

Game Multimedia Engine

Getting Started

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

Getting Started

Voice Service Activation Guide

Operation Guide

Native SDK Quick Access

Getting Started

Voice Service Activation Guide

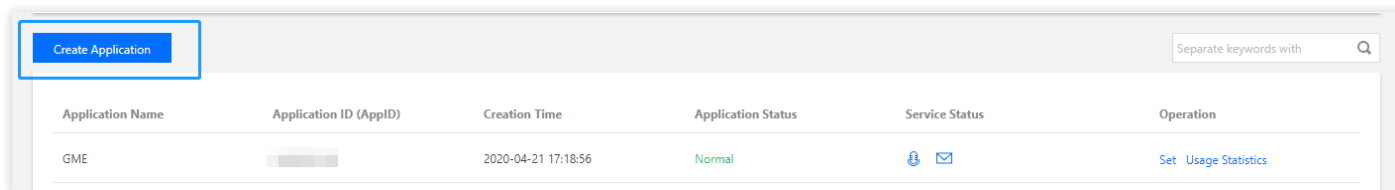
Last updated : 2021-07-23 14:57:10

This document describes how to apply for the SDKs of voice services for Tencent Cloud GME.

Creating Services

Creating an application

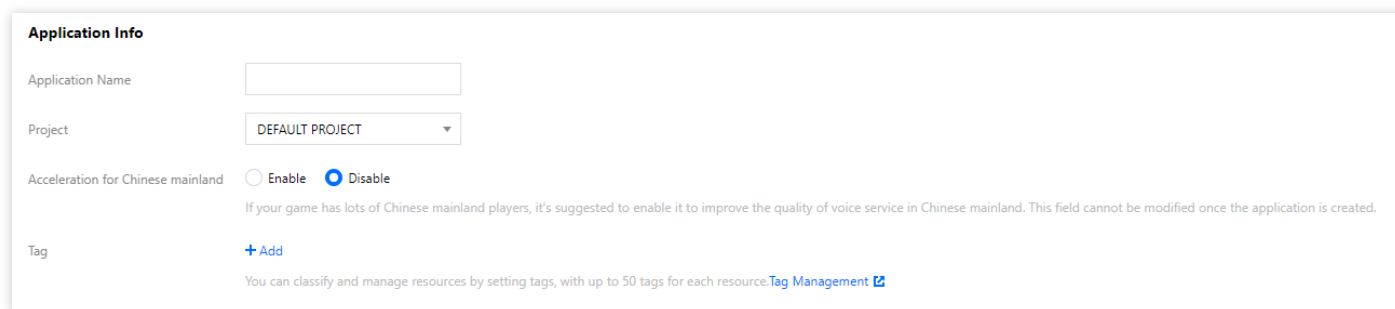
1. Log in to the [GME console](#) and click **Service Management** on the left sidebar to go to the **Service Management** page.
2. Click **Create Application**, enter the application information, and enable the services as needed.



The screenshot shows the GME console interface. At the top left, there is a blue button labeled 'Create Application'. To the right, there is a search bar with the text 'Separate keywords with' and a magnifying glass icon. Below these elements is a table with the following columns: Application Name, Application ID (AppID), Creation Time, Application Status, Service Status, and Operation. The table contains one row with the following data: Application Name: GME, Application ID (AppID): [redacted], Creation Time: 2020-04-21 17:18:56, Application Status: Normal, Service Status: [icons], and Operation: [Set] [Usage Statistics].

| Application Name | Application ID (AppID) | Creation Time | Application Status | Service Status | Operation |
|------------------|------------------------|---------------------|--------------------|----------------|--------------------------|
| GME | [redacted] | 2020-04-21 17:18:56 | Normal | [icons] | [Set] [Usage Statistics] |

3. Enter the application information.

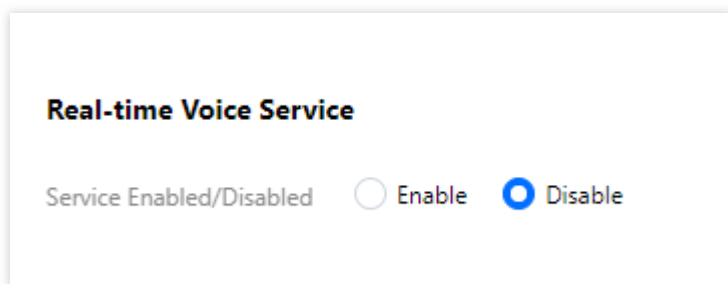


The screenshot shows the 'Application Info' form. It has the following fields and options:

- Application Name: A text input field.
- Project: A dropdown menu with 'DEFAULT PROJECT' selected.
- Acceleration for Chinese mainland: Radio buttons for 'Enable' and 'Disable', with 'Disable' selected. Below this is a note: 'If your game has lots of Chinese mainland players, it's suggested to enable it to improve the quality of voice service in Chinese mainland. This field cannot be modified once the application is created.'
- Tag: A '+ Add' button. Below this is a note: 'You can classify and manage resources by setting tags, with up to 50 tags for each resource. [Tag Management](#)'.

