pulled 1 - Fan 2 - Follow 3 - Mutual follow
StartCursor	String	Required	Start position for paging: Empty on homepage, fill in the NextCursor field returned in the previous page's response for subsequent requests.
WantNum	Integer	Required	Number of entries to fetch: Default returns 100 entries per page, supports up to 1000 entries.

Sample response

```
{
  "FollowItem": [
    {
      "To_Account": "UserID_002",
      "FollowTime": 1702262118,
      "ProfileItem": [
        {
          "Tag": "Tag_Profile_IM_Nick",

```



```
        "Value": "NickNameTest2"
      },
      {
        "Tag": "Tag_Profile_IM_AllowType",
        "Value": "AllowType_Type_DenyAny"
      },
      {
        "Tag": "Tag_Profile_IM_SelfSignature",
        "Value": "I'm Test2"
      },
      {
        "Tag": "Tag_Profile_Custom_Test",
        "Value": "Custom Data2"
      }
    ],
    "profile_sequence": 123
  },
  {
    "To_Account": "UserID_003",
    "FollowTime": 1702262118,
    "ProfileItem": [
      {
        "Tag": "Tag_Profile_IM_Nick",
        "Value": "NickNameTest3"
      },
      {
        "Tag": "Tag_Profile_IM_AllowType",
        "Value": "AllowType_Type_AllowAny"
      },
      {
        "Tag": "Tag_Profile_IM_SelfSignature",
        "Value": "I'm Test3"
      },
      {
        "Tag": "Tag_Profile_Custom_Test",
        "Value": "Custom Data3"
      }
    ],
    "profile_sequence": 111
  }
],
"NextCursor": "",
"ActionStatus": "OK",
"ErrorCode": 0,
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

## Response Fields

Field	Type	Description
FollowItem	Array	Fans/Follows/Mutual Followers List.
To_Account	String	User's UserID of fans/followers/mutual followers
FollowTime	Integer	Focus on time.
NextCursor	String	Starting position of the next page. If empty, it indicates that all data has been pulled.
ActionStatus	String	Request processing result: OK: processing successful. FAIL: processing failure.
ErrorCode	Integer	Error Code: 0: processing successful. Non-zero: processing failure.
ErrorInfo	String	Detailed error information.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Request parameter error. Please check the request parameters according to the error description.
30002	SDKAppID does not match.
30003	Requested user account does not exist.
30004	Request requires App administrator permissions.
30006	Server internal error, please try again.
30007	Network timeout. Please try again later.

# API Debugging Tool

Debug this interface using the [REST API Online Debugging Tool](#).

## References

Follow user ([v4/follow/follow\\_add](#))

Unfollow user ([v4/follow/follow\\_delete](#))

Check follow relationship ([v4/follow/follow\\_check](#))

Get user's follow count, fan count, and mutual follow count ([v4/follow/follow\\_get\\_info](#))

# Checking Follow Relationship

Last updated : 2025-06-09 17:29:48

## Feature Description

Support batch checking follow relationship

## API Call Description

### Sample Request URL

```
https://xxxxxx/v4/follow/follow_check?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Dedicated domain name corresponding to the country/region where the SDKAppID is located: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Mumbai: adminapiind.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/follow/follow_check	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created.
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format, which should always be <code>json</code> .
-------------	--

### Maximum Calling Frequency

200 times/second.

### Sample Requests

```
{
  "From_Account": "UserID_001",
  "To_Account": [
    "UserID_002",
    "UserID_003"
  ]
}
```

### Request Fields

Field	Type	Attribute	Description
From_Account	String	Required	Request check the follow relationship between the user and To_Account.
To_Account	Array	Required	To-be-checked list of users, the number of To_Accounts per request should not exceed 100.

### Sample response

```
{
  "ResultItem": [
    {
      "To_Account": "UserID_002",
      "ResultCode": 0,
      "ResultInfo": "",
      "FollowType": 0
    },
    {
      "To_Account": "UserID_003",
      "ResultCode": 0,
      "ResultInfo": "",
      "FollowType": 0
    }
  ],
  "ActionStatus": "OK",
  "ErrorCode": 0,
}
```

```
"ErrorInfo": "",
"ErrorDisplay": ""
}
```

Response Fields

Field	Type	Description
ResultItem	Array	Verify the result object array.
To_Account	String	UserID of the user who requests the check.
ResultCode	Integer	Processing result of To_Account: 0: Processing is successful. Non-zero: Processing failed.
ResultInfo	String	Error information of To_Account, this field is empty when successful.
FollowType	Integer	Attention relationship between To_Account and From_Account when verification is successful: 0 - From_Account does not follow To_Account, and To_Account does not follow From_Account. 1 - From_Account does not follow To_Account, but To_Account follows From_Account. 2 - From_Account follows To_Account, but To_Account does not follow From_Account. 3 - From_Account follows To_Account, and To_Account also follows From_Account.
ActionStatus	String	Result of request processing OK: Processing is successful. FAIL: Processing failed.
ErrorCode	Integer	Error Code: 0: Processing is successful. Non-zero: Processing failed.
ErrorInfo	String	Detailed Error Information.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Request parameter error. Please check the request parameters according to the error description.
30002	SDKAppID does not match.
30003	Requested user account does not exist.
30004	Request requires App administrator permissions.
30006	Server internal error, please try again.
30007	Network timeout. Please try again later.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Follow user ([v4/follow/follow\\_add](#))

Unfollow user ([v4/follow/follow\\_delete](#))

Pull follow, fans, and mutual followers list ([v4/follow/follow\\_get](#))

Get user's follow count, fan count, and mutual follow count ([v4/follow/follow\\_get\\_info](#))

# Get The Fans, Follows, And Mutual Followers Count

Last updated : 2025-06-09 17:28:54

## Feature Description

Batch query the number of fans, followers, and mutual connections for designated users.

## API Call Description

### Sample Request URL

```
https://xxxxxx/v4/follow/follow_get_info?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request Parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Dedicated domain name corresponding to the country/region where the SDKAppID is located: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Mumbai: adminapiind.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/follow/follow_get_info	Request API.
sdkappid	SDKAppID assigned by the Chat console when an app is created.
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .



random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format, which should always be <code>json</code> .

Maximum Calling Frequency

200 times/second.

Sample Requests

```
{
  "From_Account": "UserID_001",
  "To_Account": ["UserID_001", "UserID_002", "UserID_003"]
}
```

Request Fields

Field	Type	Attribute	Description
From_Account	String	Required	UserID of the user who initiated the pull operation.
To_Account	Array	Required	User ID of users pending count, the number of To_Account in a single request should not exceed 20.

Sample response

```
{
  "FollowInfo": [
    {
      "To_Account": "UserID_001",
      "ResultCode": 0,
      "ResultInfo": "",
      "FollowerCount": 0,
      "FollowingCount": 2,
      "MutualFollowingCount": 0
    },
    {
      "To_Account": "UserID_002",
      "ResultCode": 0,
      "ResultInfo": "",
      "FollowerCount": 1,
      "FollowingCount": 0,
      "MutualFollowingCount": 0
    }
  ],
}
```

```
{
  "To_Account": "UserID_003",
  "ResultCode": 0,
  "ResultInfo": "",
  "FollowerCount": 1,
  "FollowingCount": 0,
  "MutualFollowingCount": 0
},
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "ErrorInfo": "",
  "ErrorDisplay": ""
}
```

Response Fields

Field	Type	Description
FollowInfo	Array	Pulled result object array.
To_Account	String	User's UserID.
ResultCode	Integer	Processing result of To_Account 0: Processing is successful. Non-zero: Processing failed.
ResultInfo	String	Error description of To_Account, this field is empty when successful.
FollowerCount	Integer	Fan count of To_Account.
FollowingCount	Integer	Follow count of To_Account.
MutualFollowingCount	Integer	Mutual follow count of To_Account.
ActionStatus	String	Request processing result: OK: Processing is successful. FAIL: Processing failed.
ErrorCode	Integer	Error Code: 0: Processing is successful. Non-zero: Processing failed.
ErrorInfo	String	Detailed Error Information.

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
30001	Request parameter error. Please check the request parameters according to the error description.
30002	SDKAppID does not match.
30003	Requested user account does not exist.
30004	Request requires App administrator permissions.
30006	Server internal error, please try again.
30007	Network timeout. Please try again later.

## Debugging Tool for APIs

Debug this interface using the [REST API Online Debugging Tool](#).

## References

Follow user ([v4/follow/follow\\_add](#))

Unfollow user ([v4/follow/follow\\_delete](#))

Check follow relationship ([v4/follow/follow\\_check](#))

Pull follow, fans, and mutual followers list ([v4/follow/follow\\_get](#))

# Global Mute Management

## Setting Global Mute

Last updated : 2024-07-16 11:29:54

### Background

Global muting includes the global muting of one-to-one messages and that of group messages.

If the global muting of one-to-one messages is enabled for an account, during the muting period, no one-to-one messages can be sent, the sender receives error code 20012 (for details, see [Error Codes](#)), and [Before a One-to-One Message Is Sent](#) webhook is not triggered. After the muting period expires, the Chat backend system automatically disables the global muting of one-to-one messages, and then all one-to-one messages can be sent normally. For the permanent global muting of one-to-one messages, the muting period never expires. By default, the global muting of one-to-one messages is disabled for accounts.

If the global muting of group messages is enabled for an account, during the muting period, no group messages can be sent, the sender receives error code 10017 (for details, see [Error Codes](#)), and [Before Group Message Is Sent](#) webhook is not triggered. After the muting period expires, the Chat backend system automatically disables the global muting of group messages, and then all group messages can be sent normally. For the permanent global muting of group messages, the muting period never expires. By default, the global muting of group messages is disabled for accounts.

### Feature Description

This API is used to enable the global muting of one-to-one messages for an account.

This API is used to enable the global muting of group messages for an account.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/openconfigsvr/setnospeaking?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/openconfigsvr/setnospeaking	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to json .

### Maximum call frequency

200 calls per second

### Sample request

```
{
  "Set_Account": "lumotuwe",
  "C2CmsgNospeakingTime": 4294967295, // `C2CmsgNospeakingTime` and
  `GroupmsgNospeakingTime` are optional, but either of them must be specified.
  "GroupmsgNospeakingTime": 7200
}
```

### Request fields

Field	Type	Required	Description
Set_Account	String	Required	Account for which muting will be enabled

C2CmsgNospeakingTime	Integer	Optional	Muting period (in seconds) of one-to-one messages. The value is a non-negative integer, and the maximum value is 4294967295 (0xFFFFFFFF in hexadecimal format). 0: the account is unmuted 4294967295: the account is permanently muted Other values: a specific muting period of the account
GroupmsgNospeakingTime	Integer	Optional	Muting period (in seconds) of group messages. The value is a non-negative integer, and the maximum value is 4294967295 (0xFFFFFFFF in hexadecimal format). 0: the account is unmuted 4294967295: the account is permanently muted Other values: a specific muting period of the account

Sample response

```
{
  "ErrorCode": 0,
  "ErrorInfo": "",
}
```

Response fields

Field	Type	Description
ErrorCode	Integer	Request error code. 0 : Successful; other values: Failed
ErrorInfo	String	Error information

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
130001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.

130004	The JSON request does not contain the <code>Set_Account</code> field.
130005	The <code>Set_Account</code> field in the JSON request is invalid.
130008	Neither <code>GroupmsgNospeakingTime</code> nor <code>C2CmsgNospeakingTime</code> in the JSON request is specified.
130014	A system error occurred. Try again or contact technical support.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Querying global mute ([v4/openconfigsvr/setnospeaking](#))

# Querying Global Mute

Last updated : 2024-07-16 11:29:54

## Background

Global muting includes the global muting of one-to-one messages and that of group messages.

By default, the global muting of one-to-one messages is disabled for accounts. If the global muting of one-to-one messages is enabled for an account, all one-to-one messages cannot be sent during the muting period. After the muting period expires, the Chat backend system automatically disables the global muting of one-to-one messages, and then all one-to-one messages can be sent normally. For the permanent global muting of one-to-one messages, the muting period never expires.

By default, the global muting of group messages is disabled for accounts. If the global muting of group messages is enabled for an account, all group messages cannot be sent during the muting period. After the muting period expires, the Chat backend system automatically disables the global muting of group messages, and then all group messages can be sent normally. For the permanent global muting of group messages, the muting period never expires.

## Feature Description

This API is used to query the global muting of one-to-one messages for an account.

This API is used to query the global muting of group messages for an account.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/openconfigsvr/getnospeaking?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located.



	China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openconfigsvr/getnospeaking	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

200 calls per second

## Sample request

