

Chat FAQs 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

FAQs

uni-app FAQs

Purchase

SDK

SDK

Account Authentication

User Profile and Relationship Chain

Message

Group

Audio-Video Group

Nickname and Profile Photo

FAQs

uni-app FAQs

Last updated : 2025-03-10 11:17:07

FAQs

What is UserSig?

UserSig is a password with which you can login to Chat service. It is the ciphertext generated by encrypting information such as userID.

How can I generate a UserSig?

The issuance method for UserSig involves integrating the calculation code of UserSig into your server, and providing an interface oriented towards your project. When UserSig is needed, your project sends a request to the business server to access the dynamic UserSig. For more information, please see [generate UserSig](#).

Caution

The method for accessing UserSig adopted in the quick-start guide of chat-uikit-uniapp is to configure a SECRETKEY in the local code. In this method, the SECRETKEY is susceptible to decompilation and reverse engineering. In the event that your key is leaked, attackers can steal your Tencent Cloud traffic. Therefore, **this method is only suitable for locally running function debugging**. For the correct method of issuing UserSig, please refer to the above text.

TUIKit uses TypeScript for development. Is it possible to integrate it into a JavaScript project?

This project uses TypeScript for development but also supports direct integration into JavaScript projects.

Is integration possible in a project created with vue-cli?

Currently, projects created with vue-cli are not supported.

What should I do if the audio message time display error occurs when building a native app with uni-app?

When building a native app with uni-app, the `recorderManager.onStop` callback does not include `duration` and `fileSize`. Users are required to supplement the duration and fileSize.

Record the time using `setInterval`, to calculate duration.

Locally calculate the file size, where the file size = (audio bit rate) × time length (in seconds) / 8, for a rough estimate.

For detailed code, please refer to the implementation of [uni-app TUIKit source code](#).

Caution

The Voice Message Service object must include attributes `duration` and `fileSize`, without `fileSize`, the duration of the Voice Message Service will display as an incorrect numeric string.

How can we implement the uploading of file messages?

[uni-app official documentation](#) suggests that `chooseFile` does not support APP choosing non-media files.

Therefore, customers should choose the appropriate plugin from the [plugin market](#) based on their specific needs.

Refer to the available plugins in the plugin library: [iOS Upload File Plugin](#), [Android Upload File Plugin](#) (unofficial) to complete file selection and format data in accordance with the file message data format requirements.

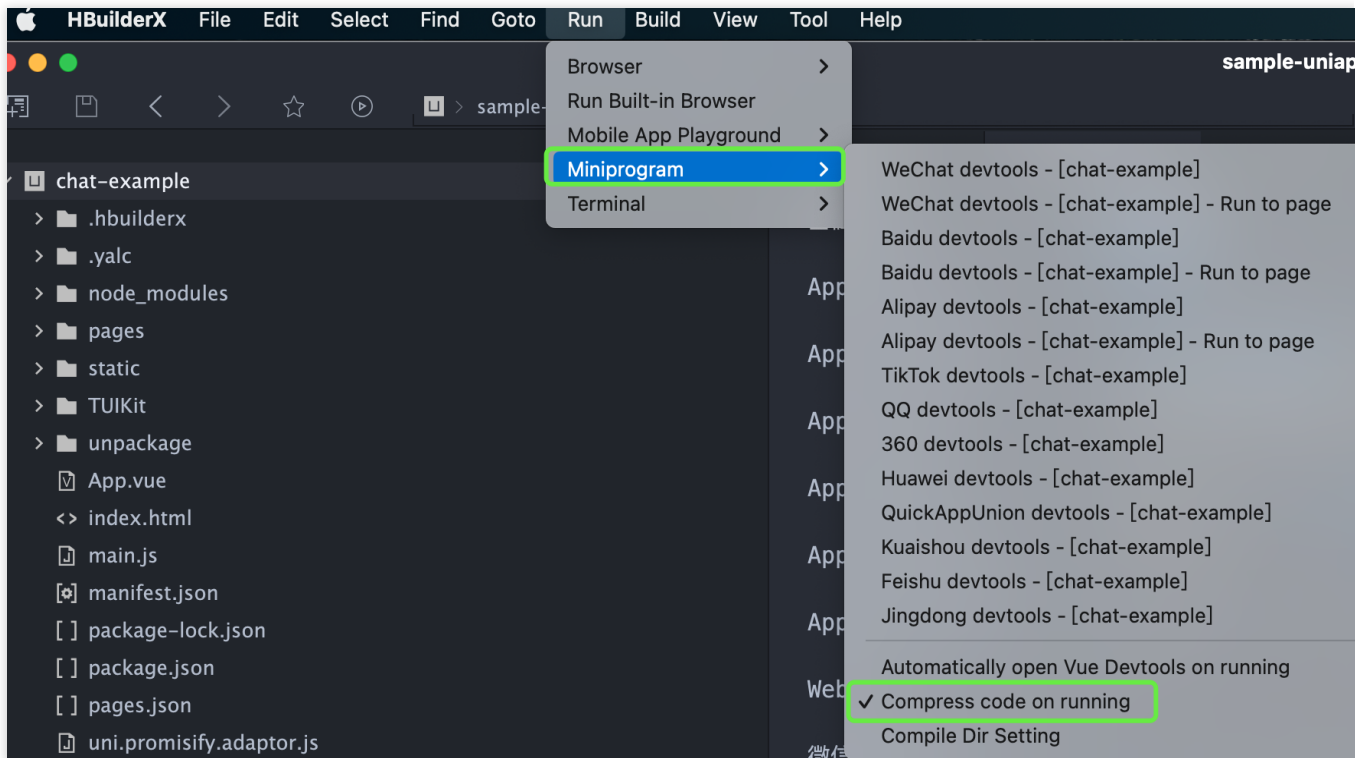
The file parameters are as follows:

