

# **Instant Messaging**

## **FAQs**

### **Product Documentation**



## Copyright Notice

©2013-2019 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 Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. 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

## FAQs

Purchase

SDK

Account Authentication

User Profile and Relationship Chain

Message

Group

Audio-Video Group

# FAQs

## Purchase

Last updated : 2020-09-16 14:58:43

### **Do IM pro edition and IM flagship edition have DAU limits?**

Fees charged for usage exceeding the free quota of the professional or flagship package are based on DAU and peak group count that exceed the free quota. Fees are charged according to their peak values within a calendar month on a postpaid basis. Fees incurred will be deducted on the third day of the next month.

If the DAU of each day within a calendar month is lower than 10,000, no DAU fees are incurred for that month. If the DAU is higher than 10,000 on any day within a calendar month, you will be charged based on the highest DAU value in that month.

For more billing information, see [Pricing](#).

### **What should I do if an error occurs after I have created 100 accounts in IM?**

You can create up to 100 accounts in the trial edition. To use the accounts in a production environment, you need to configure a standard billing plan for each SDKAppID. In addition, you can call the [API for deleting accounts](#) to delete unwanted accounts in the trial edition. **User data cannot be recovered after the accounts are deleted**, so proceed with caution.

### **Does IM support global access?**

IM provides highly interconnected, reliable, and secure global network connections and proprietary multi-level optimal addressing algorithms to implement scheduling across private and public networks. When terminals log in, the IM SDK connects to the nearest access nodes or cache nodes. Global access and cache nodes include the following:

- China: South China, North China, East China, Hong Kong, Taiwan regions, etc.
- Other countries (regions):
  - Asia: Singapore, Indonesia, UAE, Thailand, Japan, Vietnam, and India
  - Europe: United Kingdom, Netherlands, Germany, Italy, Norway, France, and Russia
  - South America: Brazil
  - North America: United States, Canada, and Mexico
  - Oceania: Australia
  - Africa: South Africa

### **Since IM allows the creation of up to 50 AVChatRooms, how many AVChatRooms can be created after I create 3 and then disband 3?**

If you create 3 AVChatRooms and then disband 3, you can still create 50. The number of AVChatRooms you can create is a net value. Disbanded AVChatRooms do not count against the quota. If you have an AVChatRoom that has no members but has not been disbanded, it is still included in the quota.

### **What is the maximum number of groups I can create?**

The number of groups that you can create is unrestricted, but if you create more than 100,000 groups, you will need to pay certain resource fees. The cumulative group count is a net value. If you disband a group, it is not included in the total group count.

The fees are as follows: if the cumulative group count exceeds 100,000, the excessive part will be charged 149.99 USD/month per 100,000 groups (counts less than 100,000 will be billed as 100,000). For more information, see [Pricing](#).

# SDK

Last updated : 2021-01-04 15:17:24

## Does TUIKit source code support Androidx?

Yes.

## Why do I encounter a 6012 error or "TLSSDK exchange ticket fail" upon login?

- The initialization API and the login API must be called separately and cannot be called successively (because there are asynchronous operations in the initialization method).
- If you are using IM Trial Edition and upgrade to IM Pro Edition, after the upgrade, you can configure your SDKAppID in the console to ensure proper login. For pricing details, see [Pricing](#).

## Why do I encounter a 6013 "SDK Not Initialized" error?

If the 6013 "SDK Not Initialized" error occurs, try the following troubleshooting methods:

1. Check whether you performed other operations such as receiving and sending messages before logging in successfully.
2. Check whether you were forced offline when logging in by an account already logged in on another device. By default, the IM SDK allows one account to log in on one device at a time. For more information, see [Multi-Device Login](#).
3. For Android, verify that all library files have been loaded or repossessed by the system during use.

## Why do I encounter "code: 6205 desc: QALSERVICE not ready"?

This error is caused by calling `stopQALService` after initialization. To correct it, refer to the [configuration method](#) shared by one of our customers.

## Why do not emojis appear or appear as gibberish in messages?

IM does not provide emoji packs. You must align the actual parsing by yourself.