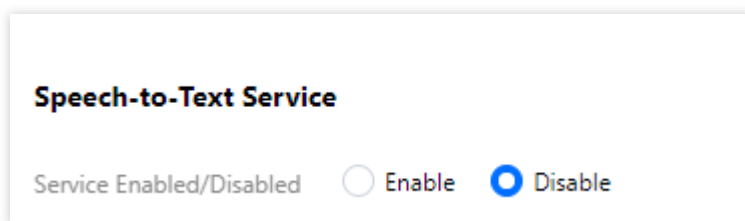
- Application Name: enter the application name, which will be displayed in the application list.
 - Project: **DEFAULT PROJECT** is selected by default. You can also select the created projects. For how to create a project, see [Project Management](#).
 - Tag: click **Add** to add tags. For how to create a tag, see [Tag Management](#).
4. Enable or disable the **Real-time Voice Service**.
Real-time Voice Service is billed by voice duration. You can enable it as needed and purchase in

the [Purchase Page](#).



5. Enable or disable the **Speech-to-Text Service**.

You can enable the **Speech-to-Text Service** as needed and select the supported language mode. The two modes have different prices. For details, see [Billing Rules](#).



6. Click **OK**.

Service Settings

After being created, the application will be displayed in the application list. Click **Set** on the right side of an application to go to the details page.

| Application Name | Application ID (AppID) | Creation Time | Application Status | Service Status | Operation |
|------------------|------------------------|---------------------|--------------------|----------------|--|
| GME | 1400355150 | 2020-04-21 17:18:56 | Normal | | Set Usage Statistics |


Total items: 1

20 / page 1 / 1 page

Setting application

Click **Modify** in the **Application Info** section to modify the corresponding information.

Application Info Modify

| | |
|-----------------------------------|---|
| Application Name | GME |
| Application Status | Normal |
| Acceleration for Chinese mainland | Enable |
| Project | DEFAULT PROJECT |
| Tag |  |
| Creation Time | 2020-04-21 17:18:56 |
| Last Modified | 2020-04-21 17:18:56 |

Enabling/Disabling service

Click **Modify** in the **Real-time Voice Service** section to enable or disable the service.

Real-time Voice Service Modify

| | |
|----------------|----------|
| Service Status | Enabled |
| Sound Quality | Standard |

Key Parameters

In **Authentication Info**, you can obtain the AppID and permission key required for the SDK voice services.

Manage API Key

⚠ Safety Warning

- API key is an important certificate to request for creating Tencent Cloud API. With the API, you can operate all your Tencent cloud resources. For your property and service security, please keep the key safe and change it regularly.
- Please do not upload or share your key information by any means (such as GitHub). Once leaked to external channels, it may cause significant loss of your cloud assets
- There are security risks if you call the TencentCloud API using a low-version TLS protocol. TLS 1.2 or later is recommended.

📘 Usage Notes

- The API Keys is used to generate a signature when you call the Tencent Cloud API. [Check the algorithm for generating a signature.](#)
- Your API key represents your account identity and permissions, and acts as your login password. Do not disclose it to others.
- Last Access Time is when this API key last accessed the TencentCloud API to use the Last Accessed Service in last 30 days. If no access records exist or the most recent records are from more than 30 days ago, no records will be displayed. The records are from CloudAudit. [See the list of services and APIs supported by CloudAudit.](#)

Create Key

| APPID | Key | Creation Date | Last Access Time | Last Accessed Service | Status | Action |
|-----------|---|---------------------|------------------|-----------------------|---------|---------|
| [blurred] | SecretId: [blurred] SecretKey: *****Show | 2019-06-27 17:25:38 | - | - | Enabled | Disable |

Note :

- The permission key here will be used as a parameter when accessing the SDK.
- Change of the key on this page takes effect within 15 minutes to 1 hour. It is not recommended to change it frequently.
- Only the account that creates the game, root account, and global collaborators can **reset the key**.
- For more information about authentication, see [Authentication Key](#).

