

「音声小さい」事象の検証方法 (Android編)

株式会社バイキューブ

2021年10月



Androidアプリにて「音声小さい」というお問い合わせを良くいただきます。

原因は様々ありますが、具体的な調査が必要な場合は、agoraにて通話情報とSDKログを元に解析調査を行います。

本ドキュメントでは、Androidアプリにて「音声小さい」事象が発生した場合に、ログ解析に必要な情報と、それを取得するための再現検証の実施方法（基本編）、参考情報について記載しています。

1. はじめに
2. 「音声小さい」事象のあるえる原因 (Androidの場合)
3. agoraのログ解析調査に必要な情報
4. ステップ1: 事象の情報整理(チェックリスト)
5. ステップ2: 再現検証の準備
6. ステップ3: 再現検証
7. ステップ4: agoraへログ解析依頼
8. (番外編)音量を自分で確認するには(APIなど)
9. (番外編)APIで利用して音量を改善させる方法

「音声小さい」事象のあるえる原因 (Androidの場合)

「音声小さい」事象について、ありえる原因は主に以下になります。

[ありえる原因]

- デバイス側(又はヘッドセット)の音量が小さかった。
- ネットワークの通信状態が良くなかった
- イヤホンの問題(故障、もしくはデバイスとの相性が悪い)
- Androidの機種依存(SDKとの相性が悪い)
- SDK側の不具合

上記は一例になりますが、
具体的な原因の調査が必要な場合は、agoraにて通話情報とSDKログを元に解析調査を行います。

Agoraのログ解析時に必要な情報は以下になります。

【必要情報】

APPID	アプリを識別するID	[参考] APP ID、Channel ID、User ID(uid)とは何ですか？
Channel名	アプリ内で開催する通話会議を識別するID	
UID	Channel ID内に入室した参加者を識別するID	※送信側、受信側それぞれ分かるように記載
SDKバージョン	ご利用になっているSDKバージョン	
発生日時	事象が発生した日時(yyyy/MM/dd HH:mm:ss)	
発生機種/環境	発生デバイス(モデル名やOSバージョンなど) 利用したネットワーク環境Wifiや4Gなど)	※送信側、受信側両方を記載
症状	発生した症状と頻度 (「ステップ4: agoraへログ解析依頼」の例を参照)	※送信側、受信側両方について詳しく記載
ログファイル	SDKログ(【SDKログの設定】を参照)	※送信側、受信側両方を記載

[参考] [ログ解析を依頼したいのですが、どのような情報が必要でしょうか？](#)

→上記の情報が揃っているかどうか確認してください。「**ステップ1: 事象の情報整理**」へ

ステップ1: 事象の情報整理 (チェックリスト)

現時点でログ解析を依頼するための情報が十分かどうか確認します。

【チェックリスト】

✓ **発生日時 (いつ)**

※発生日時は日時で必要

✓ **発生時の利用環境 (どこで)**

- ・デバイス、イヤホン、ネットワーク環境
- ・自拠点(送信側)、他拠点(受信側)の両方の情報が揃っているか

✓ **症状(だれが、どのように)**

- ・発生時の状態 (全く聞こえない、聞こえるが小さい、たまに小さくなる、その他)
- ・自拠点(送信側)、他拠点(受信側)のどちらの状態が良いのか
- ・発生頻度(毎回発生するのか、1回だけなのか、数回に1回なのか?)

✓ **Agora 通話情報**

- ・APPID、Channel名、UID(送信側/受信側)、SDKバージョン

✓ **Agora SDKログ**

→ 全て揃っていない場合は再現検証をして必要情報を取得します。「ステップ 2: 再現検証の準備」へ

ステップ2:再現検証の準備

再現検証の実施時の注意点は以下になります。

【実施時の注意点】

- ・自拠点(送信側)、他拠点(受信側)の2台で実施してください。
- ・自拠点(送信側)、他拠点(受信側)の検証は、それぞれ別の部屋で実施してください。
- ・検証者1名で自拠点(送信側)、他拠点(受信側)を検証する場合は、イヤホンを装着するなどし、ハウリングなどの干渉が起きないようにしてください。
- ・利用環境は、なるべく発生時と同じ利用環境で実施してください。
 - ・ネットワーク (有線、無線、公衆回線(4G、5G)等)
 - ・デバイス機種 (iPhone、Android、Windows、macの各機種)
※デバイスはなるべく同機種で検証されることを推奨します。
 - ・OS (iOS、Android、Windows、macOS)
 - ・ヘッドセット (有線マイク付きイヤホン、Bluetoothヘッドセット、マイクスピーカーなど)
 - ・カメラ(デバイス付属、Webカメラ)
- ・AgoraのSDK環境は揃えてご実施ください。
 - ・Audio Profile
 - ・SDKバージョン