There are two ways to use emojis:

- Use index in TIMFaceElem to identify the index of the emojis. For example, an Android device and an iOS device have the same set of emojis and index 2 represents the smile emoji. When index=2, both devices send and receive the same indexed emoji.
- Use data in TIMFaceElem. For example, the names of emoji images are string-type. Use the word "smile" to represent the smile emoji, and store "smile" in data. Then, both iOS and Android devices use data as the key to retrieve and display the corresponding emoji.

You can also use the two fields at the same time. For example, use data to specify the emoji set and index to indicate the specific index in this emoji set. This way, you can implement many different emoji effects as in QQ. It is possible to store more complex data structures in data as long as the devices follow the same parsing rules.

### **Why do I encounter error 10002 during operations?**

This status code is returned when a network exception, timeout, or ticket switching failure occurs during server-side authentication. If you see this status code, wait a moment and try again.

### **Why do I encounter error 6200 or 6201 when receiving or sending messages?**

This status code is returned when the client is offline, times out, or is disconnected. Error 6200 is defined as no network connection during request sending, whereas error 6201 is defined as no network connection during response sending. When you encounter this status code, check the network condition or wait for the network to restore and try again.

### **Why do I encounter error 20003 when receiving and sending messages?**

This error code is returned when UserID is invalid or UserID has not been imported to IM. To correct it, check whether UserID has been imported to Tencent Cloud.

### **Why do I encounter error 6010 when playing audio messages?**

This error code is returned when audio messages that are no longer retained by the roaming server are requested. To correct it, you can [increase the message roaming duration](#) or download audio files to local storage and play them. However, expired files cannot be restored. Different SDK versions allow you to increase the retention duration for different message types. For more information, see [Message Storage](#).

### **What do I encounter errors 70001, 70003, 70009, and 70013 during account authentication?**

These errors occur because UserID and UserSig do not match. To correct it, check the validity of UserID and UserSig. Error 70001 indicates that UserSig has expired, error 70003 indicates a verification failure due to a truncated UserSig, error 70009 indicates UserSig failed public key verification, and error 70013 indicates a UserID mismatch.

### **Why do I encounter the error code -2 or -5 when the web client app is using the IM SDK?**

- -2: indicates that a web app request to the server failed, usually due to network problems. Web apps send requests to the server by using the HTTP Long Polling method, and this status code is returned when a network problem occurs. To correct it, check your network or try again.

- -5: when the login attempt is not completed and the SDK has not received the a2key and tinyID returned by the server, directly calling other APIs returns the "tinyid or a2 is empty" error message and -5 error code.

## What should I do when the SDK on the armeabi platform reports "java.lang.UnsatisfiedLinkError: No implementation found for"?

Copy jni/armeabi-v7a/libImSDK.so in the IM SDK's aar file to the src/main/jniLibs/armeabi directory in your source code project and load it in build.gradle.

## How can I resolve x86\_64 and i386 architecture errors when the app has been launched in App Store?

This error occurs because App Store does not support the x86\_64 and i386 architecture. The solution is as follows:

1. Clear the project compilation cache:
  - Choose **Product** > **clean**, press Alt until "clean build Folder..." appears, and wait for the operation to complete.
2. Remove the x86\_64 and i386 architecture not supported by App Store:
  - a. Choose **TARGETS** > **Build Phases**.
  - b. Click the plus sign and select **New Run Script Phase**.
  - c. Add the following code:

```
APP_PATH="${TARGET_BUILD_DIR}/${WRAPPER_NAME}"

# This script loops through the frameworks embedded in the application and
# removes unused architectures.

find "$APP_PATH" -name '*.framework' -type d | while read -r FRAMEWORK
do
FRAMEWORK_EXECUTABLE_NAME=$(defaults read "$FRAMEWORK/Info.plist" CFBundleExecutable)
FRAMEWORK_EXECUTABLE_PATH="$FRAMEWORK/$FRAMEWORK_EXECUTABLE_NAME"
echo "Executable is $FRAMEWORK_EXECUTABLE_PATH"

EXTRACTED_ARCHS=()

for ARCH in $ARCHS
do
echo "Extracting $ARCH from $FRAMEWORK_EXECUTABLE_NAME"
lipo -extract "$ARCH" "$FRAMEWORK_EXECUTABLE_PATH" -o "$FRAMEWORK_EXECUTABLE_PATH-$ARCH"
EXTRACTED_ARCHS+=("$FRAMEWORK_EXECUTABLE_PATH-$ARCH")
done
```



```

echo "Merging extracted architectures: ${ARCHS}"
lipo -o "$FRAMEWORK_EXECUTABLE_PATH-merged" -create "${EXTRACTED_ARCHS[@]}"
rm "${EXTRACTED_ARCHS[@]}"

echo "Replacing original executable with thinned version"
rm "$FRAMEWORK_EXECUTABLE_PATH"
mv "$FRAMEWORK_EXECUTABLE_PATH-merged" "$FRAMEWORK_EXECUTABLE_PATH"

done