```
{
  "Get_Account": "lumotuwe"
}
```

## Request fields

Field	Type	Required	Description
Get_Account	String	Required	Account for which muting information is queried

## Sample response

```
{
  "ErrorCode": 0,
  "ErrorInfo": "",
  "C2CmsgNospeakingTime": 4294967295,
  "GroupmsgNospeakingTime": 7196
}
```

Response fields

Field	Type	Description
ErrorCode	Integer	Request error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information
C2CmsgNospeakingTime	Number	The muting period for one-to-one messages, in seconds. The value is a non-negative integer. The value <code>0</code> indicates that message muting is disabled. The maximum value <code>4294967295</code> (or <code>0xFFFFFFFF</code> in hexadecimal) indicates that permanent muting is enabled for the account. Other values indicate the specific muting period of the account. For example, the value <code>3600</code> indicates that the muting period of the account is one hour.
GroupmsgNospeakingTime	Number	The muting period for group messages, in seconds. The value is a non-negative integer. The value <code>0</code> indicates that message muting is disabled. The maximum value <code>4294967295</code> (or <code>0xFFFFFFFF</code> in hexadecimal) indicates that permanent muting is enabled for the account. Other values indicate the specific muting period of the account. For example, the value <code>3600</code> indicates that the muting period of the account is one hour.

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
130001	Failed to parse the JSON format. Check whether the JSON request meets JSON specifications.
130002	The JSON request does not contain the <code>Get_Account</code> field.
130003	The <code>Get_Account</code> field in the JSON request is invalid.
130014	A JSON system error occurred. Try again or contact technical support.

# API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Setting global mute ([v4/openconfigsvr/setnospeaking](#))

# Operations Management

## Pulling Operations Data

Last updated : 2025-01-14 14:47:52

### Feature Overview

The app admin can pull operations data for the last 30 days through this API. [The operational data fields that can be pulled](#) are described later in this document.

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/openconfigsvr/getappinfo?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/openconfigsvr/getappinfo	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating</a>

	UserSig.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

Query the operations data of the last 30 days for SDKAppID.

Basic format

Pull all fields by default.

```
{}
```

Specifying fields to be pulled

Specify fields to be pulled in `RequestField`.

```
{
  "RequestField": [
    "ChainIncrease",
    "ChainDecrease"
  ]
}
```

Request fields

Field	Type	Required	Description
RequestField	Array	No	This field is used to specify the operations data fields to be pulled. If this field is not specified, all fields will be pulled by default. For details, see <a href="#">the operations data fields that can be pulled</a> below.

Sample response

Basic format

```
{
  "ErrorCode": 0,
  "ErrorInfo": "OK",
  "Result": [{
    "APNSMsgNum": "84",
    "ActiveUserNum": "2014",
```

```

    "AppId": "1104620500",
    "AppName": "Real-Time Communication Scenario Free trial",
    "C2CAPNSMsgNum": "84",
    "C2CDownMsgNum": "11040",
    "C2CSendMsgUserNum": "9",
    "C2CUpMsgNum": "52209",
    "CallBackReq": "73069",
    "CallBackRsp": "72902",
    "ChainDecrease": "16",
    "ChainIncrease": "18",
    "Company": "LinYE",
    "Date": "20160607",
    "DownMsgNum": "11869",
    "GroupAPNSMsgNum": "0",
    "GroupAllGroupNum": "41913",
    "GroupDestroyGroupNum": "35019",
    "GroupDownMsgNum": "829",
    "GroupJoinGroupTimes": "121438",
    "GroupNewGroupNum": "35904",
    "GroupQuitGroupTimes": "108292",
    "GroupSendMsgGroupNum": "5189",
    "GroupSendMsgUserNum": "12",
    "GroupUpMsgNum": "8433",
    "LoginTimes": "13708",
    "LoginUserNum": "2094",
    "MaxOnlineNum": "62",
    "RegistUserNumOneDay": "1052",
    "RegistUserNumTotal": "53091",
    "SendMsgUserNum": "19",
    "UpMsgNum": "60642",
  }
}

```

### Specifying fields to be pulled

```

{
  "ErrorCode": 0,
  "ErrorInfo": "OK",
  "Result": [{
    "ChainDecrease": "8",
    "ChainIncrease": "8",
    "Date": "20160605"
  },
  {
    "ChainDecrease": "17",
    "ChainIncrease": "17",
    "Date": "20160604"
  }
]
}

```

```
}
]
}
```

## Response fields

Field	Type	Description
Result	Array	The requested operations data from the last 30 days
ErrorCode	Integer	Error code. 0 : Successful; other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
130001	JSON parsing error of the request
130009	SQL opening error
130010	SQL pinging error
130011	SQL query error
130012	SQL result parsing error

## Operations Data Fields That Can Be Pulled

Field	Description
AppName	App name
AppId	SDKAppID of the app

Company	Customer name
ActiveUserNum	Number of active users
RegistUserNumOneDay	Number of newly registered users
RegistUserNumTotal	Total number of registered users
LoginTimes	Number of logins
LoginUserNum	Number of logged-in users
UpMsgNum	Number of upstream messages
SendMsgUserNum	Number of message sending users
APNSMsgNum	Number of pushed APNs messages
C2CUpMsgNum	Number of upstream messages (C2C)
C2CDownMsgNum	Number of downstream messages (C2C)
C2CSendMsgUserNum	Number of message sending users (C2C)
C2CAPNSMsgNum	Number of pushed APNs messages (C2C)
MaxOnlineNum	Maximum number of online users
DownMsgNum	Total number of downstream messages (C2C and group)
ChainIncrease	Increase in relationship chain pairs
ChainDecrease	Decrease in relationship chain pairs
GroupUpMsgNum	Number of upstream messages (group)
GroupDownMsgNum	Number of downstream messages (group)
GroupSendMsgUserNum	Number of message sending users (group)
GroupAPNSMsgNum	Number of pushed APNs messages (group)
GroupSendMsgGroupNum	Number of message sending groups
GroupJoinGroupTimes	Total number of joined groups
GroupQuitGroupTimes	Total number of left groups
GroupNewGroupNum	Number of newly added groups



GroupAllGroupNum	Total number of groups
GroupDestroyGroupNum	Number of disbanded groups
CallBackReq	Number of callback requests
CallBackRsp	Number of callback responses
Date	Date

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Downloading Recent Messages

Last updated : 2024-07-16 11:29:54

## Feature Overview

The app admin can use this API to obtain the download addresses for all one-to-one or group message records in the app that occur at a specified point in time for the last seven days.

### Note

You can download images, audio, files, and short videos from message records. This feature is only applicable to Chat SDK 4.X or later. The download can be performed based on the URL fields in chat records. If you are using Chat SDK 2.X or 3.X, you cannot obtain the preceding information in this method. If you need this feature, upgrade your Chat SDK to version 4.X or later.

Message records are stored as logs and compressed by using GZip. After obtaining the download addresses through the API, you can download and process the message records yourself. Message record files are generated every hour according to Beijing time. For example, the data generated at midnight (00:00-00:59) will be processed from 01:00. Typically, the data can be processed within one hour. However, if the message quantity is large, it will take longer to process them. The message record files are valid for only seven days and will be deleted after seven days regardless of the download status. Deleted records cannot be exported again. The download address obtained through this API has an expiration date. Please download the message records before the download address expires. If the download address becomes invalid, obtain the download address again through this API.

This API is used only to download historical chat records for the last seven days for backup, statistics, or other purposes. We do not recommend that you use it for real-time online businesses.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/open_msg_svc/get_history?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description

xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code>
v4/open_msg_svc/get_history	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Request format. The value is fixed to <code>json</code> .

## Maximum call frequency

10 times/second

## Sample request

```
{
  "ChatType": "C2C",
  "MsgTime": "2015120121"
}
```

## Request fields

Field	Type	Required	Description
ChatType	String	Required	Message type. <code>C2C</code> indicates a one-to-one message, whereas <code>Group</code> indicates a group message.
MsgTime	String	Required	Period for downloading message records. MsgTime uses Beijing time and does not support other time zones. For example, 2015120121 indicates that messages in the period of 21:00 to 21:59 on December 1, 2015 will be downloaded. This field needs to be specified by a hour. Each request can only be used to obtain all one-to-one or group message records that occur at the specified hour on the specific day.

Sample response

```
{
  "File": [
    {
      "URL":
"https://download.tim.qq.com/msg_history/2/9b8f8f063b73f61698ce11e58207e89ade40
.gz",
      "ExpireTime": "2015-12-02 16:45:23",
      "FileSize": 65207,
      "FileMD5": "cceece008bb7f469a47cf8c4b7acb84e",
      "GzipSize": 1815,
      "GzipMD5": "c3a0269dde393fd7a8bb18bfdeae2e"
    }
  ],
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	The processing result of the request. OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code. 0 : Successful; other values: Failed
ErrorInfo	String	Error information
File	Array	Download information of the message record file
URL	String	Download address of the message record file
ExpireTime	String	Expiration time of the download address. Always download the file before the address expires. If the address expires, obtain a new address through the API.
FileSize	Integer	File size (in bytes) before GZip compression
FileMD5	String	File MD5 before GZip compression
GzipSize	Integer	File size (in bytes) after GZip compression
GzipMD5	String	File MD5 after GZip compression

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
1001	Invalid request. Check whether the request URL is correct.
1002	Invalid parameter. Check whether the account is the admin, required fields are specified, and the values meet protocol requirements.
1003	System error.
1004	The file has not been generated yet, or no message is delivered in the requested period.
1005	File expired.

## Format of Message Record Files

```
//For one-to-one messages
{"SdkAppId":1104620500,"ChatType":"C2C","MsgTime":"2015120121","MsgList":[
  {"From_Account":"peakerdong","To_Account":"qiyuelihuo2018","MsgTimestamp":1448
  974806,"MsgSeq":3452069198,"MsgRandom":45838,"MsgBody":
  [{"MsgType":"TIMTextElem","MsgContent":{"Text":"Quartering"}}]},
  {"From_Account":"group_root","To_Account":"group_test4","MsgTimestamp":14489748
  08,"MsgSeq":462709847,"MsgRandom":19196437,"MsgBody":
  [{"MsgType":"TIMTextElem","MsgContent":{"Text":"hi, beauty"}}]}
]}

//For group messages
{"SdkAppId":1104620500,"ChatType":"Group","MsgTime":"2015120121","MsgList":[
  {"From_Account":"Test_1","GroupId":"@TGS#1FDFVPAE2","MsgTimestamp":1448975384,"
  MsgSeq":1,"MsgBody":[{"MsgType":"TIMTextElem","MsgContent":{"Text":"Private
  activate"}}]},
  {"From_Account":"Test_1","GroupId":"@TGS#1FDFVPAE2","MsgTimestamp":1448975384,"
  MsgSeq":1,"MsgBody":[{"MsgType":"TIMTextElem","MsgContent":{"Text":"Private
  activate"}}]}
]}
```

The first line of the file records basic information about the file. Each following line records a message until the last line that ends with "]]". For the format of each message, see the definitions in [TIMMsgElement Objects](#).