ステップ2:再現検証の準備

以下の環境を準備します。

【検証時に利用するもの】

以下の検証環境を 2セット(自拠点(送信側)、他拠点(受信側))用意します。

(注)再現検証のため、なるべく発生時と同じ環境をご用意ください。

1. デバイス

2. ヘッドセット(有線マイク付きイヤホン、Bluetoothヘッドセット、マイクスピーカー)

3. カメラ(デバイス付属、Webカメラ) ※必要であれば

4. ネットワーク環境(有線、無線、公衆回線(4G、5G)等)

5. agoraサンプル

1-1.SDKをダウンロードしてください。[SDKダウンロードリンク\(Android\)](#)

1.2.SDKに付属しているサンプル「Agora-Android-Tutorial-1to1」をビルドしてください。

※その他、ライブ配信の場合は「OpenLive-Android」、「API-Example」があります。

→【SDKログの設定】へ(次ページ)

ステップ2:再現検証の準備

【SDKログの設定】

Nativeアプリ(Android/iOS)の場合、ログ解析にはSDKログが必ず必要になります。
そのため、検証前に事前にSDKログの設定を設定しておきます。

[設定方法]

バージョンによって設定例が異なります。

- **v3.3.0以降の場合**

- **filePath**: 保存可能なパスを指定
- **level**: INFO で指定
- **fileSize**: 適宜可能なサイズを指定
※最大2048KB、第5世代まで

[コード例]

```
RtcEngineConfig.LogConfig logConfig = new RtcEngineConfig.LogConfig();
logConfig.level =
Constants.LogLevel.getValue(Constants.LogLevel.LOG_LEVEL_INFO);
String ts = new SimpleDateFormat("yyyyMMdd").format(new Date());
logConfig.filePath = "/sdcard/" + ts + ".log";
logConfig.fileSize = 2048;
```

```
RtcEngineConfig config = new RtcEngineConfig();
config.mAppId = getString(R.string.agora_app_id);
config.mEventHandler = iRtcEngineEventHandler;
config.mContext = context.getApplicationContext();
config.mAreaCode = getAreaCode();
config.mLogConfig = logConfig;
```

```
mRtcEngine = RtcEngine.create(config);
```

[参考] [SDKログの設定\(v3.3.0以降\)](#)

ステップ2:再現検証の準備

- v3.3.0 より前の場合

- `setLogFile`: 保存可能なパスを指定
- `setLogFilter`: INFO で指定
- `setLogfileSize`: 適宜可能なサイズを指定
※最大2048KB、第5世代まで

[参考] [SDKログの設定\(v3.3.0より前\)](#)

[コード例]

```
engine.setLogFilter(LOG_FILTER_DEBUG);  
  
String ts = new SimpleDateFormat("yyyyMMdd").format(new  
Date());  
String filepath = "/sdcard/" + ts + ".log";  
File file = new File(filepath);  
engine.setLogFile(filepath);
```

→準備ができれば検証を開始する「**ステップ3:再現検証**」へ
(※【録音ファイルの設定】が必要な場合は次ページへ)

ステップ2:再現検証の準備

【録音ファイルの設定】

★このステップは agora から指示があった場合のみ実施してください。

(基本的に Agora のログ解析時は、通話情報と SDK ログを使用します。さらに詳細に確認が必要な場合、agora より依頼があり録音ファイルを取得して送付します。)

[設定方法]

以下を設定してご実施ください。

1. プライベートAPIを設定してください。(場所はCreate後でJoinの前になります)

```
mRtcEngine.setParameters("{\"che.audio.start_debug_recording\":\"all\""}");
```

2. Androidに以下の許可を追加してください。

```
android.permission.WRITE_EXTERNAL_STORAGE.
```

3. 退室前に以下のAPIをセットしてください。

```
mRtcEngine.setParameters("{\"che.audio.stop_debug_recording\":\"true\""});
```