Parameter	Type	Meaning
name	String	File Name (Necessary)
size	Number	File size (essential, cannot be 0)
type	String	File type (essential)
path	String	File Path (Necessary)
lastModified	String	Modification Time (Not Necessary)

Refer to the example format:

```
const fileData = {
  files: [
    {
      name: '', // File name
      size: 10, //Local file size
      type: 'pdf', //File type
      path: '', //Local file path
      lastModified: '', // Modification time
    },
  ],
};
```

How can I elect to compress the code during runtime in a mini program?



What should I do when an exception error occurs while running in a mini program?

The issue may relate to the version of WeChat Developer Tools. Please use the latest developer tools, and ensure you have a stable debug library version.

What should I do if an error like this occurs when introducing native audio and video plugins?

```

78 TIM 09:46:50.460:ModuleManager._startMainThreadTimer
09 TIM 09:46:50.463:NetMonitorModule.start networkType:wifi __INFO
39 TIM 09:46:50.465:SignModule.login userID:ML
71 [Vue warn]: Error in v-on handler: "TypeError: undefined is not an object (evaluating 'uni.$TUICalling.login')"
04 (found at pages/TUI-Login/login.vue:1) __ERROR
35 TypeError: undefined is not an object (evaluating 'uni.$TUICalling.login') __ERROR
66 TIM 09:46:50.652:ProtocolHandler.getKeyMap unknown protocolName:push_imsdk_purchase_bitsv2 __WARN
97 TIM 09:46:50.682:SignModule.login ok. helloInterval:120 instanceID:247147882 timeStamp:1642038410
28 TIM 09:46:50.110:baseTime from server: Thu Jan 13 2022 09:46:50 GMT+0800 (CST) offset: -576 __INFO

```

Make a [custom base](#) in accordance with the uni-app's [native plugin debugging guide](#).

What should I do if HBuilder reports an error: SyntaxError: Unexpected token '!'?

If the following error message appears after following the above access steps, it means that your current version of HBuilder is too low, please upgrade to the latest version.

```
10:48:05.229
10:48:06.118
10:48:06.122
10:48:06.123
10:48:06.435 ERROR Error loading vue.config.js:
10:48:06.439 ERROR SyntaxError: Unexpected token '.'
10:48:06.439
10:48:06.444   const setup = (plugin) => plugin.esbuild?.setup ?? ((build2) => {
10:48:06.444                                     ^
10:48:06.448 SyntaxError: Unexpected token '.'
10:48:06.456   at wrapSafe (internal/modules/cjs/loader.js:915:16)
10:48:06.457   at Module._compile (internal/modules/cjs/loader.js:963:27)
10:48:06.465   at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
10:48:06.465   at Module.load (internal/modules/cjs/loader.js:863:32)
10:48:06.473   at Function.Module._load (internal/modules/cjs/loader.js:708:14)
10:48:06.480   at Module.require (internal/modules/cjs/loader.js:887:19)
10:48:06.481   at require (internal/modules/cjs/helpers.js:74:18)
10:48:06.488
10:48:06.498   at Module._compile (internal/modules/cjs/loader.js:999:30)
10:48:06.507   at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
```

How should I handle it if, after building the uniapp, I get a white screen and a prompt that says: '[Vue warn]: Property or method "isPC" is not defined'?