If the file is small, you can use the JSON database to parse the entire file. `MsgList` indicates the message array for the specified period. For example:

```
# Python sample code
import gzip, json
with gzip.open('1104620500_Group_2015120121.gz', 'rb') as fp:
    info = json.load(fp)
    for msg in info['MsgList']:
        pass #do sth with msg
```

If the file is large, we recommend that you parse it line by line. For example:

```
# Python sample code
import gzip, json
with gzip.open('1104620500_Group_2015120121.gz', 'rb') as fp:
    cnt = -1
    for line in fp:
        line = line.strip().rstrip(b',')
        if line == b']]': break
        if cnt < 0:
            info = json.loads(line + b']]')
        else:
            msg = json.loads(line)
            #do sth with msg
        cnt += 1
```

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Getting Server IP Addresses

Last updated : 2024-07-16 11:29:54

## Feature Overview

Sometimes, you may need to obtain the list of server IP addresses for security purposes. The app admin can use this API to obtain the list of server IP addresses or IP ranges used by the SDK and webhook.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/ConfigSvc/GetIPList?  
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json  
&nettype=0
```

### Request parameters

The following table describes the modified parameters when this API is called. For other parameters, see [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/ConfigSvc/GetIPList	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.

contenttype	Request format. The value is fixed to <code>json</code> .
nettype	Request a list of server IP addresses of a specified type: nettype=0: List of all server IP addresses nettype=1: List of the server IP addresses accessed by the native SDK inside China (including Hongkong, China) nettype=2: List of the server IP addresses accessed by the native SDK outside China nettype=3: List of the server IP addresses accessed by the web SDK inside China (including Hongkong, China) nettype=4: List of the server IP addresses accessed by the web SDK outside China nettype=5: List of egress addresses used by Chat for webhooks inside China (including Hongkong, China) nettype=6: List of egress addresses used by Chat for webhooks outside China

Note

For the webhook business, the business server needs to add the Chat service egress addresses to the allowlist. You can set `nettype` to 5 (or 6) according to the business server address region to get the list of Chat service egress addresses.

Maximum call frequency

200 calls per second

Sample request

```
{ }
```

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorCode": 0,
  "IPList": [ "127.0.0.2", "183.192.202.0/25" ]
}
```

Response fields

Field	Type	Description
IPList	Array	List of server IP addresses
ErrorCode	Integer	Error code. <code>0</code> : Successful; other values: Failed
ErrorInfo	String	Error information



## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#).

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

# Chatbots

## Pulling Chatbot Accounts

Last updated : 2024-07-16 11:29:54

### Feature Overview

This API is used to pull the list of all chatbot accounts. A chatbot account is a special account whose `userid` must start with `@RBT#` .

### API Calling Description

#### Sample request URL

```
https://xxxxxx/v4/openim_robot_http_svc/get_all_robots?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

#### Request parameters

The following table lists the parameters used when this API is called and their descriptions. For other parameters, see the [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim_robot_http_svc/get_all_robots	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .
usersig	Signature generated by the app admin account. For details, see

	Generating UserSig.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second.

Sample request

```
{ }
```

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Robot_Account": [
    "@RBT#001",
    "@RBT#002",
    "@RBT#003"
  ]
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information
Robot_Account	Array	List of bot UserIDs

Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively.

For public error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal error. Try again.
10008	Invalid request. For example, the request requires app admin permissions or the number of created chatbots reaches the upper limit.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Querying Accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Setting Profiles ([v4/profile/portrait\\_set](#))

# Creating Chatbot Accounts

Last updated : 2025-06-09 10:06:15

## Feature Overview

This API is used to create a chatbot account. A chatbot account is a special account whose `userid` must start with `@RBT#`. Users can set the nicknames, profile photos, and statuses for chatbot accounts when creating them.

### Note:

Repeatedly calling this API with the same chatbot `userid` will create only one chatbot account.

Each Chat account can create up to 20 chatbot accounts.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/openim_robot_http_svc/create_robot?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table lists the parameters used when this API is called and their descriptions. For other parameters, see the [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim_robot_http_svc/create_robot	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .

usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

```
{
  "UserID": "@RBT#001",
  "Nick": "Jarvis",
  "FaceUrl": "www.example.com/head.jpg",
  "SelfSignature": "hello, world"
}
```

Request fields

Field	Type	Required	Description
UserID	String	Yes	Chatbot username, which is a string of up to 32 bytes in length and must start with <code>@RBT#</code> .
Nick	String	No	Chatbot nickname
FaceUrl	String	No	URL of the bot's profile photo
SelfSignature	String	No	Status of the chatbot, which is up to 500 bytes.

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
-------	------	-------------

ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#). The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal error. Try again.
10004	Invalid parameter. Check whether the required fields are all set and whether the parameter settings meet the protocol requirements.
10008	Invalid request. For example, the request requires app admin permissions or the number of created chatbots reaches the upper limit.
40006	An internal server error occurred during profile setting. Try again later.
40601	The value of the profile field exceeds the length limit of 500 bytes.
70398	The number of accounts exceeds the limit allowed. To create more than 100 accounts, upgrade your app. For upgrade instructions, see <a href="#">Purchase Guide</a> .

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Querying Accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Setting Profiles ([v4/profile/portrait\\_set](#))

# Deleting Chatbot Accounts

Last updated : 2024-07-16 11:29:54

## Feature Overview

This API is used to delete chatbots. A chatbot account is a special account whose `userid` must start with `@RBT#` .

### Note:

This API can set a bot account to invalid.  
The UserID of the bot account will not be deleted.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/openim_robot_http_svc/delete_robot?
sdkappid=88888888&identifier=admin&usersig=xxx&random=99999999&contenttype=json
```

### Request parameters

The following table lists the parameters used when this API is called and their descriptions. For other parameters, see the [RESTful API Overview](#).

Parameter	Description
xxxxxx	Domain name corresponding to the country/region where your SDKAppID is located. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/openim_robot_http_svc/delete_robot	Request API
sdkappid	SDKAppID assigned by the Chat console when an app is created
identifier	App admin account. For more information, see the <b>App Admin</b> section in <a href="#">Login Authentication</a> .



usersig	Signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295
contenttype	Request format, which should always be <code>json</code> .

Maximum call frequency

200 calls per second

Sample request

```
{
  "Robot_Account": "@RBT#001"
}
```

Request fields

Field	Type	Required	Description
Robot_Account	String	Yes	Chatbot username, which is a string of up to 32 bytes in length and must start with <code>@RBT#</code> .

Sample response

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response fields

Field	Type	Description
ActionStatus	String	Request result. <code>OK</code> : Successful. <code>FAIL</code> : Failed
ErrorCode	Integer	Error code. <code>0</code> : Successful. Other values: Failed
ErrorInfo	String	Error information

## Error Codes

The returned HTTP status code for this API is always 200 unless a network error (such as error 502) occurs. The specific error code and details can be found in the response fields `ErrorCode` and `ErrorInfo` respectively. For public error codes (60000 to 79999), see [Error Codes](#). The following table describes the error codes specific to this API:

Error Code	Description
10002	Internal error. Try again.
10004	Invalid parameter. Check whether the required fields are all set and whether the parameter settings meet the protocol requirements.
10008	Invalid request. For example, the request requires app admin permissions or the number of created chatbots reaches the upper limit.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## References

Querying Accounts ([v4/im\\_open\\_login\\_svc/account\\_check](#))

Setting Profiles ([v4/profile/portrait\\_set](#))

# Official Channel Management

## Official Channel Related

### Creating Official Channel

Last updated : 2025-05-27 10:12:36

## Feature Overview

This API is used by the app administrator to create an official channel.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/create_official_account?sdkappid=8
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/create_official_account	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .

usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{
  "OfficialAccountUserID": "@TOA#_test_id", // The prefix of the custom official
  "Owner_Account": "107867",
  "CustomString": "test_custom_string",
  "FaceUrl": "http://this.is.face.url",
  "Introduction": "This is test Introduction",
  "MaxSubscriberNum": 1000,
  "Name": "test_official_account",
  "Organization": "test_organization"
}
```

Request Fields

Field	Type	Required	Description
Owner_Account	String	Yes	The account ID that owns the official channel (must be an <a href="#">imported</a> account), representing the owner of the official channel.
OfficialAccountUserID	String	No	ID of the official channel. In order to make the official channel ID simpler and easier to remember and spread, Tencent Cloud supports <a href="#">customizing the official channel ID</a> when the app creates an official channel through the RESTful API. If not specified, the system will generate an official channel ID by default and return it.
Name	String	Yes	Name of the official channel, up to 150 bytes in length, using UTF-8 encoding, with 1 Chinese character occupying 3 bytes.
Introduction	String	No	Introduction to the official channel, up to 400 bytes in

			length, using UTF-8 encoding, with 1 Chinese character occupying 3 bytes.
FaceUrl	String	No	Avatar of the official channel, up to 500 bytes in length.
MaxSubscriberNum	Integer	No	The maximum number of subscribers to an official channel, with a default value of 100,000 when not specified.
Organization	String	No	Group organization the official channel belongs to, up to 500 bytes in length, using UTF-8 encoding, with 1 Chinese character occupying 3 bytes.
CustomString	String	No	The custom field of the official channel, up to 3,000 bytes. It can be used by the business layer to meet the requirements of special scenarios.

## Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "OfficialAccountUserID": "@TOA#_test_id"
}
```

## Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.
OfficialAccountUserID	String	The official channel ID after successful creation. The ID defined by the user or assigned by the Chat backend is returned.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use. Please modify and retry.
131000	This AppId has not been enabled with official channel service. Please enable it and retry.
130010	The number of created official channels exceeds the limit. Please dissolve some official channels or upgrade the service.

## References

[Terminating Official Accounts](#)

## Possibly Triggered Callbacks

[Before an Official Account Is Created](#)

[After an Official Account Is Created](#)

# Terminating Official Channel

Last updated : 2025-05-27 10:12:39

## Feature Overview

This API is used by the app administrator to terminate an official channel.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/destroy_official_account?sdkappid=
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/destroy_official_account	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{
  "Official_Account": "@TOA#_test_id"
}
```

Request Fields

Field	Type	Required	Description
Official_Account	String	Yes	The ID of the official channel to be terminated. Use the OfficialAccountUserID field returned by the API during creation.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.



ErrorInfo	String	Error information.
-----------	--------	--------------------

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel user does not exist, or once existed but has now been terminated.
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use. Please modify and retry.
131000	This AppId has not been enabled with official channel service. Please enable it and retry.

## References

Creating Official Accounts ([v4/official\\_account\\_open\\_http\\_svc/create\\_official\\_account](#))

## Possibly Triggered Callbacks

[After an Official Account Is Terminated](#)

# Modifying Official Channel Information

Last updated : 2025-05-27 10:12:45

## Feature Overview

This API is used by the app administrator to modify relevant information of an official channel, such as the name, avatar, and introduction.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/modify_official_account_base_info?
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/official_account_open_http_svc/modify_official_account_base_info	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is

	created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

### Maximum Call Frequency

200 times/second

### Sample Request Packets

```
{
  "Official_Account": "@TOA#_@TOA#dOEF",
  "CustomString": "modify_custom_string",
  "FaceUrl": "http://this.is.face.url",
  "Introduction": "This is modify test Introduction",
  "Name": "test_modify_official_account",
  "Organization": "modify_organization",
  "MaxSubscriberNum":1000
}
```

### Request Fields

Field	Type	Required	Description
Official_Account	String	Yes	ID of the official channel to be modified. Use the OfficialAccountUserID field returned by the API during creation.
Name	String	No	Name of the official channel, up to 150 bytes in length, using UTF-8 encoding, with 1 Chinese character occupying 3 bytes.
Introduction	String	No	Introduction to the official channel, up to 400 bytes in length, using UTF-8 encoding, with 1 Chinese character occupying 3 bytes.
FaceUrl	String	No	Avatar of the official channel, up to 500 bytes in length.

Organization	String	No	Group organization the official channel belongs to, up to 500 bytes in length, using UTF-8 encoding, with 1 Chinese character occupying 3 bytes.
CustomString	String	No	The custom field of the official channel, up to 3,000 bytes. It can be used by the application layer to meet the requirements of special scenarios.
MaxSubscriberNum	Integer	No	The maximum number of subscribers to an official channel, with a default value of 100,000 when not specified.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.

Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
------------	-------------

6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel user does not exist, or once existed but has now been terminated.
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use. Please modify and retry.
131000	This AppId has not been enabled with official channel service. Please enable it and retry.

## References

Creating Official Accounts ([v4/official\\_account\\_open\\_http\\_svc/create\\_official\\_account](#))

## Possibly Triggered Callbacks

After Official Account Information Is Modified (<1>)

# Obtaining Official Channel Information

Last updated : 2025-05-27 10:16:51

## Feature Overview

This API is used by the app administrator to obtain relevant information of an official channel.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/get_official_account_info?sdkappid
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/get_official_account_info	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{
  "OfficialAccountIdList": [    // List of official channels requested, with a ma
    {
      "Official_Account": "@TOA#_test_for_penn"
    },
    {
      "Official_Account": "@TOA#_@TOA#dZDP"
    }
  ],
  "ResponseFilter": {          // Filter fields
    "OfficialAccountBaseInfoFilter": [    // Filter fields for official cha
      "CreateTime",
      "Name",
      "Owner_Account",
      "LastMsgTime",
      "SubscriberNum",
      "Introduction",
      "FaceUrl",
      "Organization",
      "CustomString"
    ]
  }
}
```