4. 検証終了後、出力されたファイルより録音された音声を確認することができます。















出力先:Android/Data/{ドメイン}/files以下
(wav、log、dataファイルなどが出力されます)

5. フォルダ以下一式をお送りください。

(注) サイズが非常に大きくなる場合がありますので、必要であればお持ちのファイル転送ツールなどでお送りください。

※Pixel4XLの出力例

Pixel 4 XL > 内部共有ストレージ > Android > data > io.agora.tutorials1v1vcall > files

名前	種類
 aec_coeff.dat	DAT ファイル
 aec_farin_16000hz_1ch.wav	WAV ファイル
 aec_linear_16000hz_1ch.wav	WAV ファイル
 aec_nearin_16000hz_1ch.wav	WAV ファイル
 aec_nearout_16000hz_1ch.wav	WAV ファイル
 af_agc_16000hz_1ch.wav	WAV ファイル
 af_ed_16000hz_1ch.wav	WAV ファイル
 af_ns_16000hz_1ch.wav	WAV ファイル
 agorasdk.log	テキストドキュメント
 agorasdk_1.log	テキストドキュメント
 bf_eff_16000hz_1ch.wav	WAV ファイル
 bf_eff_32000hz_1ch.wav	WAV ファイル
 far_in_48000hz_2ch.wav	WAV ファイル
 microphone_db.dat	DAT ファイル

ステップ3:再現検証

【再現検証①】Agoraサンプルで音声を検証する (必須)

1. 自拠点(送信側)、他拠点(受信側)がそれぞれ入室する。

- ・デバイスからマイク、カメラの許可ダイアログが表示される場合は許可しておく。
- ・デバイスの **音量を最大** にしておく(デバイスの音量バーを表示しupボタンをクリック)
- ・通話情報(開始日時、Channel名、UID(送信側、受信側))を記録しておく。

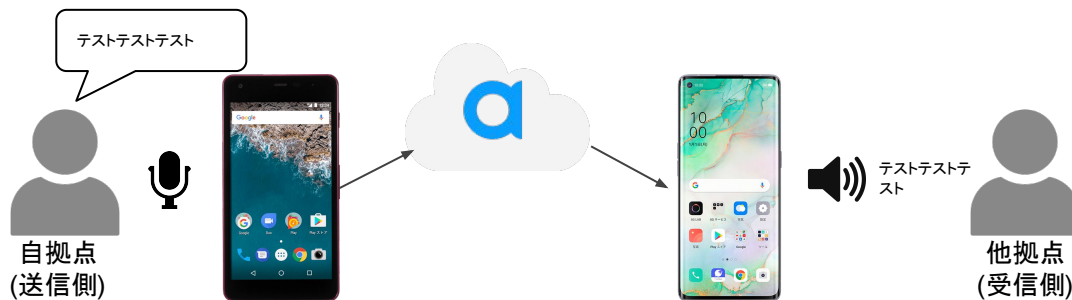
(お互いの入室確認後)

2. 自拠点(送信側)から音声を発する。

- ・”テストテストテスト”など連続して音声を発生する。(最低でも30秒以上)

3. 他拠点(受信側)で音声を聞く。

確認1: 自拠点(送信側)の音声状態(全く聞こえない、小さい、たまに聞こえるなど)、実施日時を記録する。



ステップ3:再現検証

(続き)

4. 他拠点(受信側)から音声を発する。

- ・”テストテストテスト”など連続して音声を発生する。(最低でも30秒以上)

5. 自拠点(送信側)で音声を聞く。

- ・**確認2: 自拠点(送信側)の音声状態(全く聞こえない、小さい、たまに聞こえるなど)、実施日時を記録する。**

6.1.から5.の手順を複数回繰り返す。(できれば3回以上は行ってください)

- ・**確認3: 発生回数(毎回、〇回に1回、〇回以降に毎回発生など)、実施日時を記録する。**

7. アプリを退室する。

8. SDKログの確認をする。

- ・setLog()APIで指定した場所にagorasdk.log というファイルがあるか確認する。
- ・ファイルの更新日時が検証日時あたりになっている。念のため中身も確認する。

→再現検証完了後、【ステップ4: agoraへログ解析依頼】へ
可能であれば【再現検証②】、【再現検証③】も実施 (次ページへ)