```

The screenshot shows the Xcode Build Phases tab. A red circle labeled '1' highlights a plus sign (+) in the top left corner. Below it, a list of build phases is shown, including 'Run Script' which is circled in red and labeled '2'. A red arrow labeled '3' points to the script content within the 'Run Script' phase. The script content is as follows:

```

Shell /bin/sh
1 APP_PATH="${TARGET_BUILD_DIR}/${WRAPPER_NAME}"
2
3 # This script loops through the frameworks embedded in the application and
4 # removes unused architectures.
5 find "$APP_PATH" -name '*.framework' -type d | while read -r FRAMEWORK

```

Additional text in the screenshot includes a URL <http://blog.csdn.net/yutaotst> and another URL [https://blog.csdn.net/qq\\_32117417](https://blog.csdn.net/qq_32117417).

3. Package the app and upload it again.

# Account Authentication

Last updated : 2021-02-24 15:45:58

## What should I do if the following issue occurs when I use APIs with JNI?

```
Exception in thread "main" java.lang.UnsatisfiedLinkError:
com.tls.tls_sigcheck.tls_gen_signature_ex2(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)I
at com.tls.tls_sigcheck.tls_gen_signature_ex2(Native Method)
at Demo.main(Demo.java:31)
```

After ruling out the possibility of an incorrect dynamic library path, check whether the package path in the code is correct. API names are generated through the package path during JNI dynamic library compilation. Therefore, do not compile by modifying the package path in `tls_sigcheck.java`, otherwise, the above issue will occur.

## What should I do if the following issue occurs when I use APIs with JNI?

```
Exception in thread "main" java.lang.UnsatisfiedLinkError: *** Can't load IA 32-bit *** on a AMD
64-bit platform
```

This indicates a typical mismatch between JVM and the dynamic library. To correct it, use a 32-bit dynamic library to load a 32-bit JVM.

## How can I generate a UserSig?

For details, see [Generating UserSig](#).

## What is the validity period of UserSig?

As an important credential for IM login authentication, UserSig has a default validity period of 180 days, which can only be modified through the native API. Other APIs and tools cannot modify the validity period. The validity period of UserSig can be set to a maximum of 50 years. You are advised to set the validity period to two months for your account security. For more information, see [Generating UserSig](#).

## Can I delete an account?

- Accounts in the **Pro Edition** cannot be deleted. To deactivate an account, you can call the [v4/im\\_open\\_login\\_svc/kick](#) API to invalidate the login state of the account owner.
- Accounts in the **Trial Edition** can be deleted. You can call the [v4/im\\_open\\_login\\_svc/account\\_delete](#) API to delete accounts that are no longer needed. Note that **user data cannot be recovered after accounts are deleted**.

## Why do I encounter error 70009 upon login?

Check whether the public and private keys match. Do not use the private key of the IM demo to generate UserSig.

## What should I do if the following issues occur when the IM account is not imported?

- **Error 10019 is reported?**

The UserID in the request does not exist. Check whether each `Member_Account` in `MemberList` is correct.

- **Error 10007 is reported?**

You do not have the operation permissions. (For example, a common member in a public group attempts to remove a member from the group, but only the app administrator has the permission to do so.)

- **Error 20003 is reported?**

The UserID of the sender or recipient is invalid or does not exist. Check whether the UserID has been imported into the IM console.

**Note :**

You need to generate accounts and import them into IM via an API call. The accounts can be used only after imported.