Request Fields

Field	Type	Required	Description
OfficialAccountIdList	Array	Yes	List of official channel IDs requested.
Official_Account	String	Yes	Official account ID requested. Use the OfficialAccountUserID field returned by the API during creation.
ResponseFilter	Object	No	Filter fields for the returned official channel information. If not specified, all fields are returned by default, including the

			<p>OfficialAccountBaseInfoFilter filter. The valid values include:</p> <p>CreateTime: creation time of the official channel</p> <p>Name: name of the official channel</p> <p>Owner_Account: user who owns the official channel</p> <p>LastMsgTime: time of the last message within the official channel</p> <p>SubscriberNum: current number of subscribers to the official channel</p> <p>Introduction: introduction to the official channel</p> <p>FaceUrl: avatar URL of the official channel</p> <p>Organization: organization of the official channel</p> <p>CustomString: custom field at the official channel level</p>
--	--	--	--

## Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,      // This ErrorCode indicates the result of the entire request
  "OfficialAccountInfoList": [  // List of official channel information returned
    {
      "ErrorCode": 0,      // This ErrorCode indicates the result of obtaining
      "ErrorInfo": "",
      "Name": "test_modify_official_account3",
      "Official_Account": "@TOA#_test_for_penn",
      "CreateTime": 1696577604,
      "Owner_Account": "107867",
      "LastMsgTime": 1699501664,
      "SubscriberNum": 780031,
      "Introduction": "test compare introduction",
      "FaceUrl": "http://this.is.face.url1",
      "Organization": "test compare organization",
      "CustomString": "modify_custom_string"
    },
    {
      "ErrorCode": 0,
      "ErrorInfo": "",
      "Name": "Test Environment Test Official Account",
      "Official_Account": "@TOA#_@TOA#dZDP",
      "CreateTime": 1694514910,
      "Owner_Account": "109635",
      "LastMsgTime": 1698894911,
      "SubscriberNum": 1004,
      "Introduction": "waynezhou",
      "FaceUrl": ""
    }
  ]
}
```



```
      "Organization": "Tencent",
      "CustomString": ""
    }
  ]
}
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.
OfficialAccountInfoList	Array	The returned official channel list field. The results are filtered information based on the filter fields set in the filter.

Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel does not exist, or once existed but has now been terminated.

130007	Incorrect official channel ID. Please modify and retry.
131000	This AppId has not been enabled with official channel service. Please enable it and retry.

## References

- [Creating Official Accounts](#)
- [Terminating Official Accounts](#)

# Obtaining Subscriber Information of Official Channel

Last updated : 2025-05-27 10:16:47

## Feature Overview

This API is used by the app administrator to obtain the information of all users subscribing to a specific official channel.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/get_subscriber_info?sdkappid=88888
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/official_account_open_http_svc/get_subscriber_info	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{  "Official_Account": "@TOA#_test_for_penn",
  "Limit": 50,
  "Next": ""
}
```

Request Fields

Field	Type	Required	Description
Official_Account	String	Yes	Official account ID requested. Use the OfficialAccountUserID field returned by the API during creation.
Limit	Integer	No	The maximum number of members' information that can be obtained at one time, not to exceed 200. If not specified, the information of all members in the group will be obtained.
Next	String	Yes	The position of the subscriber last pulled. For the first call, fill in ""; for subsequent pulls, use the Next value returned in the response.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "Next": "TGSOTk=",           // The marker for the subsequent pull. If not empty
  "SubscriberNum": 780031,    // Total number of subscribers to this official cha
  "SubscriberList": [
    {
      "Subscriber_Account": "millaccount13356",           // Subscriber accou
      "SubscribeTime": 1698821747,                       // Time of subscription
      "CustomString": "",                                // Custom field at
```

```
        "MsgFlag": "AcceptAndNotify",                // The message receiving op
        "LastSendMsgTime": 0                        // Time of the last
    },
    {
        "Subscriber_Account": "millaccount13100",
        "SubscribeTime": 1698821690,
        "CustomString": "",
        "MsgFlag": "AcceptAndNotify",
        "LastSendMsgTime": 0
    }
    // The rest of the subscriber information is omitted below.
]
}
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.
Next	String	The marker for the subsequent pull. The server response packet returns Next as "" for the last pull, indicating there is no subsequent subscriber information, and the client should end the query.
SubscriberNum	Integer	Returns the number of all subscribers under the official channel.
SubscriberList	Array	Returned subscriber list, including the following information: Subscriber_Account: subscriber account SubscribeTime: time of subscription CustomString: custom field at the subscriber level MsgFlag: the message receiving option for subscribers, similar to the message receiving option for group members LastSendMsgTime: time of the last message sent

Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use. Please modify and retry.
131000	This AppId has not been enabled with official channel service. Please enable it and retry.
130010	The number of created official channels exceeds the limit. Please dissolve some official channels or upgrade the service.

## References

[Creating Official Accounts](#)

# Subscriber Relevant

## Adding Subscribers

Last updated : 2025-05-27 10:16:40

### Feature Overview

The API is used by the app administrator to make a user subscribe to an official channel and become a subscriber of the official channel.

### API Calling Description

#### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/add_subscriber?sdkappid=88888888&i
```

#### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/add_subscriber	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{
  "Official_Account": "@TOA#_@TOA#dOEF",           // The desired official cha
  "SubscriberList": [                               // Up to 100 subscribers can be added at a time.
    {
      "Subscriber_Account": "107867"               // UserID of the subscriber to be a
    },
    {
      "Subscriber_Account": "966582476"
    }
  ]
}
```

Request Fields

Field	Type	Required	Description
Official_Account	String	Yes	ID of the official channel to be subscribed to. Use the OfficialAccountUserID field returned by the API during creation.
SubscriberList	Array	Yes	Array of subscribers to be added.
Subscriber_Account	String	Yes	UserID of the subscriber to be added.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "SubscriberList": [
    {
      "Subscriber_Account": "107867",
```



```
        "Result": 2
    },
    {
        "Subscriber_Account": "966582476",
        "Result": 1           // Addition result. 0: failed; 1: succeeded; 2: already followed
    }
]
}
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.
SubscriberList	Array	Returned results of added subscribers.
Subscriber_Account	String	Returned subscriber UserID.
Result	Integer	Addition result: 0: failed. 1: succeeded. 2: already followed this official channel.

Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.

20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel user does not exist, or once existed but has now been terminated.
130006	The number of subscribers for this official channel has reached its limit. Please refer to <a href="#">Pricing</a> to purchase or upgrade related packages and expand the maximum number of subscribers for the official channel. (After upgrading, you need to call the <a href="#">Modifying Official Account Information</a> API to update the maximum number of subscribers supported.)
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use.
131000	This AppId has not been enabled with official channel service.
130010	The number of official channels a subscriber can join has exceeded the limit. Please check and unsubscribe from some official channels.

## References

[Creating Official Accounts](#)

[Deleting Subscribers](#)

## Possibly Triggered Callbacks

[Before an Official Account Is Subscribed To](#)

[After an Official Account Is Subscribed To](#)

[When the Official Account's Subscription Limit is Reached](#)

# Deleting Subscribers

Last updated : 2025-05-27 10:16:37

## Feature Overview

This API is used by the app administrator to make a user unsubscribe from an official channel and be removed from the list of subscribers.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/delete_subscriber?sdkappid=88888888
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/official_account_open_http_svc/delete_subscriber	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to

	4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{
  "Official_Account": "@TOA#_@TOA#dOEF",  // The desired official channel (required)
  "SubscriberToDel_Account": [             ////UserID of the subscriber to be deleted
    "107867",
    "966582476"
  ]
}
```

Request Fields

Field	Type	Required	Description
Official_Account	String	Yes	ID of the official channel to be subscribed to. Use the OfficialAccountUserID field returned by the API during creation.
SubscriberList	Array	Yes	Array of subscribers to be added.
Subscriber_Account	String	Yes	UserID of the subscriber to be added.

Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "SubscriberList": [
    {
      "Subscriber_Account": "107867",
      "Result": 1
    },
    {
      "Subscriber_Account": "966582476",

```

```
        "Result": 1
      }
    ]
  }
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.
SubscriberList	Array	Returned results of deleted subscribers.
Subscriber_Account	String	Returned subscriber UserID.
Result	Integer	Deletion result: 0: failed. 1: succeeded. 2: not subscribed to this official channel.

Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.

130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel user does not exist, or once existed but has now been terminated.
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use.
131000	This AppId has not been enabled with official channel service.

## References

[Creating Official Accounts](#)

[Adding Subscribers](#)

## Possibly Triggered Callbacks

[After a Subscriber Is Deleted](#)

# Modifying Subscriber Information

Last updated : 2025-05-27 10:16:31

## Feature Overview

This API is used by the app administrator to modify relevant information of subscribers.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/modify_subscriber_info?sdkappid=88
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/official_account_open_http_svc/modify_subscriber_info	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

## Maximum Call Frequency

200 times/second

## Sample Request Packets

```
{
  "Official_Account": "@TOA#_test_for_penn",
  "Subscriber_Account": "107867",
  "CustomString": "test_modify_subscriber_custom_string"
}
```

## Request Fields

Field	Type	Required	Description
Official_Account	String	Yes	ID of the official channel to be modified. Use the OfficialAccountUserID field returned by the API during creation.
Subscriber_Account	String	Yes	ID of the subscriber whose information is to be modified.
CustomString	String	Yes	The custom field of the subscriber, which can be used by the application layer to implement some special feature requirements.

## Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response Fields

Field	Type	Description
-------	------	-------------



ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel user does not exist, or once existed but has now been terminated.
130007	Incorrect official channel user ID. Please modify and retry.
130008	The official channel user ID is already in use.
131000	This AppId has not been enabled with official channel service.

## References

[Creating Official Accounts](#)

[Adding Subscribers](#)



# Obtaining Subscribed Official Channel

Last updated : 2025-05-27 10:16:24

## Feature Overview

This API is used by the app administrator to obtain the list information of all official channels subscribed by a user.

## API Calling Description

### Sample Request URL

```
https://xxxxxx/v4/official_account_open_http_svc/get_subscribed_official_account_li
```

### Request Parameters

The table below only lists the parameters modified when this API is called and their descriptions. For details on the parameters, refer to [RESTful API Overview](#).

Parameter	Description
xxxxxx	The dedicated domain name corresponding to the country/region of the SDKAppID: China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/official_account_open_http_svc/get_subscribed_official_account_list	Request API.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.

identifier	The app admin account. For details, see <a href="#">App Admin</a> .
usersig	The signature generated by the app admin account. For details, see <a href="#">Generating UserSig</a> .
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	The value is always <code>json</code> .

Maximum Call Frequency

200 times/second

Sample Request Packets

```
{
  "Subscriber_Account": "107867",           //Subscriber
  "Limit": 200,
  "Offset": 0,
  "ResponseFilter":{                        //Filter fields
    "OfficialAccountBaseInfoFilter":[ //Filter fields for official channel inform
      "CreateTime",
      "Name",
      "Owner_Account",
      "LastMsgTime",
      "SubscriberNum",
      "Introduction",
      "FaceUrl",
      "Organization",
      "CustomString"
    ],
    "SelfInfoFilter":[                     //Filter fields for subscriber information
      "SubscribeTime",
      "CustomString"
    ]
  }
}
```

Request Fields

Field	Type	Required	Description
Subscriber_Account	String	Yes	ID of the subscriber whose information is to be modified.