To use the demo, you need to replace the AppID of the Tencent Cloud test account in the corresponding interface of the demo with the AppID you obtained in the console, and change the permission key of voice chat in the GetAuthBuffer function of AVChatViewController.

API Documentation

Depending on the platform or engine you are using, you can access the SDK by referring to the following documentations:

Documents for Unity:

[Project Configuration](#)

[API Documentation](#)

Documents for Unreal Engine:

[Project Configuration](#)

[API Documentation](#)

Documents for Cocos2D:

[Project Configuration](#)

[API Documentation](#)

Documents for Windows:

[Project Configuration](#)

[API Documentation](#)

Documents for iOS:

[Project Configuration](#)

[API Documentation](#)

Documents for Android:

[Project Configuration](#)

[API Documentation](#)

Documents for Mac:

[Project Configuration](#)

[API Documentation](#)

Documents for HTML5:

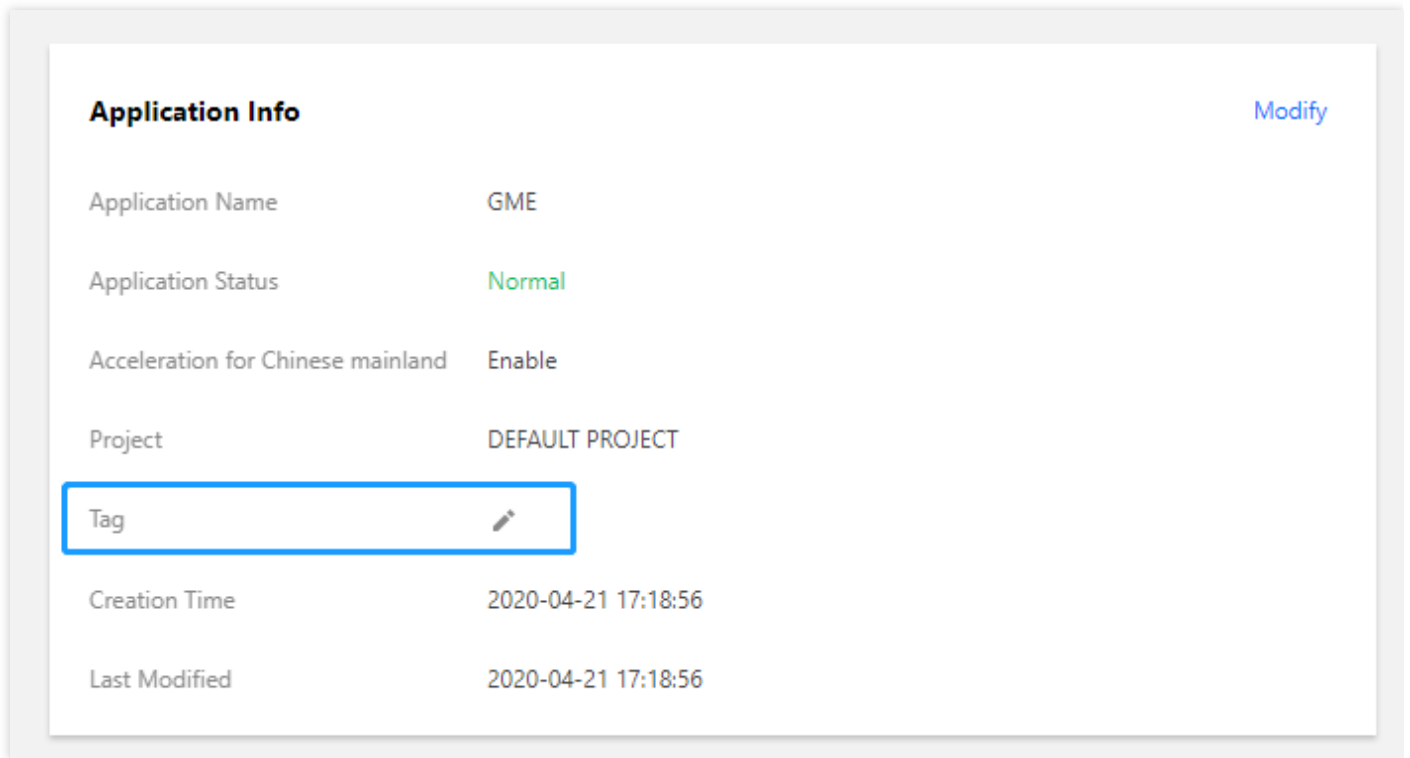
[Project Configuration](#)

[API Documentation](#)