# User Profile and Relationship Chain

Last updated : 2021-01-04 15:17:49

## Why did I encounter error code 30001?

There are several possible reasons for the occurrence of error code 30001:

1. A request parameter is invalid.
2. User A initiates a friend request to user B, but user B is already in user A's friend list.
3. User A initiates a friend deletion request to user B, but user B is not in user A's friend list.
4. User A initiates a friend update request to update friend B's relationship chain data, but friend B is not in user A's friend list.
5. User A initiates a request to fetch the relationship chain data of friend B, but friend B is not in user A's friend list.
6. User A initiates a blocklist request to user B, but user B is already in user A's blocklist.
7. User A initiates a request to remove user B from the blocklist, but user B is not in user A's blocklist.

When the IM backend returns an error code, it also returns the detailed error information, and users can identify the cause of the error based on the error information.

## What should I do if error code 30004/40004 is returned when the RESTful API of the profile/relationship chain system is called?

The request packet of the profile/relationship chain system contains a From\_Account field, which is used to indicate the request sender. If the value of the From\_Account field in the request packet is inconsistent with the actual sender, the IM backend considers that the request was initiated by the app backend. In this case, the backend checks whether the request sender has app admin permissions. If the actual sender is not an admin, error 30004/40004 is returned.

## How can I configure settings for messages sent by non-friend users?

To allow non-friend users to send messages, disable the relationship chain for one-to-one chat message verification in the [IM console](#). To forbid non-friend users to send messages, enable the verification. It takes 5 minutes for the setting to take effect.

# Message

Last updated : 2021-01-05 19:23:37

## Large numbers of vendor conflicts occur after the simultaneous integration of IM and TPNS. What should I do?

Currently, IM uses the vendor JAR package provided by [TPNS](#). You can solve this problem by referring to the [IM Offline Push \(Android\)](#) documentation and replacing relevant dependency packages.

## What if a message fails to be received or is lost?

- One-to-one chat message
  - Confirm that the message was sent successfully.
  - Confirm that the recipient has logged in successfully.
  - Confirm that the specified conversation that sent the message is consistent with the recipient's conversation.
- Group message
  - Confirm that the message was sent successfully.
  - Confirm that the recipient has logged in successfully.
  - Confirm that the recipient is a group member.

Whether it's a C2C message or a group message, if the problem cannot be confirmed through the above steps, you need to continue to confirm the following issues:

1. Confirm that a message listener has been registered.
2. Confirm that the sender added `elem` to the message when sending the message (check the return value of `addElement` when sending a message).
3. For Android users, confirm that multiple message listeners have been registered and that `true` was returned in the message listeners.

## What should I do if offline push messages fail to be received?

- APNs

Refer to the offline push (iOS) documentation and confirm the following issues:

- Confirm that the correct certificate has been uploaded to the Tencent Cloud console.
- Confirm that the token was successfully uploaded to Tencent Cloud after successful login.
- Confirm that the correct certificate ID was reported when you reported the token.
- Confirm that foreground/background switch events have been reported correctly.
- Check if the message contains only `TIMCustomElem`, with the `desc` attribute being left empty.

- Check if the deduplication identifiers such as `MsgRandom` were set to the same value, resulting in push failures due to deduplication.
- For a group message, confirm that no alert has been enabled.
- Android  
Refer to the offline push documentation and confirm the following issues:
  - Confirm that the correct push certificate has been uploaded.
  - Confirm that the token has been reported successfully.
  - When not using third-party offline push (Huawei, Xiaomi, or Meizu), confirm that the QALService process is active. If the process is not active, offline push messages will fail to be received. In that case, the auto-start permissions of the system need to be relied on.
  - In the case of multiple processes, confirm that IM SDK initialization was performed only in the main process. If not, you need to modify the configuration to ensure that IM SDK initialization is performed only in the main process.
  - When using third-party offline push (such as Xiaomi, Huawei, and Meizu), you can first use the corresponding third-party console to directly push a message and confirm whether the mobile phone can receive the message. If the mobile phone fails to receive the message, there are two possible reasons:
    - 1) The user failed to correctly integrate third-party offline push. Please refer to the relevant documentation.
    - 2) The mobile phone is not compatible with the offline push. For example, some Huawei mobile phones cannot receive Huawei offline push.
  - When using OPPO offline push, confirm that MasterSecret, instead of AppSecret, was entered in the Android push certificate on the IM console.