Limit	Integer	No	Number of official channels pulled in a single pull, currently supporting a maximum value of 200. If not specified, the default is 200.
Offset	Integer	No	Indicates starting pulling from which official channel.
ResponseFilter	Object	No	<p>Includes two filters: OfficialAccountBaseInfoFilter and SelfInfoFilter.</p> <p>OfficialAccountBaseInfoFilter: Indicates the basic information fields of official channels to be pulled, mainly including:</p> <p>CreateTime: creation time of the official channel</p> <p>Name: name of the official channel</p> <p>Owner_Account: user who owns the official channel</p> <p>LastMsgTime: time of the last message within the official channel</p> <p>SubscriberNum: current number of subscribers to the official channel</p> <p>Introduction: introduction to the official channel</p> <p>FaceUrl: avatar URL of the official channel</p> <p>Organization: organization of the official channel</p> <p>CustomString: custom field at the official channel level</p> <p>If not specified, all fields will be pulled.</p> <p>SelfInfoFilter: Indicates the need to pull subscriber's own information, mainly including:</p> <p>SubscribeTime: time when the subscriber subscribed to the official channel</p> <p>CustomString: custom field at the subscriber level</p> <p>If not specified, all fields will be pulled.</p>

## Sample Response Packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TotalCount": 8,      //The total number of subscribed official channels, unrelat
  "OfficialAccountInfoList": [      //List of official channels returne
    {
      "Name": "test_mill_subscriber_OA",
      "Official_Account": "@TOA#_test_mill_subscriber_OA",
      "Owner_Account": "107867",
      "LastMsgTime": 1698894911,
      "SubscriberNum": 499800,
      "Introduction": "This is test Introduction",
      "FaceUrl": "http://this.is.face.url",
```

```
        "Organization": "Tencent",
        "CustomString": "test_custom_string",
        "SelfInfo": {
            "SubscribeTime": 1700655351,
            "CustomString": ""
        }
        //...There might be a lot more information to follow, but here information o

    ],
    "CompleteFlag": 1,           //Indicates whether the pull has ended. 1 means it
    "Next": 0                    //If the pull has ended, returns 0.
}
```

Response Fields

Field	Type	Description
ActionStatus	String	Processing result: OK: succeeded. FAIL: failed.
ErrorCode	Integer	Error code: 0: succeeded. Other values: failed.
ErrorInfo	String	Error information.
OfficialAccountInfoList	Array	The official channel information obtained. The results are filtered information based on the filter fields set in the filter.
TotalCount	Integer	Number of official channels subscribed by the user.
CompleteFlag	Integer	Indicates whether the pull has ended. 1: The pull has ended. 0: The pull has not ended.
Next	Integer	Indicates the next pull. If the current pull has not ended, it can be filled in the offset of the next pull as the offset for pagination.

Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information.

For common error codes (60000 to 79999), see [Error Codes](#).

The following table describes the error codes specific to this API:

Error Code	Description
6017	Invalid request parameter. Please modify and retry.
10004	Internal server error. Please try again.
20001	Invalid request packet. Please modify and retry.
20005	Internal server error. Please try again.
130002	Insufficient operation permissions. Please check the request parameters based on the error message.
130004	The official channel user does not exist, or once existed but has now been terminated.
130007	Incorrect official channel user ID. Please modify and retry.
131000	This AppId has not been enabled with official channel service.

## References

[Creating Official Accounts](#)

[Adding Subscribers](#)

# All/Tagged Push

## Pushing to All/Tagged Users

Last updated : 2025-05-27 10:21:30

All-user/Tag Push supports sending specific content, and also allows for the delivery of personalized content to specific user groups based on Tag, attribute, such as Membership Activities, Regional Notifications, etc. It aids in User Acquisition, Conversion, Activation Promotion, and other operational work phases, while also supporting Push Delivery Reports, Self-service Push Troubleshooting Tool. For more details, please see [Effect Display](#).

### Note:

1. An account must have logged in or been manually imported to receive messages from All-staff/Tag Push.
2. Universal Push Notification does not support Message Modification.

## Feature Overview

Supports sending pushes to all users.

Supports sending pushes based on user attributes.

Supports sending pushes by user Tag.

When an administrator pushes a message, the recipient sees the message sender as the administrator.

When an administrator specifies an account to push messages to other accounts, the recipient sees the sender not as the administrator, but as the account specified by the administrator.

Supports roaming, with the roaming storage duration consistent with that of regular message storage.

Due to the large number of accounts that need to be issued for All-user/Tag Push, it takes some time to issue all accounts (depending on the total number of accounts).

By setting the OnlineOnlyFlag to 1, push notifications can be sent without saving the session, roaming, and unread messages.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/timpush/push?usersig=xxx&identifier=admin&sdkappid=88888888&random
```

### Request parameters

Parameter	Description



https	Request protocol: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/timpush/push	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Chat console when creating an app.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

## Call frequency limit

This API includes all-user, attribute, and Tag Push. By default, it can be invoked up to 100 times per day, with an interval of more than 1 second between each push.

### Note:

API Frequency Limit. By default, you can make 100 free calls per day. Each additional call will cost an extra \$10 per day. If you need to adjust the API frequency, please go to the [Chat Console](#).

## Sample request packets

### Mass Push Example

Administrator conducts a mass push:

```
{
  "From_Account": "admin",
  "MsgRandom": 3674128,
  "OnlineOnlyFlag": 0, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
  "MsgBody": [
    {
```

```

        "MsgType": "TIMTextElem",
        "MsgContent": {
            "Text": "hi, beauty"
        }
    },
    "OfflinePushInfo": {
        "PushFlag": 0, // 0 means offline push enabled, 1 means offline push
disabled
        "Title": "Offline Push Title",
        "Desc": "Offline Push Content"
    }
}

```

Administrator directs a specific account to conduct a mass push (in the example, the sender's account is xiaoming):

```

{
    "From_Account": "xiaoming",
    "MsgRandom": 3674128,
    "OnlineOnlyFlag": 0, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ],
    "OfflinePushInfo": {
        "PushFlag": 0, // 0 means offline push enabled, 1 means offline push
disabled
        "Title": "Offline Push Title",
        "Desc": "Offline Push Content"
    }
}

```

#### Note:

From\_Account is the message sender account, which supports specifying any existing account. If the sender is not specified or does not exist, the API caller account will be used by default.

Only push to online users (Online Push):

```

{
    "From_Account": "xiaoming",
    "MsgRandom": 3674128,

```

```

    "OnlineOnlyFlag": 1, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ],
    "OfflinePushInfo": {
        "PushFlag": 1 // 0 means offline push enabled, 1 means offline push
disabled
    }
}

```

**Note:**

OnlineOnlyFlag set to 1 indicates push operations for both online and offline users, but OfflinePushInfo.PushFlag set to 1 means there will be no offline push. Therefore, the above example signifies that only online users will be pushed to.

**Push by User Tag Example**

The administrator pushes messages to users tagged with both "Stock A" and "Stock B":

```

{
    "From_Account": "admin",
    "MsgRandom": 124032,
    "OnlineOnlyFlag": 0, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
    "Condition": {
        "TagsAnd": ["Stock A", "Stock B"]
    },
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ],
    "OfflinePushInfo": {
        "PushFlag": 0, // 0 means offline push enabled, 1 means offline push
disabled
        "Title": "Offline Push Title",
        "Desc": "Offline Push Content"
    }
}

```

```
}
}
```

**Note:**

From\_Account is the message sender account, which supports specifying any existing account. If the sender is not specified or does not exist, the API caller account will be used by default.

The administrator pushes messages to users following either "Stock A" or "Stock B":

```
{
  "From_Account": "admin",
  "MsgRandom": 124032,
  "OnlineOnlyFlag": 0, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
  "Condition": {
    "TagsOr": ["Stock A", "Stock B"]
  },
  "MsgBody": [
    {
      "MsgType": "TIMTextElem",
      "MsgContent": {
        "Text": "hi, beauty"
      }
    }
  ],
  "OfflinePushInfo": {
    "PushFlag": 0, // 0 means offline push enabled, 1 means offline push
disabled
    "Title": "Offline Push Title",
    "Desc": "Offline Push Content"
  }
}
```

**User Attribute Push**

The administrator sends messages to Super Platinum members in Shenzhen:

```
{
  "From_Account": "admin",
  "MsgRandom": 389475,
  "OnlineOnlyFlag": 0, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
  "Condition": {
    "AttrsAnd": {
      "Membership Level": "Super Platinum Member",

```

```

        "city": " Shenzhen "
    },
    },
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ],
    "OfflinePushInfo": {
        "PushFlag": 0, // 0 means offline push enabled, 1 means offline push
disabled
        "Title": "Offline Push Title",
        "Desc": "Offline Push Content"
    }
}

```

**Note:**

From\_Account is the message sender account, which supports specifying any existing account. If the sender is not specified or does not exist, the API caller account will be used by default.

The administrator sends messages to Shenzhen users or Super Platinum members:

```

{
    "From_Account": "admin",
    "MsgRandom": 389475,
    "OnlineOnlyFlag": 0, // 0 means storing roaming and unread messages with a
push concurrency limit of 200 users/second; 1 means not storing roaming and
unread messages
    "Condition": {
        "AttrsOr": {
            "Membership Level": "Super Platinum User",
            "city": " Shenzhen "
        }
    },
    },
    "MsgBody": [
        {
            "MsgType": "TIMTextElem",
            "MsgContent": {
                "Text": "hi, beauty"
            }
        }
    ],
    "OfflinePushInfo": {

```

```
"PushFlag": 0, // 0 means offline push enabled, 1 means offline push disabled
"Title": "Offline Push Title",
"Desc": "Offline Push Content"
}
```

## Request packet fields

Field	Type	Attribute	Description
From_Account	String	Optional	Message Sender Account (supports specifying any existing account) <b>Note:</b> If the sender is not specified or the specified sender does not exist, the account of the API caller is used by default.
MsgRandom	Integer	Required	32-bit unsigned integer random number, with a value range from 0 to 4294967295 Backend uses message deduplication within the same second. <b>Please ensure it is a random number.</b>
OnlineOnlyFlag	Integer	Optional	The default is 0, indicating storing roaming and unread messages with a push concurrency limit of 200 people/second. 1 indicates not storing history messages and not counting unread messages <b>Note:</b> If you need to adjust the push concurrency limit, please contact <a href="#">Tencent Cloud customer service</a> or business staff.
Condition	Object	Optional	Condition has 4 types, which are: Attribute's "Or condition" AttrsOr Attribute's "And condition" AttrsAnd Tag's "Or condition" TagsOr Tag's "And condition" TagsAnd <b>Note:</b> AttrsOr, AttrsAnd, TagsOr, TagsAnd cannot coexist. If there is no Condition, it will be pushed to all users.
TagsOr	Array	Optional	The union of Tag conditions. A Tag is a string that does not exceed 50 bytes. <b>Note:</b> Attribute pushing and Tag pushing cannot both be used as push conditions at the same time. The number of Tags in TagsOr condition cannot exceed 10.

TagsAnd	Array	Optional	The intersection of Tag conditions. A Tag is a string that does not exceed 50 bytes. <b>Note:</b> Attribute pushing and Tag pushing cannot both be used as push conditions at the same time. The number of Tags in TagsAnd condition cannot exceed 10.
AttrsOr	Object	Optional	The union of Attribute conditions. <b>Note:</b> Attribute pushing and Tag pushing cannot both be used as push conditions at the same time.
AttrsAnd	Object	Optional	The intersection of Attribute conditions. <b>Note:</b> Attribute pushing and Tag pushing cannot both be used as push conditions at the same time.
MsgBody	Array	Required	Message content, please refer to <a href="#">MsgBody Message Content Description</a> (a message can include multiple message elements, so MsgBody is of Array type)
MsgType	String	Required	TIM message object types currently supported include: TIMTextElem (Text message) TIMLocationElem (Location message) TIMFaceElem (Expression message) TIMCustomElem(Custom message) TIMSoundElem( VMS ) TIMImageElem(Image message) TIMFileElem(File message) TIMVideoFileElem(Video message)
MsgContent	Object	Required	For each MsgType, use different MsgContent formats, please refer to <a href="#">TIMMsgElement object</a> 's Definition
OfflinePushInfo	Object	Optional	Offline Push Notification Configuration. For details, please refer to the <a href="#">OfflinePushInfo Description</a>

### Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "TaskId": "667015d4_537529d8_2000005e80aa873_d03ac87_56f5e750"
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing FAIL: Indicates failure
ErrorCode	Integer	Error code
ErrorInfo	String	Error message
TaskId	String	Push task ID

## Error codes

Unless a network error occurs (e.g., a 502 error), the HTTP status code for this interface will always be 200. **The real error codes and messages are indicated in the response body by ErrorCode and ErrorInfo.** For common error codes (60000 to 79999), see the [Error Code](#) document.