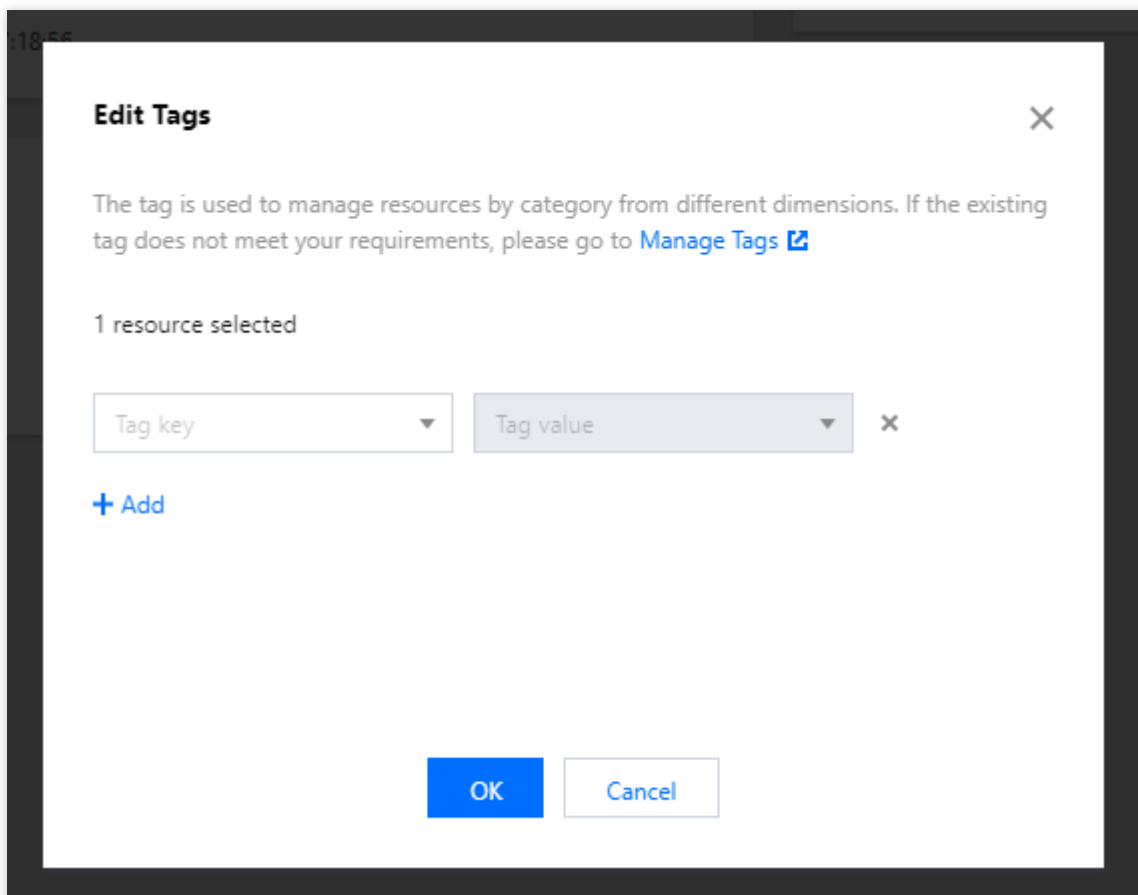
Tag Management

If you need to create tags when [creating an application](#), you can follow the directions below:

1. In the **Application Info** page when creating an application, you can click [Tag Management](#) to go to the **Tag List** page.



2. Click **Create** and enter the related information in the **Add tag** window.





3. Click **OK**.

Operation Guide

Last updated : 2020-05-06 16:31:14

This document describes how to view the console usage statistics in the GME SDK, which divide into three services: voice chat, voice messaging and speech-to-text, and phrase filtering.

Log in to the [GME Console](#) and click **Usage Statistics** in the application you want to query.

| Application Status | Service Status | Operation |
|--------------------|---|--|
| Normal |   | Settings Usage Statistics |

Usage Statistics of Voice Chat

The usage statistics display the DAU data of a voice chat application.

- By default, the total DAU data for the last 7 days is displayed, and the usage data for different time periods and regions can be viewed.
- **Voice Duration**: it displays yesterday's voice duration information of the application.
- **Voice DAU**: it displays yesterday's voice DAU information of the application.
- **Voice PCU**: it displays yesterday's voice PCU information of the application.