When using either APNs push or Android offline push, if the problem cannot be located through the above steps, you need to continue on to confirm the following issues:

1. Confirm that the recipient ID is consistent with the target user ID of the push message.
2. Confirm that you have enabled the offline push listener (Android).
3. Confirm that you have enabled mute notifications. For iOS, refer to the instructions for setting custom push alert sounds. For Android, refer to the instructions for setting a global offline push configuration.
4. Confirm that the message is an online message sent through the `sendOnlineMessage` API or that `MsgLifeTime` was set to `0` during push through a RESTful API.
5. Confirm whether an identifier indicating no offline push has been set for the message. For iOS, refer to the Customizing Offline Message Attributes document; for Android, refer to the instructions for setting the offline push configuration for a single message.

6. If you still cannot identify the problem, you can provide the relevant information to tech support personnel for troubleshooting.

## How is a group @ message processed?

There is no essential difference between an in-group @ message and an ordinary message, although the user specified by @ will see a special UI effect. For example, the message will be highlighted in red on the QQ message list. For the specific implementation, you can refer to the following scheme:

1. When sending a message, listen for keyboard events to detect whether the @ character is input. If it is input, display the group member list on the UI of the sender, allowing the sender to select the @ target user. Here, it is assumed that the selected user is user1.
2. After the @ target user is selected, add @ and the ID of the selected user, for example "@user1", to the message input box.
3. Add a `TIMCustomElem` in the message, and in the `TIMCustomElem`, add a custom message protocol to mark the message as an @ message.

A simple protocol definition can be as follows:

```
{
  "type": "REMIND",
  "target": "user1"
}
```

The sample code for constructing an @ message is as follows (here, the Android platform is used as an example):

```
// Send a text message and specify user1 as the @ target
TIMMessage msg = new TIMMessage();
// Construct the text message elements
TIMTextElem txtElem = new TIMTextElem();
txtElem.setText("@user1 nice to meet u");
if(msg.addElement(txtElem) != 0){
  Log.e(TAG, "add text elem failed");
  return;
}
try{
  // Specify the custom message protocol
  JSONObject remindProto = new JSONObject();
  remindProto.put("type", "REMIND");
  remindProto.put("target", "user1");
  // Construct custom message elements based on the custom protocol
  TIMCustomElem customElem = new TIMCustomElem();
  customElem.setDesc("remind msg");
  customElem.setData(remindProto.toString().getBytes("utf-8"));
}
```

```
if(msg.addElement(customElem) != 0){
Log.e(TAG, "add custom elem failed");
return;
}
}catch(Exception e){
Log.e(TAG, "build custom elem failed");
return;
}
```

### Note :

In the configuration, `TIMTextElem` is not required. If you are sure that foul language filtering is not needed, you can fill the message content of `TIMTextElem` to the `desc` attribute in `TIMCustomElem` .

4. After constructing the message, send the message to the group.
5. After group members receive the message, check whether the message protocol in `TIMCustomElem` is the @ message protocol. If yes, go to the next step. Otherwise, skip it.
6. Check whether the @ target user is consistent with the currently logged-in user. If yes, perform special processing on the UI. Otherwise, no processing is needed.

## How is a red packet message processed?

Red packet messages are similar to @ messages and can be implemented via `TIMCustomElem` . Red packet messages require apps to perform special processing on the UI. For example, if the system detects that the current message is a red packet message, the message should be displayed in the form of a red packet.

In addition, as red packet messages are important messages, we recommend that you set the message priority to high when sending a red packet message. As far as possible, this ensures that the red packet message will reach the target user even if the message frequency limit is reached (currently, the default frequency limit for in-group messages is 40 messages per second, and that for one-to-one chat messages is 5 messages per second).

For more information about message priority, see [Message Priority](#).

### Note :

The payment feature of red packet messages requires manual integration of the corresponding payment SDK. At present, the IM SDK does not provide this payment feature.

The process for constructing a simple red packet message is as follows (Android):



```
// Construct a new message
TIMMessage msg = new TIMMessage();
try{
// Specify the custom message protocol
JSONObject redPacket= new JSONObject();
redPacket.put("type", "RED_PACKET");
redPacket.put("amount", 2018);
redPacket.put("msg", "Happy new year!");

// Construct custom message elements based on the custom protocol
TIMCustomElem customElem = new TIMCustomElem();
customElem.setDesc("red packet");
customElem.setData(redPacket.toString().getBytes("utf-8"));
if(msg.addElement(customElem) != 0){
Log.e(TAG, "add custom elem failed");
return;
}
}catch(Exception e){
Log.e(TAG, "build custom elem failed");
return;
}

// Set the message priority to high
msg.setPriority(TIMMessagePriority.High);
```

## What's the storage period of an IM message?

Historical message storage is available for one-to-one chat messages and non-audio-video group messages. You can log in to the [IM console](#) to modify the relevant configuration. The default configuration for different packages is as follows:

- Trial edition: 7 days, with no extension supported
- Pro edition: 7 days, with extension supported
- Flagship edition: 30 days, with extension supported

The extension of the historical message storage period is a value-added service. For more information, see [Value-added Service Pricing](#).