Private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90002	The MsgBody in the JSON format request packet does not match the message format description, or MsgBody is not of Array type. Please refer to the Definition of <a href="#">TIMMsgElement Object</a> .
90005	The JSON format request packet body is missing the MsgRandom field or the MsgRandom field is not of Integer type.
90007	The MsgBody type in the JSON format request packet body is not of Array type. Please change it to Array type.
90009	The request requires App Administrator permissions.
90010	The JSON format request packet does not conform to the message format description. Please refer to the Definition of <a href="#">TIMMsgElement Object</a> .
90020	Tag length exceeds the limit (cannot exceed 50 bytes).
90022	There are duplicate Tags in the TagsOr and TagsAnd conditions of the push.
90024	Pushes are too frequent; the interval between two pushes must be more than 1 second.



90026	Error in offline message storage time.
90032	The number of tags in the push conditions exceeds 10, or the number of Tags in the add Tag request exceeds 10.
90033	Attribute is invalid.
90039	Push by Attribute and Push by Tag cannot coexist.
90040	One of the tags in the Push Conditions is empty.
90045	All-staff/Tag Push feature is not activated.
90047	Push Attempts exceed the Daily Limit (default is 100 times).
90056	Request body for all-staff push is too large, currently supports a maximum length of 10K.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Obtaining Application Attribute Names

Last updated : 2025-05-27 10:21:33

## Feature Overview

Administrator obtaining application attribute name. Please [Set Application Attribute Name](#) first.

### Sample request URL

```
https://xxxxxx/v4/timpush/get_attr_name?usersig=xxx&identifier=admin&sdkappid=88888
```

### Request parameters

Parameter	Description
https	Request protocol is HTTPS Request method is POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/get_attr_name	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: json

### Call frequency limit

100 times per second.

Sample request packets

```
{ }
```

Request packet fields

No.

Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "AttrNames": {
    "0": "sex",
    "1": "city",
    "2": "country"
  },
  "AttrTypes": {
    "0": 0,
    "1": 0,
    "2": 1 // Number key 2 corresponds to the attribute "country", which
can only be set by account type 1 (the RegistrationID is automatically
generated when the push service registration is succ
  }
}
```

Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
AttrNames	Object	Contains multiple key-value pairs. Each pair represents the name corresponding to a specific attribute. For example, "0":"xxx" indicates that the name of attribute 0 is xxx

AttrTypes	Object	Containing multiple key-value pairs. Each key-value pair indicates the account type corresponding to the nth attribute. "0": 0 means the account type of attribute No. 0 is 0 (IM account type). "0": 1 means the account type of attribute No. 0 is 1 (RegistrationID automatically generated when the push service registration is successful).
-----------	--------	---

## Error codes

Unless a network error occurs (e.g., 502 error), the HTTP return code of this interface is always 200. **The actual error code and error message are indicated by ErrorCode and ErrorInfo in the response payload.** For public error codes (60000 to 79999), refer to the [Error Codes](#) documentation.

The private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Setting Application Attribute Names

Last updated : 2025-06-09 10:07:06

## Feature Overview

Each application can set custom User Attributes, with a maximum of 10. This interface allows you to set the name of each attribute. Once set, they can be used for push notifications by User Attributes, etc.

### Sample request URL

```
https://xxxxxx/v4/timpush/set_attr_name?usersig=xxx&identifier=admin&sdkappid=88888
```

### Request parameters

Parameter	Description
https	Request protocol: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: <code>console.tim.qq.com</code> Singapore : <code>adminapisgp.im.qcloud.com</code> Seoul : <code>adminapikr.im.qcloud.com</code> Frankfurt : <code>adminapiger.im.qcloud.com</code> Silicon Valley : <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/timpush/set_attr_name	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

## Call frequency limit

100 times per second.

## Sample request packets

Setting the 0th attribute of the application means gender, the 1st attribute means city, the 2nd attribute means country.

```
{
  "AttrNames": {
    "0": "sex",
    "1": "city",
    "2": "country"
  },
  "AttrTypes": {
    "0": 0, // ccount type corresponding to setting attributes, which does
    not support modification afterward.
    "1": 0,
    "2": 0
  }
}
```

## Request packet fields

### AttrNames Object description

Field	Type	Attribute	Description
Number key	String	Required	Indicates which attribute (between "0" and "9")
Attribute name	String	Required	The attribute name must not exceed 50 bytes. The application can have up to 10 push attributes (numbered from 0 to 9), with the user defining the meaning of each attribute

### AttrTypes Object description

Field	Type	Attribute	Description
Number key	String	Optional	Indicates which attribute (between "0" and "9"), must be a Number Key included in AttrNames.
Attribute name	Interger	Optional	The account type corresponding to the attribute defaults to 0. An attribute can only correspond to one account type. For details, see <a href="#">Introduction to Typical Push Scenarios</a> . 0: The account type represents the RegistrationID automatically generated when the push service registration is successful. Account type represents: Users log in to IM account to transmit UserID.

## Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing FAIL: Indicates failure
ErrorCode	Integer	Error code
ErrorInfo	String	Error message

## Error codes

Unless a network error occurs (such as a 502 error), the HTTP return code for this interface is always 200. **The actual error code and error information are represented by ErrorCode and ErrorInfo in the response body.**

Common error codes (60000 to 79999) are detailed in the [Error codes](#) documentation.

Private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.



## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Obtaining User Attributes

Last updated : 2025-05-27 10:21:39

## Feature Overview

Get User Attributes (must be called with an admin account). You can retrieve up to 100 users' attributes at a time. Before use, please first [set the application attribute names](#) and then [set the user attributes](#).

### Sample request URL

```
https://xxxxxx/v4/timpush/get_attr?usersig=xxx&identifier=admin&sdkappid=88888888&r
```

### Request parameters

Parameter	Description
https	Request protocol: HTTPS. Request method: POST.
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/timpush/get_attr	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

### Call frequency limit

100 times per second.

### Sample request packets

```
{
  "To_Account": [
    "Mary",
    "xiaoming",
    "xiaohua"
  ]
}
```

### Request packet fields

Field	Type	Attribute	Description
To_Account	Array	Required	Target User Account List

### Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "UserAttrs": [
    {
      "To_Account": "xiaoming",
      "Attrs": {
        "sex": "male",
        "city": "ShenZhen"
      }
    },
    {
      "To_Account": "xiaohua",
      "Attrs": {}
    }
  ],
  "ErrorList": [
    {
      "ErrorCode": 70107, // account does not exist
      "To_Account": "Mary"
    }
  ]
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
UserAttrs	Object Array	User Tag Content List.
ErrorList	Object Array	Set attributes. Some users may succeed while others may fail. Error message list for failed accounts.

### Field description of json Object in the UserAttrs array

Field	Type	Description
To_Account	String	User account.
Attrs	Object	Attribute content. Each attribute is a key-value pair, with the key as the attribute name and the value as the attribute value for this user. the user attribute value must not exceed 50 bytes.

### Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

## Error codes

Unless a network error occurs (e.g., a 502 error), the HTTP status code for this interface will always be 200. **The real error codes and messages are indicated in the response body by ErrorCode and ErrorInfo.** For common error codes (60000 to 79999), see the [Error Code](#) document.

Private error codes for this API are as follows:

--	--

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Setting User Attributes

Last updated : 2025-05-27 10:21:42

## Feature Overview

The administrator sets attributes for users. Each time, a maximum of 100 users can be assigned attributes. Before use, please first [set the application attribute name](#).

### Sample request URL

```
https://xxxxxx/v4/timpush/set_attr?usersig=xxx&identifier=admin&sdkappid=88888888&r
```

### Request parameters

Parameter	Description
https	The request protocol is: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/timpush/set_attr	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

### Call frequency limit

100 times per second.

### Sample request packets

```
{
  "UserAttrs":
  [
    {
      "To_Account": "379C2F0D-290D-47AE-94D1-919058C39C77", // System-generated RegistrationID upon successful push service registration
      "Attrs": {
        "sex": "female",
        "city": "NewYork"
      }
    },
    {
      "To_Account": "xiaoming",
      "Attrs": {
        "sex": "male",
        "city": "ShenZhen"
      }
    }
  ]
}
```

### Request packet fields

Field	Type	Attribute	Description
To_Account	String	Required	Target User Account.
Attrs	Object	Required	Attribute Collection. Each attribute is a key-value pair, where the key is the attribute name, and the value is the corresponding attribute value for the user. User attribute values cannot exceed 50 bytes.

### Sample response packets

All operations are successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Partially successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ErrorList": [
    {
      "ErrorCode": 90035, // A property can only be applicable to one
account type
      "To_Account": "379C2F0D-290D-47AE-94D1-919058C39C77"
    }
  ]
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
ErrorList	Object Array	Set attributes. Some users may succeed while others may fail. Error message list for failed accounts.

### Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

## Error codes

Unless a network error occurs (e.g., 502 error), the HTTP return code of this interface is always 200. **The actual error code and error message are indicated by ErrorCode and ErrorInfo in the response payload..** For public error codes (60000 to 79999), refer to the [Error Codes](#) documentation.



The private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
90018	The number of requested accounts exceeds the limit.
90033	Attribute is invalid.
91000	Internal service error, please try again.
90035	A property is only applicable to one account type. When registering a push service successfully, RegistrationID is automatically generated for the Chat account type. For details, see <a href="#">Introduction to Typical Push Scenarios</a> .

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Deleting User Attributes

Last updated : 2025-05-27 10:21:45

## Feature Overview

The administrator removes attributes from users. Note that you can only remove attributes for up to 100 users at a time. Before using it, please [set the application attribute names](#).

### Sample request URL

```
https://xxxxxx/v4/timpush/remove_attr?usersig=xxx&identifier=admin&sdkappid=88888888
```

### Request parameters

Parameter	Description
https	Request protocol: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/timpush/remove_attr	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

### Call frequency limit

100 times per second.

### Sample request packets

```
{
  "UserAttrs": [
    {
      "To_Account": "Mary",
      "Attrs": [
        "sex",
        "city"
      ]
    },
    {
      "To_Account": "xiaoming",
      "Attrs": [
        "sex",
        "city"
      ]
    }
  ]
}
```

### Request packet fields

Field	Type	Attribute	Description
UserAttrs	Array	Required	The array of user attributes. A single user attribute consists of To_Account and Attrs.
To_Account	String	Required	Target User Account.
Attrs	Array	Required	Set of attribute names, note that only attribute names are needed here, not key-value pairs.

### Sample response packets

All successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Partially successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ErrorList": [
    {
      "ErrorCode": 70107,
      "To_Account": "Mary"
    }
  ]
}
```

Response packet field description

Field	Type	Description
ActionStatus	String	Request processing result: OK: processing successful. FAIL: processing failed.
ErrorCode	Integer	Error Codes.
ErrorInfo	String	Error message.
ErrorList	Object Array	Set attributes may be partially successful for some users and partially failed for others. Error message list for failed accounts.

Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

Error codes

Unless a network error occurs (e.g., a 502 error), the HTTP status code for this interface will always be 200.**The real error codes and messages are indicated in the response body by ErrorCode and ErrorInfo.** For common error codes (60000 to 79999), see the [Error Code](#) document.

Private error codes for this API are as follows:

--	--

Error code	Description
90001	Failed to parse JSON format, please check if the request packet conforms to JSON specification.
90009	The request requires App administrator permissions.
90018	The number of requested accounts exceeds the limit.
90033	Invalid Attribute.
91000	Internal server error, please retry.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Obtaining User Tags

Last updated : 2025-05-27 10:24:11

## Feature Overview

Retrieve User Tags (must be called with an admin account). You can only retrieve tags for a maximum of 100 users at a time.

### Sample request URL

```
https://xxxxxx/v4/timpush/get_tag?usersig=xxx&identifier=admin&sdkappid=88888888&ra
```

### Request parameters

Parameter	Description
https	The request protocol is: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/get_tag	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: json

### Call frequency limit

100 times per second.

Sample request packets

```
{
  "To_Account": [
    "xiaoming",
    "xiaohong",
    "Mary"
  ]
}
```

Request packet fields

Field	Type	Attribute	Description
To_Account	Array	Required	Target User Account List.

Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "UserTags": [
    {
      "To_Account": "xiaoming",
      "Tags": ["GoldMember", "WeekPassUser"]
    },
    {
      "To_Account": "xiaohong",
      "Tags": ["PlatinumMember", "MonthPassUser"]
    }
  ],
  "ErrorList": [
    {
      "ErrorCode": 70107,
      "To_Account": "Mary"
    }
  ]
}
```

Response packet field description

--	--	--

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
UserTags	Object Array	User Tag Content List.
ErrorList	Object Array	Set attributes. Some users may succeed while others may fail. Error message list for failed accounts.

Field description of json Object in the UserTags array

Field	Type	Description
To_Account	String	User account.
Tags	Array	Tag array. The maximum length of a single tag should not exceed 50 bytes. Each tag is a string type.

Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

## Error codes

Unless a network error occurs (e.g., 502 error), the HTTP return code of this interface is always 200. **The actual error code and error message are indicated by ErrorCode and ErrorInfo in the response payload..** For public error codes (60000 to 79999), refer to the [Error Codes](#) documentation.

The private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.



90009	The request requires App Administrator permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

- [All-user/Tag Push](#)
- [Set application attribute name](#)
- [Get application attribute name](#)
- [Set user attributes](#)
- [Delete user attributes](#)
- [Get User Attributes](#)
- [Add User Tags](#)
- [Get User Tags](#)
- [Delete User Tags](#)
- [Clear User Tags](#)
- [Push Recall](#)

# Adding User Tags

Last updated : 2025-05-27 10:24:17

## Feature Overview

Administrator adds tags for users.

### Note:

Each request can add tags for up to 100 users. A maximum of 10 tags can be added per user in the request body. A single user can set up to 100 tags. If the number of tags for the user exceeds 100, please delete old tags before adding new ones.

An application can set a maximum of 1,000 tags. This is the deduplicated total for all users' tags.

The maximum length of a single tag is 50 bytes.

## Sample request URL

```
https://xxxxxx/v4/timpush/add_tag?usersig=xxx&identifier=admin&sdkappid=88888888&ra
```

## Request parameters

Parameter	Description
https	The request protocol is: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/add_tag	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.

random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: json

Call frequency limit

100 times per second.

Sample request packets

```
{
  "UserTags": [
    {
      "To_Account": "xiaoming",
      "Tags": ["GoldMember", "WeekPassUser"]
    },
    {
      "To_Account": "379C2F0D-290D-47AE-94D1-919058C39C77" //
Automatically generated RegistrationID when push service registration is
successful
      "Tags": ["PlatinumMember", "MonthPassUser"]
    }
  ]
}
```

Request packet fields

Field	Type	Attribute	Description
UserTags	Object Array	Required	Account tag information.

Field description of json Object in the UserTags array

Field	Type	Attribute	Description
To_Account	String	Required	Target User Account.
Tags	Array	Required	Tag array: Each Tag must be a string type and the maximum length of a single Tag should not exceed 50 bytes.

Sample response packets

All successful:

```
{
```

```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0
}
```

Partially successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ErrorList": [
    {
      "ErrorCode": 90035, // A tag can only be applicable to one account
      "To_Account": "379C2F0D-290D-47AE-94D1-919058C39C77"
    }
  ]
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
ErrorList	Object Array	Set attributes may result in partial user success and partial failure. Error message list for failed accounts.

Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

## Error codes

Unless a network error occurs (e.g., 502 error), the HTTP return code of this interface is always 200. **The actual error code and error message are indicated by ErrorCode and ErrorInfo in the response payload.** For public error codes (60000 to 79999), refer to the [Error Codes](#) documentation.

The private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error, please try again.
90035	A tag can only be applicable to one account type. When registering a push service successfully, RegistrationID is automatically generated for the IM account type. For details, see <a href="#">Introduction to Typical Push Scenarios</a> .

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Deleting User Tags

Last updated : 2025-05-27 10:24:20

## Feature Overview

The administrator removes the Tag from the user. Note that you can only remove the Tag for up to 100 users at a time.

### Sample request URL

```
https://xxxxxx/v4/timpush/remove_tag?usersig=xxx&identifier=admin&sdkappid=888888888
```

### Request parameters

Parameter	Description
https	The request protocol is: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/remove_tag	Request API.
usersig	The signature generated by the App Administrator Account, see <a href="#">UserSig Background API</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: json

### Call frequency limit

100 times per second.

### Sample request packets

```
{
  "UserTags": [
    {
      "To_Account": "xiaoming",
      "Tags": ["GoldMember", "WeekPassUser"]
    },
    {
      "To_Account": "xiaohong",
      "Tags": ["Platinum Member", "MonthPassUser"]
    }
  ]
}
```

### Request packet fields

Field	Type	Attribute	Description
To_Account	String	Required	Target User Account.
Tags	Array	Required	Tag array: Each Tag must be a string type and the maximum length of a single Tag should not exceed 50 bytes.

### Sample response packets

All successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Partially successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ErrorList": [
    {
      "ErrorCode": 70107,
      "To_Account": "xiaohong"
    }
  ]
}
```

```
}  
]  
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
ErrorList	Object Array	Set attributes may be partially successful for some users and partially failed for others. Error message list for failed accounts.

Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

## Error codes

Unless a network error occurs (e.g., 502 error), the HTTP return code of this interface is always 200. **The actual error code and error message are indicated by ErrorCode and ErrorInfo in the response payload..** For public error codes (60000 to 79999), refer to the [Error Codes](#) documentation.

The private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
90018	The number of requested accounts exceeds the limit.



91000

Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API Online Test](#) tool to debug this interface.

## Reference

[All-user/Tag Push](#)[Set application attribute name](#)[Get application attribute name](#)[Set user attributes](#)[Delete user attributes](#)[Get User Attributes](#)[Add User Tags](#)[Get User Tags](#)[Delete User Tags](#)[Clear User Tags](#)[Push Recall](#)

# Deleting All User Tags

Last updated : 2025-05-27 10:24:24

## Feature Overview

The administrator deletes all Tags for the user. Note that you can delete all Tags for a maximum of 100 users at a time.

### Sample request URL

```
https://xxxxxx/v4/timpush/clear_all_tags?usersig=xxx&identifier=admin&sdkappid=8888
```

### Request parameters

Parameter	Description
https	The request protocol is: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: <code>console.tim.qq.com</code> Singapore: <code>adminapisgp.im.qcloud.com</code> Seoul: <code>adminapikr.im.qcloud.com</code> Frankfurt: <code>adminapiger.im.qcloud.com</code> Silicon Valley: <code>adminapiusa.im.qcloud.com</code> Jakarta: <code>adminapiidn.im.qcloud.com</code>
v4/timpush/clear_all_tags	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Instant Messaging Console when an application is created.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

## Call frequency limit

100 times per second.

## Sample request packets

```
{
  "To_Account": [
    "xiaoming",
    "xiaohong"
  ]
}
```

## Request packet fields

Field	Type	Attribute	Description
To_Account	Array	Required	Target User Account.

## Sample response packets

All successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

Partially successful:

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0,
  "ErrorList": [
    {
      "ErrorCode": 70107,
      "To_Account": "xiaohong"
    }
  ]
}
```

## Response packet field description

Field	Type	Description
-------	------	-------------

ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error message.
ErrorList	Object Array	Set attributes. Some users may succeed while others may fail. Error message list of failed accounts.

Field description of json Object in the ErrorList array

Field	Type	Description
To_Account	String	Target user account.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error description, which may be empty.

## Error codes

Unless a network error occurs (e.g., 502 error), the HTTP return code of this interface is always 200. **The actual error code and error message are indicated by ErrorCode and ErrorInfo in the response payload..** For public error codes (60000 to 79999), refer to the [Error Codes](#) documentation.

The private error codes for this API are as follows:

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Recalling Push

Last updated : 2025-05-27 10:24:30

If the content pushed to all staff/Tag is incorrect, it may negatively impact the product when the end user views or clicks it. Timely handling is necessary. You can choose to recall the push.

## Feature Overview

**Terminate:** Issuing push tasks takes some time, and the accounts not yet issued will have their deliveries terminated.

**Recall:** For accounts that have been issued, it supports recalling unread/roaming messages.

**Overwrite:** If the issued accounts have received offline push, it supports overwriting the push.

This interface supports the termination/recall/overwrite of push tasks for All-user/Tags. In the following text, termination/recall/overwrite is referred to as recall by default.

The recall validity period is 24 hours, starting from the task initiation time. Push tasks older than 24 hours cannot be recalled.

## API Calling Description

### Sample request URL

```
https://xxxxxx/v4/timpush/revoke?usersig=xxx&identifier=admin&sdkappid=88888888&ran
```

### Request parameters

Parameter	Description
https	Request protocol: HTTPS Request method: POST
xxxxxx	The dedicated domain corresponding to the country/region of your SDKAppID. China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com

v4/timpush/revoke	Request API.
usersig	The Signature generated by the App admin account. For details, see <a href="#">Generating UserSig</a> .
identifier	You must be an App admin account. For more details, see <a href="#">App Admins</a> .
sdkappid	SdkAppid assigned by the Chat console when creating an app.
random	A random 32-bit unsigned integer ranging from 0 to 4294967295.
contenttype	Fixed value: <code>json</code>

## Invocation Frequency

This interface invocation is limited to 1 time per second.

## Sample request packets

```
{
  "TaskId": "660cc447_537ed82a_200000cd7ee17f5_84035729_bc614e", // Push taskId
  "OfflinePushInfo": { // If roaming/unread is not specified when pushing (OnlinePushInfo)
    "Title": "Recall Title",
    "Desc": "The other party has recalled a message",
    "Ext": "{\"entity\":{\"key1\":\"value1\",\"key2\":\"value2\"}}\" /
  }
}
```

### Note:

- Supported vendors for Offline Push Overwrite: APNS/Google FCM/Huawei/Honor. Offline push overwrite is not supported for other vendors. (Google FCM notification mode supports overwrite, data mode does not support overwrite yet)
- When recalling, if the recipient is in the foreground and the default offline push (notification bar message) is read, that offline push will not be overwritten.

## Request Packet Field Description

Field	Type	Attribute	Description
TaskId	String	Required	All Staff/ Tag Push Task ID.
OfflinePushInfo	Object	Optional	Offline Push Notification Configuration, please refer to <a href="#">Offline Push OfflinePushInfo description</a> . <b>Note :</b> If OfflinePushinfo.PushFlag=1 is set or OfflinePushInfo is not set, offline push will not be overwritten.