ステップ3:再現検証

【再現検証②】ヘッドセットを違うもので検証する

事前準備: 自拠点(送信側)、他拠点(受信側)それぞれのヘッドセットを違うものに変更します。

準備完了後、「【再現検証①】Agoraサンプルで音声を検証する」と同じ検証手順で検証する。

【再現検証③】別の通話アプリで検証する

事前準備: 別の通話アプリ(※)をインストールします。

※Lineアプリの通話、Skype、Zoom、ボイスチャットアプリなど

準備完了後、「【再現検証①】Agoraサンプルで音声を検証する」と同じ検証手順で検証する。

【再現検証 その他】APIやコンソールを利用し、音量を自分で確認する

※「(番外編)音量を自分で確認するには(APIなど)」をご参照ください。

(関連するAPIの一覧やコンソールの情報を載せています)

→再現検証完了後、【ステップ4: agoraへログ解析依頼】へ

ステップ4: agoraへログ解析依頼

【ログ解析依頼】

1. 検証結果とAgoraの通話情報をまとめる。

【例】

- APPID: xxxxxxxxxxxx
- Channel名: demo
- UID (送信側/受信側): 送信側:13579 / 受信側:24680
- 発生日時: 2021/10/01 12:30:45 (日本時間)
- 発生機種 (モデル名 / OSバージョンなど)
 - 送信側: Galaxy S9 SCV38 / Android 11 / 受信側: AQUOS R3 SH-04L / Android 9
- SDKバージョン: Video SDK v3.5.1 (旧バージョン: v3.2.1)
- 症状
 - SDKのバージョンアップをしたところ特定のデバイスで音が小さい事象が発生
 - 1. 自拠点/他拠点の音声検証検証①
 - 他拠点で自拠点の音声を聞いた場合、非常に小さい音で聞こえた
 - 自拠点で他拠点の音声を聞いた場合、大きく聞こえた
 - 頻度: 100%(3回/3回) 毎回発生
 - 2. ヘッドセットを変更して検証検証②
 - 他拠点の場合、非常に小さい音で聞こえた
 - 自拠点の場合、大きく聞こえた
 - 3. 別の通話アプリで検証検証③
 - 他拠点の場合、大きく聞こえた
 - 自拠点の場合、大きく聞こえた
- ログファイル (SDKログ)
 - 送信側: agorasdk.log, agorasdk_1.log, agorasdk_3.log, agorasdk_4.log, agorasdk_5.log
 - 受信側: agorasdk.log, agorasdk_2.log

ステップ4: agoraへログ解析依頼

(続き)

2. Zendeskサポートのチケットに必要情報を記載し依頼する

弊社にて必要情報を確認後、agoraにて解析を行います。
(情報に不足がある場合は必要情報をご連絡させていただきます。)

数日以内にチケットにて一次解析結果をご返信いたします。

(注)チケットが混みあう場合は順番に処理しますので、お時間をいただくことがありますので、ご了承ください。

【参考】(音量が小さい原因が)Androidのデバイス依存の場合、APIを利用して音量を標準値以上に設定すると改善する場合があります。

詳細は「(番外編)APIで利用して音量を改善させる方法」を参照してください。

APIやコンソールを利用し、音量を自分で確認する方法があります。

- API
 1. 音声ボリュームを確認する API
 - a. [enableAudioVolumeIndication](#)
 - b. [onAudioVolumeIndication](#)
 2. 音声ボリュームを調整する API
 - a. [adjustRecordingSignalVolume](#)
 - b. [adjustUserPlaybackSignalVolume](#)
 - c. [adjustPlaybackSignalVolume](#)
 3. 音声再生ルートを確認する API
 - a. [onAudioRouteChanged\(\)](#)

- コンソール
 1. Call Searchを利用して確認する
[参考] [Call Searchの見方や通話解析の方法を教えてください](#)

(番外編)APIで利用して音量を改善させる方法

Androidデバイスにおいて、音声入力または出力が小さい場合、以下の音声ボリュームを調整するAPIを利用し、標準(0～100)以上の音量値(200～400)を設定することにより、音量を改善させる方法があります。

音声ボリュームを調整するAPI

- [adjustRecordingSignalVolume](#)
- [adjustUserPlaybackSignalVolume](#)
- [adjustPlaybackSignalVolume](#)

音量値: 200～最大400

※デフォルトは0～100

(例) 音声入力の音量を最大(400)に設定する場合
`mRtcEngine.adjustRecordingSignalVolume(400);`