# Group

Last updated : 2021-01-04 15:18:14

## How do I mute or unmute group members in a group chat?

Muting is a way to control how group members send messages. The muted member cannot send messages during the muting period. For more information on how to mute members, see the following SDK documents:

- [Muting Group Members \(Android\)](#)
- [Muting Group Members \(iOS\)](#)
- [Muting Group Members \(Web\)](#)

Additionally, app admins can mute any member in any group through a RESTful API. For more information, see the [RESTful API: Batch Muting and Unmuting](#).

## How do I query muted members and their muting periods?

Group admins and the group owner can mute or unmute members through the API provided by the IM SDK. To unmute members, set the muting period to 0.

To query information about muted members, you need to query group member profiles. For more information, see the following SDK documents:

- [Obtaining One's Own Profile in a Group \(Android\)](#)
- [Obtaining One's Own Profile in a Group \(iOS\)](#)
- [Obtaining the Group Member List \(Web\)](#)

Additionally, the app admin can query information about muted members through a RESTful API. For more information, see the [RESTful API: Obtaining the List of Muted Members](#).

## How do I view group messages from before I joined the group?

To view historical group messages sent before you joined the group, historical messages for the type of this group must support cloud storage. Based on group types and set use cases, the following configurations apply:

- **Audio-and-video chat rooms (AVChatRoom)** and **broadcasting chat rooms (BChatRoom)** in early versions do not support historical message storage. Therefore, you cannot view messages from before you joined these two types of groups.
- For non-live streaming groups, **work groups (Work)** and **public groups (Public)** by default do not allow members to view historical messages from before they joined the groups, whereas **meeting groups (Meeting)** by default allow members to view historical messages from before

they joined the groups. To modify the default configuration, you can [submit a ticket](#) to apply for modification.

The default retention period of historical messages of non-live streaming groups is 7 days (30 days for Flagship Edition). To extend this period, log in to the [console](#) and choose **Feature Configuration**. Extending the message retention period is a paid value-added service. For more information on billing, see [Pricing](#).

## What are the differences between AVChatRoom and Meeting (ChatRoom in early versions)?

These two types of groups are designed for different scenarios. Specifically, Meeting is suitable for medium-sized groups (with less than 6,000 members), whereas AVChatRoom is suitable for large-scale live-streaming scenarios with unlimited members. The following table lists the major differences between their features:

Group Feature	Meeting	AVChatRoom
Maximum group members	6,000	Unlimited
Group member information	All group member information is saved	Only the information of 300 group members is saved
Group owner can set group admins	Yes	No
Delete group members	The group admin, group owner, and app admin can delete group members.	Not supported
Message roaming	Supported	Not supported
Members can view historical messages from before they joined the group	Yes. Messages within the retention period of historical messages can be viewed by default.	No
Notification on member changes	Not supported	Supported
App admin can import groups	Yes	No

## Why cannot I obtain the values of a group/group member custom field?

Try the following troubleshooting methods:

1. Check whether the custom field is correctly configured in the console.
2. Verify the following in the query request: the requester has the read permission and the group type supports the custom field.
3. Verify whether the configuration request of the custom field has succeeded.
4. For group custom fields:
  - iOS: before logging in to the IM SDK, navigate to TIMManager > setUserConfig > TIMUserConfig > TIMGroupInfoOption > groupCustom to set the relevant configuration.
  - Android: before logging in to the IM SDK, navigate to TIMManager > setUserConfig > TIMUserConfig > TIMGroupSettings > groupInfoOptions > setCustomTags to set the relevant configuration.
5. For group member custom fields:
  - iOS: before logging in to the IM SDK, navigate to TIMManager > setUserConfig > TIMUserConfig > TIMGroupMemberInfoOption > memberCustom to set the relevant configuration.
  - Android: before logging in to the IM SDK, navigate to TIMManager > setUserConfig > TIMUserConfig > TIMGroupSettings > memberInfoOptions > setCustomTags to set the relevant configuration.