Please refer to the documentation: [Getting Started \(uniapp\)](#).

After considering the aforementioned content, proceed to import `VueCompositionAPI` in `main.js` file, and apply `Vue.use(VueCompositionAPI)`, as demonstrated below:

```
import VueCompositionAPI from "@vue/composition-api";
Vue.use(VueCompositionAPI);
```



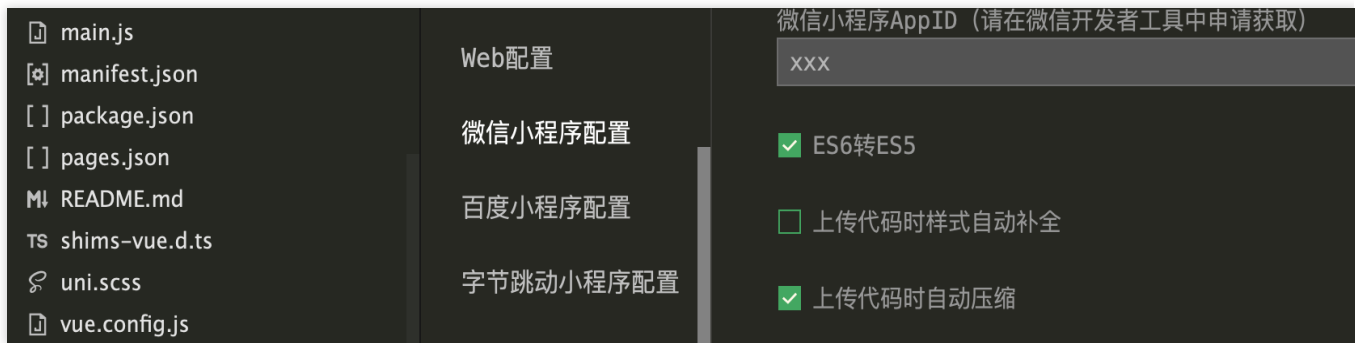
Once the applet is integrated according to the documentation, clicking "Open TUIKIT Conversation" will initiate a conversation, leaving the rest of the content whitewashed, devoid of a conversation list. How does one resolve it?

Please refer to the documentation: [quick start \(uniapp\)](#) TUIKIT Source Code Integration/Step 2: Download and Introduce TUIKit/Integrate TUIKit Components . Strictly adhere to this section's four steps in sequential order for integration, especially the final item "Miniprogram Subpackage Home Page".

Note:

The mini program defaults to integrate via the subpackage process in order to prevent the main package from being oversized. Hence, if you wish to package a mini program, please do not skip the configuration of the "Miniprogram Subpackage Home Page".

To package a mini program, please check the following options under project manifest.json > WeChat mini-program configuration



What should be done in the event the mini program needs to be launched or deployed to a production environment?

Please perform domain configuration in **WeChat Official Platform > Development > Development Management > Development Setting > Server Domain Name**:

Beginning from v2.11.2, the SDK supports WebSocket, and the subsequent versions must append the following domain names to the **valid socket domains**:

Domain name	Description	Is Mandatory
<code>wss://wss.im.qqcloud.com</code>	Web Chat service domain	Essential
<code>wss://wss.tim.qq.com</code>	Web Chat service domain	Essential

Add the following domains to the **request for valid domains**:

Domain name	Description	Is Mandatory
<code>https://web.sdk.qqcloud.com</code>	Web Chat service domain	Essential
<code>https://webim.tim.qq.com</code>	Web Chat service domain	Essential
<code>https://api.im.qqcloud.com</code>	Web Chat service domain	Essential

Add the following domain name to the **uploadFile valid domain name**:

Domain name	Description	Is Mandatory
<code>https://cos.ap-shanghai.myqcloud.com</code>	File upload domain	Essential
<code>https://cos.ap-shanghai.tencentcos.cn</code>	File upload domain	Essential
<code>https://cos.ap-</code>	File upload domain	Essential

guangzhou.myqcloud.com		
------------------------	--	--

Add the following domain name to **downloadFile** legal domain name:

Domain name	Description	Is Mandatory
<code>https://cos.ap-shanghai.myqcloud.com</code>	File download domain	Essential
<code>https://cos.ap-shanghai.tencentcos.cn</code>	File download domain	Essential
<code>https://cos.ap-guangzhou.myqcloud.com</code>	File download domain	Essential

Reference Documentation

Related to UIKit (vue2 / vue3):

[Source code of chat-uikit-uniapp \(vue2/vue3\) on GitHub](#)

[Quick start \(uniapp\)](#)

Regarding ChatEngine:

[ChatEngine API](#)

[ChatEngine npm](#)

Technical Consultation

[Click here to join the Chat community](#), where you'll receive support from experienced engineers to help overcome your challenges.