[← Usage Statistics \(GME\)](#)

Real-time Voice Service Voice-to-Text Service

Voice Duration

0 Minute

Voice DAU

0 People

Voice PCU

0 People

Last 7 days

Last 15 days

Last 30 days

2020-04-14 ~ 2020-04-20



Voice Duration

Voice DAU

Voice PCU

Total

Mainland

International

Voice Duration

Voice Duration / Minute

0

04-14

04-15

04-16

04-17

04-18

04-19

04-20

Voice Messaging and Speech-to-Text

The usage statistics display the DAU data of an offline voice application.

- **Voice Message DAU:** it displays yesterday's total number of voice DAUs of the application.
- **Number of Conversions:** it displays yesterday's number of speech-to-text conversions of the application.
- By default, the total DAU data for the last 7 days is displayed, and the usage data for different time periods can be viewed.

[← Usage Statistics \(GME\)](#)Real-time Voice Service Voice-to-Text Service

Voice Message DAUs

0 Users

Number of Conversion Texts

0 Times

Last 7 days

Last 15 days

Last 30 days

2020-04-14 ~ 2020-04-20



Voice Message DAUs

Number of Conversion Texts

Voice Message DAUs

Voice Message DAUs / Users

0

04-14

04-15

04-16

04-17

04-18

04-19

04-20

Phrase Filtering Service

The usage statistics display the duration of audio processed for phrase filtering.

- By default, the audio duration for the last 7 days is displayed, and the usage data for different time periods can be viewed.

Native SDK Quick Access

Last updated : 2021-06-22 15:05:07

This document provides a detailed description that makes it easy for Native project developers to debug and integrate the APIs for Game Multimedia Engine (GME).

This document only provides the major APIs to help you get started with GME. You can refer to the Demo to debug and integrate the APIs.