# Audio-Video Group

Last updated : 2021-01-04 15:18:28

**If I send a message in which `Message.nick` and `Message.avatar` are both empty, how can the message be processed to normally display the nickname and profile photo on the screen?**

You can call [getMyProfile](#) to obtain your nickname and profile photo.

**How can I mute a group member in a livestreaming group?**

You can use a custom message to mute a specific group member. The custom message must contain the `Members_Account` of the group member to be muted and the muting time. Send the custom message to the business backend by calling the [Callback Before Delivering Group Messages](#). The business backend will call [Muting and Unmuting Group Members](#) to mute the group member.

**How can I remove a group member from a livestreaming group?**

You can use a custom message to remove a specific group member. The custom message must contain the `Members_Account` of the group member to be removed. Set the priority of the custom message to High to prevent the message from being discarded by the backend due to the message sending frequency limit of 40 messages per second. After the SDK receives the message, it calls the [kickGroupMember API](#) to remove the group member from the group.

**A group member quits a livestreaming group on the Android, iOS, or PC client when the API for quitting a group is called on the WeChat Mini Program or web client. However, when the API for quitting a group is called on the Android, iOS, or PC client, the group member does not quit the livestreaming group on the WeChat Mini Program or web client. Why is this the case?**

The WeChat Mini Program and web client allow users to join livestreaming groups as guests. When the API for quitting a group is called on the Android, iOS, or PC client, the WeChat Mini Program or web client will not proactively trigger the quit group operation.

- To ensure synchronized group quitting on all clients, configure the [Callback After a Group Member Drops Out](#) and determine the quit platform based on the `OptPlatform` field. When the quit platform is Android, iOS, or PC, call [Send a One-to-One Message](#) to send a custom message to the group member who quits the group. The frontend screens the conversation and does not display it on the UI. After the WeChat Mini Program or web client receives the message, it calls the [quitGroup API](#).
- To ensure independent group quitting on different clients, configure [Callback After a Group Member Drops Out](#) and determine the quit platform based on the `OptPlatform` field. Call [Send a](#)

[One-to-One Message](#) to send a custom message to the group member who quits the group. The frontend screens the conversation and does not display it on the UI. After a non-quit platform receives the message, it calls the [joinGroup API](#) to join the group. To prevent multiple system notifications for joining and quitting a group, you can submit a ticket to disable system notifications for joining and quitting a group.

## Why are messages lost?

Messages can be lost due to the following conditions:

- Livestreaming groups have a message sending frequency limit of 40 messages per second. You can determine whether a message is discarded due to the frequency limit based on the callbacks before and after delivering group messages. If a message receives the callback before delivering group messages but does not receive the callback after delivering group messages, the message is discarded due to the frequency limit.
- When a group member quits a livestreaming group from the WeChat Mini Program or web client, the member may also quit on the Android, iOS, or PC client. For more information, see [FAQ 4](#).
- If the WeChat Mini Program or web client encounters an exception, check whether your SDK version is earlier than v2.7.6. If yes, upgrade your SDK to the latest version.

If the fault persists, submit a ticket to contact us.

## How can I compile statistics on likes and follows?

Customize the like or follow message type. When a user clicks the like or follow icon on the frontend to deliver a custom message, send the like or follow message to the business side by calling the [Callback Before Delivering Group Messages](#). The business side counts the number of received like and follow messages and updates the data to the group profile field every 3 to 5 seconds by calling [Modify Basic Group Profiles](#). The SDK calls the [getGroupsInfo API](#) to count the number of like or follow messages.

## How can I properly set message priorities?

To prevent important messages from being discarded, a livestreaming group provides three message priorities for all messages. The SDK preferentially obtains high priority messages. We recommend that you set the priorities of custom messages as follows:

- High: red packets, gifts, and messages for removing group members
- Normal: common text messages
- Low: likes and follows

**Are there any open-source livestreaming components that can be directly used to watch videos and chat?**

Yes. The code is also open-source. For more information, see [Tencent Cloud TWebLive](#).