Purchase

Last updated : 2025-03-10 11:17:07

Do Chat Standard edition and Pro edition have a MAU limit?

Standard edition and Pro edition、Pro Plus edition、Enterprise edition do not have an upper limit on MAU. The first 10,000 MAU per month is free. The MAU fee is pay-as-you-go on a monthly basis. Fees incurred within a month will be deducted on the third day of the next month.

How is MAU (Monthly Active Users) calculated?

MAU calculates the number of unique users that log in to app in a given month. After a user logs in successfully by calling the login function of the SDK to establish a persistent connection with the backend, MAU will be increased by one. A user that logs in repeatedly in the same month is considered as one user only for MAU. Use the login function appropriately according to business scenarios to avoid an excessively high MAU. In addition, if a user simultaneously uses Chat, Live, Call, or any combination of the products, they will only be counted once as one MAU.

How are free Audio & Video Minutes and free MAU calculated in multiple subscription packages with the same SDKAppID ?

The calculation of our audio and video hours and MAUs with the same SDKAppID follows an aggregation logic. If you purchase multiple products such as Chat, Live, Call or Conference, the resources in each product package are added together. For example, if you have purchased the Live Standard package for the same SDKAppID, which includes 10,000 free MAUs and 300,000 free audio/video minutes, and the Call 1v1 package, which includes 5,000 free MAUs and 100,000 free audio/video minutes, you have 15,000 free MAUs and 100,000 free audio/video minutes for the same month. If you have purchased the Call 1v1 package, which includes 5000 free MAUs and 100,000 free minutes, your free MAUs for the month will be 15,000 and your free minutes will be 400,000.

What should I do if I received an error when creating 100 accounts in Chat?

You can create up to 100 accounts in the Free Trial. 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 [v4/im_open_login_svc/account_delete](#) API to delete unwanted accounts in the Free Trial. **User data cannot be recovered once the account is deleted**, so proceed with caution.

Does Chat support global access?

Chat provides reliable, secure network connections with global coverage. With its proprietary multi-level optimal addressing algorithm, Chat can perform scheduling across the entire network. When terminals log in, the Chat SDK connects to the nearest access nodes. Chat access nodes include the following:

China: South China, North China, East China, Hong Kong, and Taiwan.

Other regions:

Asia: Singapore, Indonesia, UAE, Thailand, Japan, Vietnam, South Korea, and Philippines.

Europe: United Kingdom, Netherlands, Germany, Italy, Norway, France and Spain.

South America: Brazil.

North America: United States, Canada, and Mexico.

Oceania: Australia.

Africa: South Africa and Nigeria.

Chat allows the creation of up to 50 audio-video groups. How many audio-video groups can be created after I created 3 and then deleted 3?

If you created 3 audio-video groups and then deleted all of them, you can still create 50. The number of audio-video groups you can create is a net value. Deleted groups do not count against the quota. If you have an audio-video group that has no members but has not been deleted, it is still included in the quota.

Is the Call trial version free to use? Will there be any additional charges during use?

The trial version of the Call only provides you with **free usage rights for the Call function**. For specific available features, please refer to the [Call](#). The call duration usage generated during your use of this function will be billed according to the [Pay-As-You-Go](#), while chat costs will be billed based on the [Billing Overview](#) of Chat. In addition, Tencent Cloud accounts (UIN) that activate the Audio and Video Call feature will receive a free 10,000-minute duration package every month, which can be used to deduct the call duration usage generated during use.

Is it impossible to activate and use Call when Chat or TRTC applications are disabled?

No, the Call capabilities are built upon the support from Chat and TRTC. Therefore, it is necessary to ensure the availability of both your Chat and TRTC applications. If either your Chat or TRTC application is disabled, you need to enable the application first before activating/using the Call.

After successfully testing the Call using the free trial version of the SDK, is it necessary to update the SDK when switching to and purchasing the official package for launch?

No, you do not need to update the SDK. All you need to do is purchase the specified official package, and there is no need to change any SDK-related configurations.

SDK

SDK

Last updated : 2025-03-10 11:23:21

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 Chat Free edition and upgrade to Chat 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 Chat 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" error?