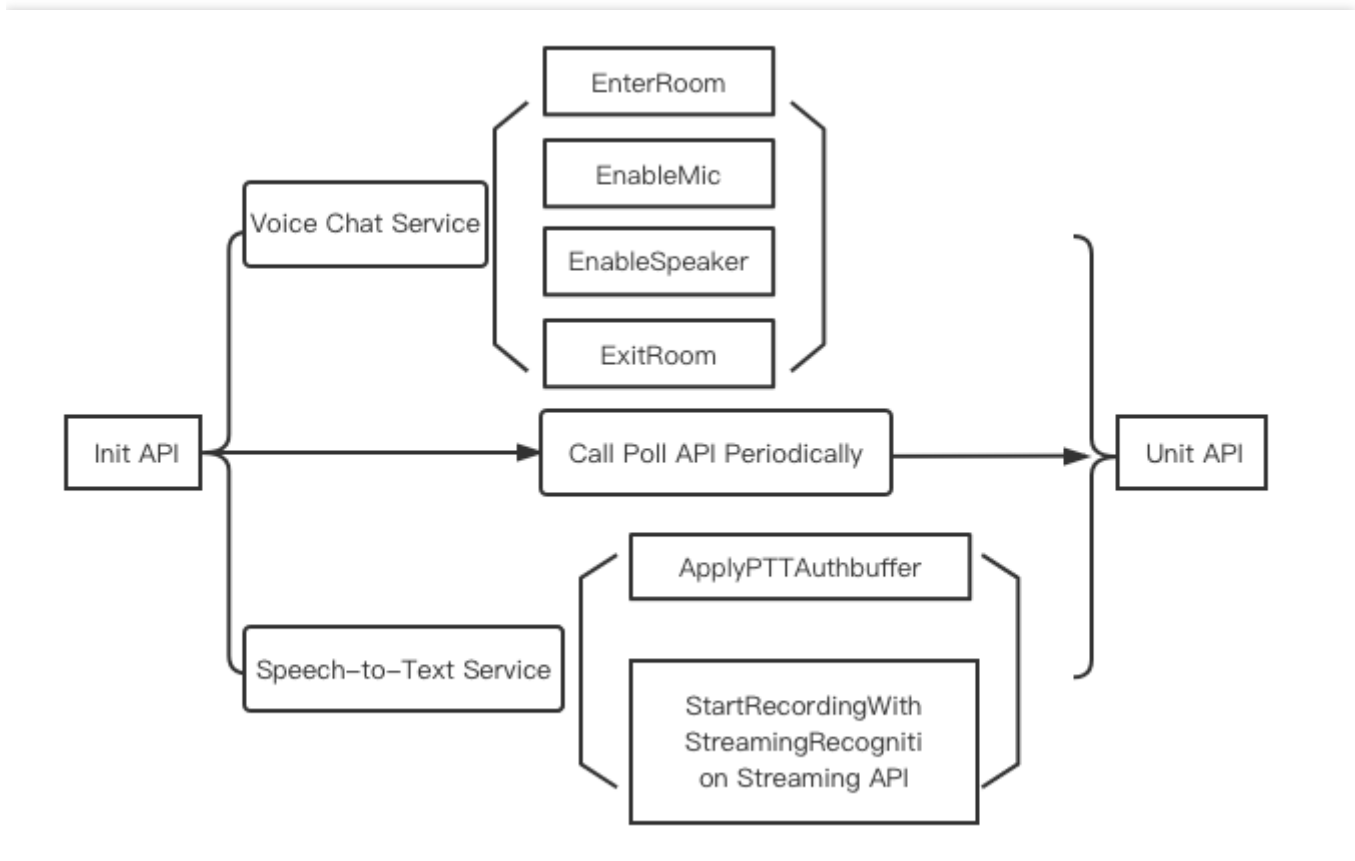
Key Considerations for Using GME

GME provides two services: voice chat service and voice message and speech-to-text service, both of which rely on key APIs such as Init and Poll.

Note on Init API

If you need to use voice chat and voice message services at the same time, **you only need to call Init API once.**

API call flowchart



Directions

Key APIs

Initializing GMEAPI: *Init*

Calling Poll periodically to trigger event callbacksAPI: *Poll*

Voice Chat

- 1 Entering a voice chat roomAPI: *EnterRoom*
- 2 Enabling or disabling the microphoneAPI: *EnableMic*
- 3 Enabling or disabling the speakerAPI: *EnableSpeaker*
- 4 Exiting a voice roomAPI: *ExitRoom*

Voice Message

- 1 Initializing authenticationAPI: *ApplyPTTAuthbuffer*
- 2 Starting streaming speech recognitionAPI: *StartRecordingWithStreamingRecognition*
- 3 Stop recordingAPI: *StopRecording*

Uninitializing GMEAPI: *UnInit*

Key API Access

Downloading demo

In the SDK download guide page, download the client's [Demo project codesDownload](#).

2. Importing the header file

- [Java](#)
- [Object-C](#)
- [C++](#)

```
import com.tencent.TMG.ITMGContext;  
import com.tencent.av.sig.AuthBuffer;  
import com.tencent.bugly.crashreport.CrashReport;
```

3. Getting singleton

To use the voice feature, get the `ITMGContext` object first.

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)

```
public static ITMGContext GetInstance(Context context)
```

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//MainActivity.java  
import com.tencent.TMG.ITMGContext;  
ITMGContext tmgContext = ITMGContext.GetInstance(this);
```

4. Setting callback

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages before room entry.

Function prototype and sample code

Register the callback function to the SDK for receiving callback messages before room entry.

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//ITMGContext
public abstract int SetTMGDelegate(ITMGDelegate delegate);

//MainActivity.java
tmgContext.SetTMGDelegate(TMGCallbackDispatcher.getInstance());
```

Callback examples

Override this callback function in the constructor to process the parameters of the callback.

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//MainActivity.java
tmgContext.SetTMGDelegate(TMGCallbackDispatcher.getInstance());

//RealTimeVoiceActivity.java
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (type == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
    {
        // Processing callbacks
    }
}

// Refer to TMGCallbackDispatcher.java, TMGCallbackHelper.java, and TMGDispatcherBase.java
```

| Parameter | Type | Description |
|-----------|----------------------------------|-------------------------------------|
| type | ITMGContext.ITMG_MAIN_EVENT_TYPE | Event type in the callback response |
| data | Intent message type | Callback message, i.e., event data |

5. Initializing SDK

- This API is used to initialize the GME service. It is recommended to call it when initializing the application.
- **For more information on how to get the `sdkAppId` parameter, please see [Voice Service Activation Guide](#).**
- **The openID uniquely identifies a user with the rules stipulated by the application developer and unique in the application (currently, only INT64 is supported).**

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)

```
public abstract int Init(String sdkAppId, String openId);
```

| Parameter | Type | Description |
|-----------|--------|---|
| sdkAppId | String | <code>AppId</code> provided by the GME service from the Tencent Cloud Console |
| OpenId | String | <code>OpenId</code> can only be in Int64 type, which is passed after being converted to a string. |

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//MainActivity.java
int nRet = tmgContext.Init(appId, openId);
if (nRet == AV_OK )
{
    GMEAuthBufferHelper.getInstance().setGEMParams(appId, key, openId);
    // Step 4/11: Poll to trigger callback
    //https://cloud.tencent.com/document/product/607/15210#.E8.A7.A6.E5.8F.91.E4.BA.8B.E4.BB.B6.E5.9B.9E.E8.B0.83
    EnginePollHelper.createEnginePollHelper();
    showToast("Init success");
}else if (nRet == AV_ERR_HAS_IN_THE_STATE) // SDK has been initialized. This operation is successful.
{
```

```
showToast("Init success");
}else
{
showToast("Init error errorCode:" + nRet);
}
```

6. Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. The `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.