## Sample response packets

```
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Response packet field description

Field	Type	Description
ActionStatus	String	Processed Request Result: OK: Indicates successful processing. FAIL: Indicates failure.
ErrorCode	Integer	Error Identifier: 0 indicates success. Non-zero indicates failure.
ErrorInfo	String	Error message.

## Error codes

Unless a network error occurs (e.g., a 502 error), the HTTP return code for this interface is always 200. The actual error code and error information are represented through ErrorCode and ErrorInfo in the response payload. For common error codes (60000 to 79999), refer to the [Error Codes](#) document.

Error code	Description
90001	Failed to parse the JSON format. Please ensure the request packet conforms to the JSON specification.
90009	The request requires App Administrator permissions.
90049	Invalid TaskId for recall, no push records. Use the <a href="#">All-user/Tags push</a> interface for the push, and only the returned TaskId can be used for recall.
90050	Duplicate recall, a push task that has already been recalled cannot be recalled again.
90051	Recall too frequent, recall frequency limit is 1 time per second.
90052	Exceeds recall validity period, recall must be done within 24 hours. Push tasks older than 24



	hours cannot be recalled.
90053	Recall invalid. The push task specified no roaming/unread (OnlineOnlyFlag=0), but OfflinePushInfo was not included during recall.
91000	Internal service error, please try again.

## API Debugging Tool

Use the [RESTful API online debugging tool](#) to debug this API.

## Reference

[All-user/Tag Push](#)

[Set application attribute name](#)

[Get application attribute name](#)

[Set user attributes](#)

[Delete user attributes](#)

[Get User Attributes](#)

[Add User Tags](#)

[Get User Tags](#)

[Delete User Tags](#)

[Clear User Tags](#)

[Push Recall](#)

# Webhooks

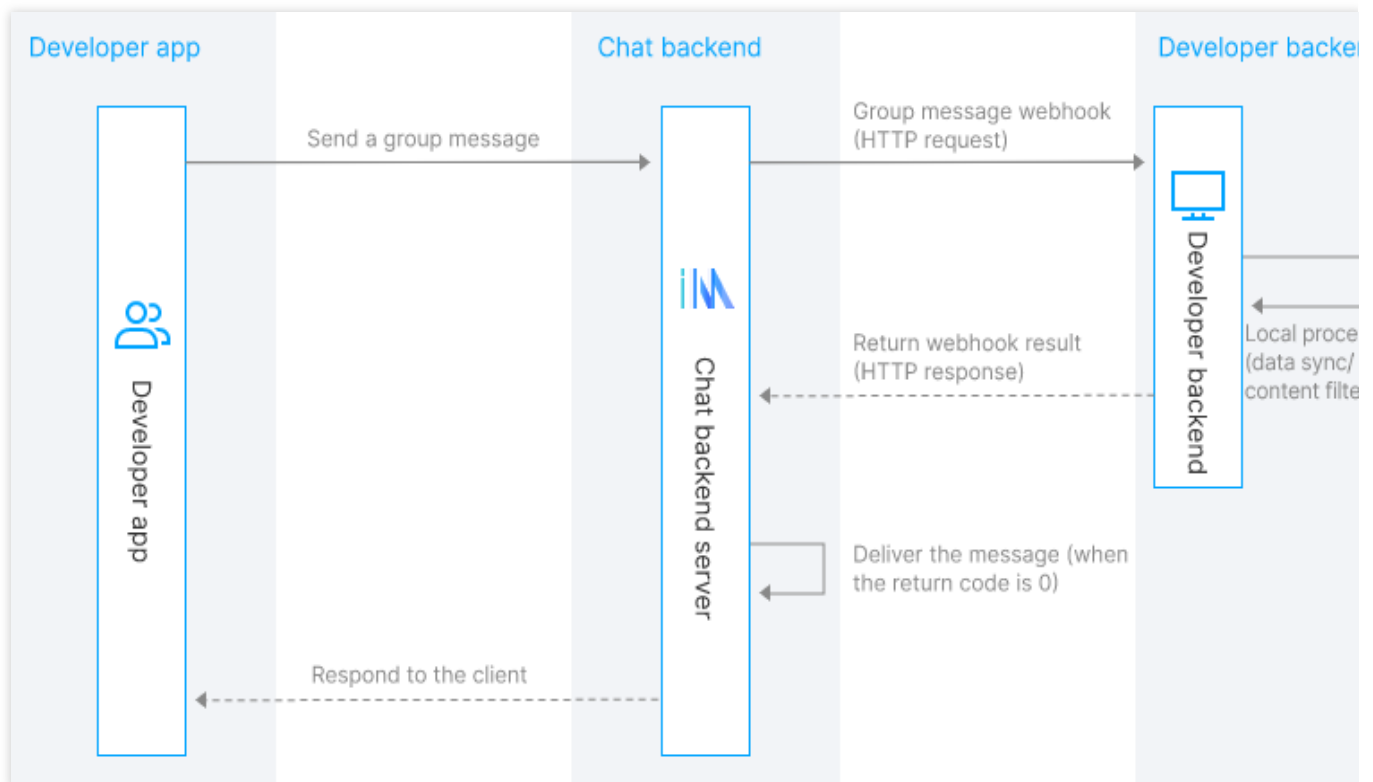
## Webhook Overview

Last updated : 2024-08-09 10:47:54

### Overview

To give you refined control over app features, Chat provides you with powerful webhooks free of charge. The webhooks use persistent connection mode by default. A webhook means that the Chat backend sends a request to the app backend server before or after an event occurs. This allows the app backend to synchronize data if necessary or intervene in the subsequent event processing. For more information about the webhooks currently supported by Chat, see the [Webhook Command List](#).

A webhook is sent to the app backend server using an HTTP/HTTPS request, and the app backend server must process the Chat webhook request and provide a response as soon as possible. Take the [Before Group Message Is Sent](#) webhook event as an example. Before the message is sent, the Chat backend sends a webhook request to the app backend server and determines whether the message should be sent based on the webhook result. Based on the webhook, the app can synchronize the message. The following figure shows the webhook process.



## Webhook Classification

Webhooks can be classified into four types according to their functions:

Online status webhooks

Relationship chain webhooks

One-to-one message webhooks

Group webhooks

Webhooks can be classified into two types by process:

Webhook before an action occurs: the purpose of this type of webhook is to allow the app backend to intervene in the processing logic of the event. Chat will determine the subsequent processing flow based on the return code of the webhook. For example, the webhook before a group message is sent is this type of webhook.

Webhook after an action occurs: the purpose of this type of webhook is to allow the app backend to implement essential data synchronization. Chat ignores the return codes of such webhooks. For example, the webhook after a member quits a group is this type of webhook.

## Webhook Protocol

Webhooks are based on HTTP/HTTPS protocols. The app backend must provide a webhook URL to Chat, and Chat uses a POST request to initiate a webhook request to the app backend. When initiating a webhook request, Chat adds the following parameters at the end of the URL provided by the app backend:

Parameter	Description
SdkAppid	App ID assigned by Chat
CallbackCommand	Webhook command word
contenttype	Optional. The value is generally a JSON string.
ClientIP	IP address of the client
OptPlatform	Client platform. Depending on the platform type, the following values are available: <code>RESTAPI</code> (requests are sent using RESTful APIs) and <code>Web</code> (requests are sent using Web SDKs), <code>Android</code> , <code>iOS</code> , <code>Windows</code> , <code>macOS</code> , <code>iPad</code> , and <code>Unknown</code> (requests are sent using an unknown device).

### Note:

"IOS" (all in uppercase) is used in the `State.StateChange` webhook, while "iOS" (the first letter is in lowercase) is used in other webhooks. Please perform compatibility processing during use.

The specific webhook content is included in the HTTP request packet. For details, see the following webhook examples.

## Webhook Examples

Webhook request example:

```
POST /?
SdkAppid=888888&CallbackCommand=Group.CallbackAfterNewMemberJoin&contenttype=json&ClientIP=$ClientIP&OptPlatform=$OptPlatform HTTP/1.1
Host: www.example.com
Content-Length: 337
{
  "CallbackCommand": "Group.CallbackAfterNewMemberJoin",
  "GroupId": "@TGS#2J4SZEDEL",
  "Type": "Public",
  "JoinType": "Apply",
  "Operator_Account": "leckie",
  "NewMemberList": [
    {
      "Member_Account": "jared"
    },
    {
      "Member_Account": "tommy"
    }
  ]
}
```

Webhook response example:

```
HTTP/1.1 200 OK
Server: nginx/1.7.10
Date: Fri, 09 Oct 2015 02:59:55 GMT
Content-Length: 75
{
  "ActionStatus": "OK",
  "ErrorInfo": "",
  "ErrorCode": 0
}
```

## Webhook Timeout Period and Retry

The timeout period for Chat webhooks to the app backend is two seconds.

Before event occurrence, webhooks are not retried. After event occurrence, webhooks are not retried by default, and you can configure whether to retry the webhooks when they time out.

To ensure a high webhook success rate, third-party apps need to process webhooks quickly. For example, the app can send a webhook response and then process the specific business logic.

## Handling Policy for Webhook Timeouts Before Event Occurrence

If a webhook times out before event occurrence, the default policy is to deliver the message.

You can also configure the handling policy for webhook timeouts before event occurrence in the console. For example, when a webhook timeout occurs before a group message is sent, you can specify whether to deliver the message.

## Security Considerations

Chat supports both HTTP and HTTPS webhooks. For HTTPS webhooks, you need to configure a certificate issued by a CA or a certificate issued by Chat free of charge in the WebServer of the app backend.

### Note:

To get a certificate issued by Chat free of charge, you need to log in to the console and configure webhook URL and download the certificate. For more information, see [Webhook Configuration](#).

Related security issues are as follows:

1. HTTP transmits data in plain text, and data confidentiality cannot be guaranteed. Therefore, HTTPS is recommended.
2. It's impossible to determine whether a webhook request really comes from Chat.

For request source security, we provide the **Webhook Authentication** solution:

1. Configure the webhook URL and enable webhook in the console.
2. During webhook URL configuration, enable authentication and configure the authentication token. Then, the signature ( `Sign` ) and signing timestamp ( `RequestTime` ) will be added to the webhook request URL. The signature algorithm is **Sign=sha256(TokenRequestTime)**.
3. The app backend authenticates the webhook request. It uses SHA256 to calculate and verify the signature based on the local authentication token and the signing timestamp ( `RequestTime` ) in the webhook URL.

Signature algorithm sample:

```
Token=xxxxyyyy
```

```
RequestTime=1669872112
```

```
Sign=sha256(xxxxxyyyy1669872112)=17773bc39a671d7b9aa835458704d2a6db81360a5940292  
b587d6d760d484061
```

## Webhook

```
URL=URL&Sign=17773bc39a671d7b9aa835458704d2a6db81360a5940292b587d6d760d484061&RequestTime=1669872112
```

## Common Reasons for Webhook Failures

If a webhook failure occurs, check whether the configured webhook service has a problem according to the following checklist:

Webhook Failure Symptom	Possible Reason
Access to the webhook URL times out	<ol style="list-style-type: none"><li>1. Chat cannot complete DNS resolution. In this case, check whether the domain name is valid on the public network. For example, if the webhook host is <code>http://notexist.com</code>, Chat cannot complete DNS resolution because this domain name does not exist.</li><li>2. Chat cannot access the IP address configured in the webhook URL. In this case, check whether this IP address is accessible from the public network. For example, if the webhook host is <code>http://10.0.0.1</code>, Chat cannot access this IP address because the domain name is a private IP address of the app.</li><li>3. The failure occurs due to the firewall policy of the app webhook service. In this case, check the firewall configuration. For example, a webhook failure occurs if the app webhook server denies all requests arriving at port 80.</li></ol>
Access denied by the webhook service	Chat can access the host, but a connection is not established. In this case, check whether the WebServer has started properly. For example, a webhook failure will occur when the WebServer of the app webhook server has not started or when the port configuration is incorrect.
HTTPS certificate configuration error of the webhook service	This can occur when the webhook type is HTTPS (or HTTPS mutual authentication). Chat can access the app webhook server, but determines that the certificate configured on the app WebServer is invalid. In this case, check that the HTTPS certificate is properly configured.
HTTPS mutual authentication configuration error of the webhook service	This can occur when the webhook type is HTTPS mutual authentication. Chat verifies that the certificate configured on the app webhook server is valid, but the app webhook server fails to verify the Chat certificate. <b>Note :</b> The HTTPS two-way authentication method has been taken offline, and we recommend you to use the more convenient <b>Webhook Authentication</b> method.
The HTTP return code of the webhook service is not 200	The webhook request is successful, but the HTTP return code in the response packet is not 200.

The webhook response packet could not be parsed	The webhook response packet is not in JSON format.
---	--

# Webhook Command List

Last updated : 2025-02-12 17:34:12

## Online Status

Webhook Type	Webhook Command Word
Webhook for status change	<a href="#">State.StateChange</a>

## Profile Relationship Chain

Webhook Type	Webhook Command Word
After a Profile Is Updated	<a href="#">Profile.CallbackPortraitSet</a>
Before a Friend Is Added	<a href="#">Sns.CallbackPrevFriendAdd</a>
Before a Friend Request Is Responded	<a href="#">Sns.CallbackPrevFriendResponse</a>
After a Friend Is Added	<a href="#">Sns.CallbackFriendAdd</a>
After a Friend Is Deleted	<a href="#">Sns.CallbackFriendDelete</a>
After a User Is Added to Blocklist	<a href="#">Sns.CallbackBlackListAdd</a>
After a User Is Removed from Blocklist	<a href="#">Sns.CallbackBlackListDelete</a>

## One-to-One Message

Webhook Type	Webhook Command Word
Before a One-to-One Message Is Sent	<a href="#">C2C.CallbackBeforeSendMsg</a>
After a One-to-One Message Is Sent	<a href="#">C2C.CallbackAfterSendMsg</a>
After a One-to-One message Is Marked as Read	<a href="#">C2C.CallbackAfterMsgReport</a>
After A One-to-One Message Is Recalled	<a href="#">C2C.CallbackAfterMsgWithdraw</a>



# Groups

Webhook Type	Webhook Command Word
Before a Group Is Created	<a href="#">Group.CallbackBeforeCreateGroup</a>