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

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

Chat 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 Chat. To fix 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 fix 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](#).

Why do I encounter errors 70001, 70003, 70009, and 70013 during account authentication?

These errors occur because UserID and UserSig do not match. To fix it, check the validity of UserID and UserSig. Among them, error 70001 means UserSig has expired, error 70003 means a verification failure due to a truncated UserSig, error 70009 means UserSig verification failed (possibly due to mixing the keys or private keys of other SDKAppIDs when generating UserSig), and error 70013 means UserID does not match.

Why do I encounter the error code -2 or -5 when the web client app is using the Chat 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 fix 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 Chat 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:

Select **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. Select **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=()

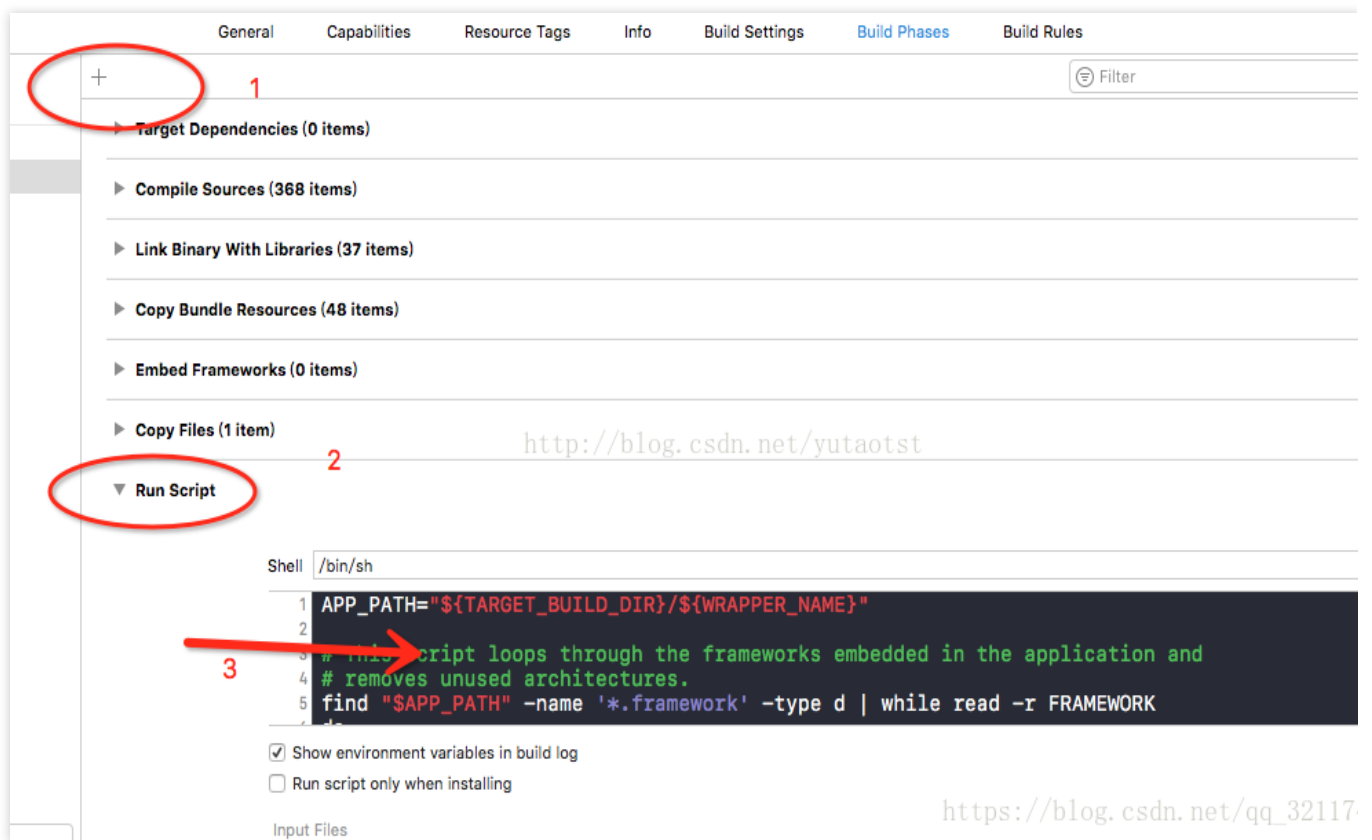
    for ARCH in $ARCHS
    do
        echo "Extracting $ARCH from $FRAMEWORK_EXECUTABLE_NAME"
        lipo -extract "$ARCH" "$FRAMEWORK_EXECUTABLE_PATH" -o
"$FRAMEWORK_EXECUTABLE_PATH-$ARCH"
        EXTRACTED_ARCHS+=("$FRAMEWORK_EXECUTABLE_PATH-$ARCH")
    done