You can refer to the `EnginePollHelper.java` file in the demo.

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//MainActivity.java
[EnginePollHelper createEnginePollHelper];

//EnginePollHelper.java
private Handler mHandler = new Handler();
private Runnable mRunnable = new Runnable() {
@Override
public void run() {
if (s_pollEnabled) {
if (ITMGContext.GetInstance(null) != null)
ITMGContext.GetInstance(null).Poll();
}
mhandler.postDelayed(mRunnable, 33);
}
};
// For the code of calling Poll periodically, please see EnginePollHelper.java.
```

7. Authentication

Generate `AuthBuffer` for encryption and authentication of relevant features.

To get authentication for voice message and speech-to-text, the room ID parameter must be set to `null`.

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)

```
AuthBuffer public native byte[] genAuthBuffer(int sdkAppId, String roomId, String openId, String key)
```

| Parameter | Type | Description |
|-----------|--------|--|
| appId | int | AppId from the Tencent Cloud console. |
| roomId | string | Room ID, which can contain up to 127 characters (For voice message, enter "null".) |
| openId | string | User ID, which is the same as <code>openId</code> during initialization. |
| key | string | Permission key from the Tencent Cloud console . |

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//GMEAuthBufferHelper.java
import com.tencent.av.sig.AuthBuffer; // Header file
public byte[] createAuthBuffer(String roomId)
{
    byte[] authBuffer;
    // Generate AuthBuffer for encryption and authentication of relevant features. For release in the
    // production environment,
    // please use the backend deployment key as detailed in https://intl.cloud.tencent.com/document/p
    // roduct/607/12218
    if (TextUtils.isEmpty(roomId))
    {
        authBuffer = AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(mAppId), "0", mOpenId, mKe
        y);
    }else
    {
        authBuffer = AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(mAppId), roomId, mOpenId, mK
        ey);
    }
    return authBuffer;
}
```

Voice Chat Access

1. Entering a room

When a user enters a room with the generated authentication information, the

`ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` message will be received as a callback. Mic and speaker are not enabled by default after room entry. The returned value of `AV_OK` indicates a success.

For more information on how to choose a room audio type, please see [Sound Quality Selection](#).

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)

```
public abstract int EnterRoom(String roomId, int roomType, byte[] authBuffer);
```

| Parameter | Type | Description |
|------------|--------|---|
| roomId | String | Room ID, which can contain up to 127 characters |
| roomType | int | Room audio type |
| authBuffer | byte[] | Authentication code |

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//RealTimeVoiceActivity.java  
byte[] authBuffer = GMEAuthBufferHelper.getInstance().createAuthBuffer(roomId);  
ITMGContext.GetInstance(this).EnterRoom(roomId, roomType, authBuffer);
```

Callback for room entry

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function for callback and processing. A successful callback means that the

room entry is successful, and the billing starts. It will be free of charge if the total call duration of the day is less than 700 minutes.

Billing references

[Purchase Guide](#)

[Billing FAQs](#)

[Will the billing continue if the client is disconnected when using the voice chat?](#)

• Sample code

Sample code for processing the callback, including room entry and network disconnection events.

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//RealTimeVoiceActivity.java
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (type == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
    {
        // Step 6/11 : Perform the enter room event
        int nErrCode = TMGCallbackHelper.ParseIntentParams2(data).nErrCode;
        String strMsg = TMGCallbackHelper.ParseIntentParams2(data).strErrMsg;
        if (nErrCode == AV_OK)
        {
            appendLog2MonitorView("EnterRomm success");
        }else
        {
            appendLog2MonitorView(String.format(Locale.getDefault(), "EnterRomm errCode:%d errMsg:%s", nErrCode, strMsg));
        }
    }
}
```

• Error code

| Error Code Value | Cause and Suggested Solution |
|------------------|---|
| 7006 | Authentication failed. Possible causes: <ul style="list-style-type: none"> ◦ The `AppID` does not exist or is incorrect. ◦ An error occurred while authenticating the `authbuff`. ◦ Authentication expired. ◦ The `openId` does not meet the specification. |
| 7007 | Already in another room. |

| Error Code Value | Cause and Suggested Solution |
|------------------|--|
| 1001 | The user was already in the process of entering a room but repeated this operation. It is recommended not to call the room entering API until the room entry callback is returned. |
| 1003 | The user was already in the room and called the room entering API again. |
| 1101 | Make sure that the SDK is initialized, `openId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally. |

2. Enabling or disabling the microphone

This API is used to enable/disable the mic. Mic and speaker are not enabled by default after room entry.

