

Agora Android SDK Reference

Contents

Required Libraries	2
Required Permissions	2
Class Reference	2
AgoraAudio	2
Methods	3
Initialization	3
Join Channel	3
Leave Channel	3
Mute	4
Get Voip Stats	4
Set Speaker	4
IAudioEventHandler Interface	4
Methods	4
OnError	5
onJoinSuccess	5
onLeaveChannel	5
onLoadAudioEngineSuccess	6
onLogEvent	6
onQuality	6
onUpdateSessionStats	7
onUserOffline	7
Know Issues	7

Required Libraries

In the libs folder you need these folders and files from the libs folder in the SDK:

- Armeabi
- Armeabi-v7a
- X86
- Agoravoice-sdk.jar

Required Permissions

In AndroidManifest.xml you need:

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

```
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
```

Class Reference

AgoraAudio

Package: io.agoravoice.voiceengine

All of these methods can be called on the main UI.

Name	Description
Initialization	Constructor. Creates AgoraAudio instance.
Join Channel	Join a channel, meaning join a call.
Leave Channel	Leave a channel, meaning leave a call.
Mute	Mute the client itself by calling mute(true) to mute the microphone and mute(false) to turn it back on.
getVoipStats	Get call duration, tx/rx bytes information.
setSpeaker	Switch between speaker and earpiece.

Methods

Initialization

public AgoraAudio(Content context, IAudioEventHandler handler);

Parameters

Name	Description
Content	Android activity context.
IAudioEventHandler	IAudioEventHandler is an interface whose methods the programmer must implement. It has callback methods to provide status as to whether there was an error joining the call as well as call quality and other information.

Return Value

AgoraAudio

Join Channel

**public void joinChannel(final String vendorKey, final String channelName,
final String optionalInfo, final int optionalUid)**

Name	Description
vendorKey	Account credentials issued by Agora Voice to app developer, i.e., a user license.
channelName	Channel name. Any descriptive name like “game1” or “call2”.
optionalInfo	Whatever the additional information the programmer wants to add.
optionalUid	User id. If you do not set one the SDK supplies one.

Return Value

none

Leave Channel

Leave channel or hang up.

public void leaveChannel();

Return Value

None

Mute

Turn off microphone

public void mute(final boolean b);

Name	Description
boolean	True to mute the call. False to unmute.

Return Value

None

Get Voip Stats

getVoipStats()

Return Value

Name	Description
SessionStats	Int avgDelay: average delay in ms Int avgLost: average packet loss rate in percent Int rtDelay: realtime delay in ms Int rtLost: realtime packet loss rate in percent Int totalBytes: total tx/rx bytes Int totalDuration: total call duration

Set Speaker

setSpeaker(boolean b)

Name	Description
boolean	False means output audio to earphone. True means output audio to speaker.

IAudioEventHandler Interface

Package: io.agoravoice.voiceengine

This event handler is called when the app joins the channel or leaves the channel. These are callback methods that provide status information.

Methods

Here are the methods you must implement.

OnError

public void onError(int errorCode)

Parameters

Name	Description
errorCode	See error codes below.

Error codes:

- EVT_LOAD_AUDIO_ENGINE_ERROR: failed to initialize audio engine.
- EVENT_START_CALL_ERROR: failed to start audio engine. Typically this is caused because the audio device is in use by another app.
- EVT_JOIN_GET_AUDIO_ADDR_TIMEOUT: voice server list timeout.
- EVT_JOIN_GET_AUDIO_ADDR_FAILED: error code received when requesting voice server list.
- EVT_JOIN_GET_AUDIO_ADDR_ZERO_ADDR: Voice Center Server (acts as a gateway, like DNS) responded that there is no voice server available.
- EVT_JOIN_LOGIN_MEDIA_TIMEOUT_ALL: login to voice server timeout.
- EVT_JOIN_LOGIN_REGET_AUDIO_ADDR: the voice central server (acts as a gateway, like DNS) tried all available voice servers but none is able to handle this call.
- EVT_NETWORK_DOWN: network is down.

Return Value

None

onJoinSuccess

public void onJoinSuccess(int sid, int uid)

Parameters

Name	Description
sid	Session id. This is an integer representation of the channel.
uid	User id. Create automatically if not supplied by the user.

Return Value

None

onLeaveChannel

public void onLeaveChannel(SessionStats sessionStats)

Parameters

Name	Description
sessionStats	Int avgDelay: average delay in ms Int avgLost: average packet loss rate in percent Int rtDelay: realtime delay in ms Int rtLost: realtime packet loss rate in percent Int totalBytes: total tx/rx bytes Int totalDuration: total call duration

[onLoadAudioEngineSuccess](#)

public void onLoadAudioEngineSuccess()

No parameters. No return value. Called when audio engine is loaded successfully.

[onLogEvent](#)

public void onLogEvent(byte[] evt)

Parameters

Name	Description
evt	The evt argument is actually log message which can be converted into String by new String(evt, "ISO-8859-1").

Return Value

None

[onQuality](#)

public void onQuality(int uid, short delay, short jitter, short lost, short lost2)

This is invoked every 2 seconds to provide call quality for every user on the call.

Parameters

Name	Description
uid	User id of this user (There can be multiple people on one call.).
delay	Delay in microseconds.
lost	Lost packet ratio as a percentage, 0 to 100.
lost2	Number of times that 2 consecutive packets were lost.
jitter	Means variation in the delay of received packets due to network congestion or queuing issues.

Return Value

None

[onUpdateSessionStats](#)

public void onUpdateSessionStats(SessionStats sessionStats)

Parameters

Name	Description
sessionStats	Int avgDelay: average delay in ms Int avgLost: average packet loss rate in percent Int rtDelay: realtime delay in ms Int rtLost: realtime packet loss rate in percent Int totalBytes: total tx/rx bytes Int totalDuration: total call duration

Return Value

None

[onUserOffline](#)

public void onUserOffline(int uid)

This callback indicates the client has not received audio packets for a period (15 seconds by default).

Parameters

Name	Description
uid	User id of this user.

Return Value

None

[Know Issues](#)

Android's Opensles library issue Agora audio engine relies on Android built-in Opensles library. There's a known issue within the Opensles that might hang the application while destroying Opensles objects. To work around this issue, it is better to acquire audio focus before joining a channel, and abandoning it after leaving channel.