    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"
```

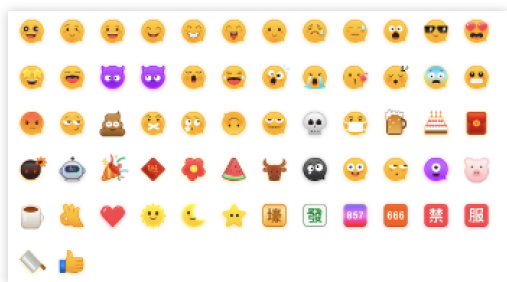
done



3. Package the app and upload it again.

Can developers directly utilize the emoticon packs demonstrated in the Chat various client demos/TUIKit projects?

To respect the copyright of emoji designs, the Chat Demo/TUIKit project does not include cutouts of large emoji elements. Please replace them with your own designs or other emoji packs for which you hold the copyright before officially launching for commercial use. **The default smiley face emoji pack shown below is copyrighted by Tencent RTC** and is available for licensed use for a fee. If you need to obtain a license, please [contact us](#).



Account Authentication

Last updated : 2025-03-10 11:17:07

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 fix 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 Chat 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](#).

Why do I encounter error 70009 upon login?

UserSig verification fails, which may be caused by mixing keys or private keys of other SDKAppIDs when generating UserSig. Please use the key or private key under the corresponding SDKAppID to regenerate UserSig.

What should I do if the following issues occur when the Chat 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 Chat console.

Note:

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

User Profile and Relationship Chain

Last updated : 2024-08-21 17:47:54

Why do 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 contacts data, but friend B is not in user A's friend list.
5. User A initiates a request to fetch the contacts 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 Chat 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/contacts system is called?

The request packet of the profile/contacts 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 Chat 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 contacts for one-to-one chat message verification in the [Chat 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 : 2025-05-08 17:32:23

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

One-to-one chat message

Check that the message was sent successfully.

Check that the recipient has logged in successfully.

Check that the specified conversation that sent the message is consistent with the recipient's conversation.

Group message

Check that the message was sent successfully.

Check that the recipient has logged in successfully.

Check that the recipient is a group member.

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

1. Check that a message listener has been registered.
2. Check 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, check 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?

Please refer to the [Push FAQ](#).

What's the storage period of an Chat message?

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

Free Trial: 7 days, with no extension supported.

Standard edition: 7 days, with extension supported.

Pro edition: 30 days, with extension supported.

Pro Plus edition: 90 days, with extension supported.

Enterprise edition: 90 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](#).

Why does it show that messages are sent successfully even though the senders are blocklisted?

When you enable **Show "Sent successfully" After Sending Messages** in the [Blocklist Check](#) section in the Chat console, blocklisted users will be prompted that the messages are sent successfully when they send messages, but the recipients will not receive the messages. When you disable this feature, blocklisted users will be prompted that message sending fails when they send messages and the SDK will receive [error code 20007](#). For more information, see the **Blocklist check** section in [Feature Configuration](#).

How to convert an image address into the downloadable domain name format by using COS?

You need to upload the image yourself. If the image uses COS private read, you need to get a signed URL to obtain the download permission ([Download via Pre-Signed URL](#)).

What are the rules for unique identification of Chat messages?

Chat uses `sg_id`, `msgID`, and `msgKey` as the unique identifiers of one-to-one and group chat messages.

Different types of Chat messages and SDKs use different unique identifiers:

For client SDKs, one-to-one and group chat messages are identified by `msg_id` in the following format: tinyid-clientTime-random.

For web SDKs of v2.17.0 or earlier, one-to-one and group chat messages are identified by `msgID` in the following format: Conversation ID-msgSeq-random-1 (messages sent by me) / 0 (messages sent by others).

For web SDKs of v2.18.0 or later, one-to-one and group chat messages are identified by `msgID` in the following format: tinyid-clientTime-random.

For server SDKs, one-to-one messages are identified by `msgKey` in the following format: clientSeq_random_serverTime.

For server SDKs, group chat messages are identified by "Group ID + msgSeq".

Do audio-video groups and community groups support @ messages?

Audio-video groups do not support sending @ messages. Community groups support @ a single user but do not support @ all users.

What is the difference between Message modification, Message extension, and Message reaction ?

Message function	Whether Ultimate edition is required	Storage method	Pull method	Will concurrency settings conflict?
Message modification	Not required	Modify information is stored directly in the message	Modify information is returned directly	There will be conflicts

		body, sharing message storage, with a maximum of 12K per message	through the message body, no need for separate pull	