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//RealTimeVoiceActivity.java  
ITMGContext.GetInstance(this).GetAudioCtrl().EnableMic(true);
```

3. Enabling or disabling the speaker

This API is used to enable/disable the speaker.

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//RealTimeVoiceActivity.java  
ITMGContext.GetInstance(this).GetAudioCtrl().EnableSpeaker(true);
```

4. Exiting the room

This API is called to exit the current room. It is an async API. The returned value of `AV_OK` indicates a successful async delivery.

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//RealTimeVoiceActivity.java  
ITMGContext.GetInstance(this).ExitRoom();
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM`. The sample code is shown below:

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//RealTimeVoiceActivity.java  
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {  
  if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_EXIT_ROOM == type)  
  {  
    // Receive the event of successful room exit  
  }  
}
```

Voice Message Access

1. Initializing authentication

Call authentication initialization after initializing the SDK. For more information on how to get the `authBuffer`, please see `genAuthBuffer` (the voice chat authentication information API).

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)


```
public abstract int ApplyPTTAuthbuffer(byte[] authBuffer);
```

| Parameter | Type | Description |
|------------|--------|----------------|
| authBuffer | String | Authentication |

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//VoiceMessageRecognitionActivity.java
byte[] authBuffer = GMEAuthBufferHelper.getInstance().createAuthBuffer("");
ITMGContext.GetInstance(this).GetPTT().ApplyPTTAuthbuffer(authBuffer);
```

2. Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call `StopRecording`** . The callback will be returned after the recording is stopped.

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)

```
public abstract int StartRecordingWithStreamingRecognition (String filePath);
public abstract int StartRecordingWithStreamingRecognition (String filePath,String language,String translatelanguage);
public abstract int StopRecording();
```

| Parameter | Type | Description |
|----------------|--------|---|
| filePath | String | Path of stored audio file |
| speechLanguage | String | The language in which the audio file is to be converted to text. For parameters, please see Language Parameter Reference List |

| Parameter | Type | Description |
|-------------------|--------|---|
| translateLanguage | String | The language into which the audio file will be translated. For parameters, please see Language Parameter Reference List (This parameter is currently unavailable. Enter the same value as that of <code>speechLanguage</code>) |

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//VoiceMessageRecognitionActivity.java
ITMGContext.GetInstance(this).GetPTT().StartRecordingWithStreamingRecognition(recordFilePath, "cmn-Hans-CN");
```

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen for callback messages in the callback function `onEvent` . The event message is

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` , namely returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

The event message will be identified in the `OnEvent` function based on the actual needs. The passed parameters include the following four messages.

| Message Name | Description |
|--------------|---|
| result | A return code for judging whether the streaming speech recognition is successful. |
| text | Text converted from speech |
| file_path | Local path of stored recording file |
| file_id | Backend URL address of recording file, which will be retained for 90 days |

• Sample code

- [Java](#)
- [Object-C](#)

- o C++

```
//VoiceMessageRecognitionActivity.java
import static com.tencent.TMG.ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE;
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (type == ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE)
    {
        // Step 1.3/3 handle ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE event
        mIsRecording = false;
        if (nErrCode == 0)
        {
            String recordFilePath = data.getStringExtra("file_path");
            mRecFilePathView.setText(recordFilePath);

            String recordFileUrl = data.getStringExtra("file_id");
            mRecFileUrlView.setText(recordFileUrl);
        }
        else
        {
            appendLog2MonitorView("Record and recognition fail errCode:" + nErrCode);
        }
    }
}
```

- Error code

| Error Code | Description | Suggested Solution |
|------------|--|---|
| 32775 | Streaming speech-to-text conversion failed, but recording succeeded. | Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion. |
| 32777 | Streaming speech-to-text converting failed, but recording and upload succeeded | The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion. |
| 32786 | Streaming speech-to-text conversion failed. | During streaming recording, wait for the execution result of the streaming recording API to return. |

3. Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

Function prototype

- [Java](#)
- [Object-C](#)
- [C++](#)

```
public abstract int StopRecording();
```

Sample code

- [Java](#)
- [Object-C](#)
- [C++](#)

```
//VoiceMessageRecognitionActivity.java  
ITMGContext.GetInstance(this).GetPTT().StopRecording();
```