Message extension	Required	Extension information is stored independently, with a maximum of 300 extensions per message, and a maximum of 1K per extension	Extension information needs to be pull separately, only supports single message pull	The same extension settings will conflict, different extension settings will not conflict
Message reaction	Required	Reaction information is stored independently, with a maximum of 10 reactions per message, and a maximum of 100 users per reaction	Reaction information needs to be pull separately, supports batch pull of multiple messages	No conflicts

Message modification :

You can call the `modifyMessage` ([Android](#) / [iOS & Mac](#) / [Windows](#)) interface to modify the original Message content.

Advantage: Simple and easy to use.

Disadvantage:

If multiple users modify a message at the same time, it may cause modify conflicts.

If the modified message content is too large, it may cause the message size to exceed the limit (maximum 12K per message).

Summary: Message modification is suitable for single-user operations and small storage requirements, such as order message scenarios, where the order status can be stored and modified through the `cloudCustomData` field of the modified message.

Message extension :

You can call the `setMessageExtensions` ([Android](#) / [iOS & Mac](#) / [Windows](#)) interface to add extension information to a message.

Advantage :

Multiple extensions can be set for a single message, and different extensions can be set simultaneously without conflict.

Extension information is stored independently and does not occupy the message body size, with a maximum of 300 extensions per message and a maximum of 1K per extension.

Disadvantage :

There will be conflicts when setting the same extension simultaneously.

Extension information needs to be pulled separately using the `getMessageExtensions` ([Android](#) / [iOS & Mac](#) / [Windows](#)) interface.

Extension information does not support batch pull of multiple messages.

Summary: Message extension is suitable for multi-user operations and large storage scenarios, such as Questionnaire survey scenarios, where each extension information is a collection of {userID: Questionnaire survey content}, and different users filling out the questionnaire survey will not conflict.

Message reaction :

You can call the `addMessageReaction` ([Android](#) / [iOS & Mac](#) / [Windows](#)) interface to add reaction information to a message.

Advantage :

Multiple reactions can be added to a single message, and reactions can be added without conflict.

Reaction information is stored independently and does not occupy the message body size, with a maximum of 10 reactions per message and a maximum of 100 users per reaction.

Reaction information supports batch pull of multiple messages.

Supports separate pull of user list using the same reaction.

Disadvantage :

Reaction information needs to be pulled separately using the `getMessageReactions` ([Android](#) / [iOS & Mac](#) / [Windows](#)) interface.

Reaction information only supports setting Reaction ID, other information is not supported, and the SDK will automatically count the use of the same reaction.

Summary: Message reaction is suitable for scenarios that require counting user lists, such as emoji reaction scenarios. Users only need to pass in the Emoji ID when adding an emoji reaction, and the SDK will automatically count the user list using the same Emoji ID.

Group

Last updated : 2025-03-10 11:17:07

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 Chat 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 Ultimate 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	No member information 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. Check the following in the query request: the requester has the read permission and the group type supports the custom field.
3. Check whether the configuration request of the custom field has succeeded.
4. For group custom fields:
iOS: Before logging in to the Chat SDK, navigate to TIMManager > setUserConfig > TIMUserConfig > TIMGroupInfoOption > groupCustom to set the relevant configuration.
Android: Before logging in to the Chat 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 Chat SDK, navigate to TIMManager > setUserConfig > TIMUserConfig > TIMGroupMemberInfoOption > memberCustom to set the relevant configuration.

Android: Before logging in to the Chat SDK, navigate to `TIMManager > setUserConfig > TIMUserConfig > TIMGroupSettings > memberInfoOptions > setCustomTags` to set the relevant configuration.

Audio-Video Group

Last updated : 2024-02-07 17:33:32

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.

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 leaves a livestreaming group from the WeChat Mini Program or web client, the member may also leave 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 problem 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](#).

Nickname and Profile Photo

Last updated : 2024-02-07 17:33:31

Nickname and Profile Photo Update in Conversations

Conversations do not store nicknames and profile photos. Therefore, the SDK needs to obtain them from user profiles or group profiles saved in the local storage and use them in conversations. For a one-to-one conversation, the SDK obtains and uses the other user's nickname and profile photo. For a group conversation, the SDK obtains and uses the group name and group profile photo. To keep local user profiles and group profiles up-to-date, the latest SDK version has the following optimizations:

Ono-to-one chats:

1. When a user actively obtains a conversation or the SDK calls back a conversation update, if the SDK detects that the local storage does not have the other user's profile, it will synchronize the user's profile from the server and save it locally.
1. If a friend's profile is changed, the SDK will receive a notification of the change from the backend and update the friend's profile locally. If a stranger's profile is changed, the backend will not send notifications, so the stranger's profile will not be updated locally. However, you can call the `getFriendsInfo` API to get strangers' profiles and update them locally.
1. When sending a message, the backend will attach the user's latest nickname and profile photo to the message body. After receiving the message, the SDK will update the nickname and profile photo locally if the user's profile already exists in the local storage.

In summary, the SDK will ensure that the nicknames and profile photos in conversations with friends are up-to-date. While with strangers, the SDK cannot guarantee that and you need to actively pull the nicknames and profile photos if needed.

Group chats:

1. After you join a group successfully, the SDK will actively obtain the group profile and save it locally.
1. When the profile of a group that you have joined is modified, the backend will notify the client, which will then update the group profile locally in time.

In summary, the SDK will ensure that the nicknames and profile photos in conversations are up-to-date for groups you have joined, but not for groups that you haven't joined or have left.

Caution:

The change of user profiles or group profiles do not trigger conversation update. Therefore, the nicknames and profile photos in a conversation will not be updated until the next conversation operation is performed (such as actively obtaining the conversation, marking the conversation as read, and sending/receiving messages).

There is no relationship chain between you and a stranger. If the stranger's profile is changed, the backend will not send notifications, and the stranger's local profile will not be updated. It will be updated only when you actively pull the

stranger's profile.

Nickname and Profile Photo Update in the Message List

Messages store nicknames and profile photos. In order to keep the nicknames and profile photos of messages up-to-date, the latest SDK version has the following optimizations:

1. When sending a message, the backend will attach the user's latest nickname and profile photo to the message body. After receiving the message, the SDK will update the nickname and profile photo locally if the user's profile already exists in the local storage.
2. After a message is successfully sent and received, the nickname and profile photo stored in the message cannot be modified. In order to get the new nicknames and profile photos for historical messages, the SDK will first query local user profiles when trying to obtain the nickname and profile photo fields via messages. If local user profiles exist, local nicknames and profile photos will be returned. (According to optimization 1, local nicknames and profile photos will be updated in real time according to new messages.) If not, the nicknames and profile photos in message bodies will be returned.

In summary, the SDK will ensure that the nicknames and profile photos of new messages are up-to-date, but not those of historical messages. The nicknames and profile photos of historical messages will be updated only when the senders' profiles exist in the local storage.

FAQs

According to the above description, the latest SDK version has a lot of optimizations regarding nickname and profile photo. If you encounter any nickname and profile photo issues, please first upgrade your SDK to the [latest version](#) and check whether the issues are solved. If not, please see the following FAQs:

Why isn't the conversation updated immediately when the nickname and profile photo are changed?

The change of nicknames and profile photos do not trigger conversation update. Therefore, the nickname and profile photo in a conversation will not be updated until the next conversation operation is performed (such as actively obtaining the conversation, marking the conversation as read, and sending/receiving messages). However, you can listen for friend or group profile update notifications to update the nicknames and profile photos in conversations.

Why isn't the conversation updated when the stranger's nickname and profile photo are changed?

There is no relationship chain between you and a stranger. If the stranger's profile is changed, the backend will not send notifications, and the stranger's local profile will not be updated. It will be updated only when you actively pull the

stranger's profile or receive a message from the stranger. The message will carry the stranger's latest nickname and profile photo, and the SDK will update them locally after receiving the message.

Why can't the nicknames and profile photos of historical messages be updated?

Please refer to [optimization 2](#) in **Nickname and Profile Photo Update in the Message List**. If the local storage does not have the senders' profiles, the nicknames and profile photos of historical messages won't be updated. They will be updated only when you actively pull the senders' profiles.

Why doesn't the SDK actively pull user profiles when conversations are updated, or messages are received?

Conversation update and message sending/receiving are both frequent events. Synchronizing user profiles every time will cause huge pressure on the client and backend, seriously affecting application performance. We don't recommend that you actively pull user profiles in both cases, because it also affects application performance. The recommended approach is to pull user profiles when users click on the profile photos of messages.