Agora Android How-to Guide

Contents

Demo App 1

Requirements: 1

Installation 1

Demonstration 2

Diagram of Program 6

IAudioEventHandler Interface 7

This manual explains how to use the Agora voice SDK by presenting a demo app and explaining how to use the SDK functions. The demo app is included when you unzip the SDK.

Demo App

The demo app shows the basics of how to join a call and leave a call.

Requirements:

- Android ADT (i.e. Eclipse with the Android Developers Toolkit)
- Two Android devices (You could use one, but you need two to make a phone call from one device to another.)

Installation

Import the project AgoraVoiceSDK-(version)-android\AgoraVoiceSDK-0.7.3-android\sample into the Eclipse workspace as an “Existing Android Applications.”

Demonstration

There are four basic functions to the Agora Voice SDK:

1. Initialization
2. Join Channel
3. Leave Channel
4. Mute Call

We explain those below with code samples. But first run the demo app to see what it does.

When you run the app the program presents three fields as shown below. When you press (-Join-) it joins a channel (call) by calling the joinChannel method on the AgoraVoice object:

public void joinChannel(final String vendorKey, final String channelName,

final String optionalInfo, final int optionalUid)

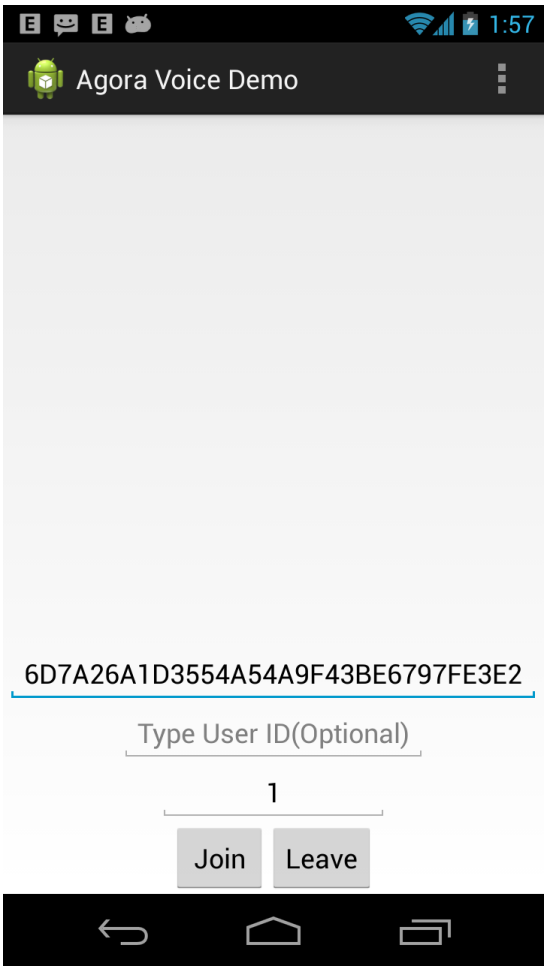
The arguments are:

vendorKey—this is the license key supplied to the Agora Voice customer to make calls over the Agora Voice cloud.

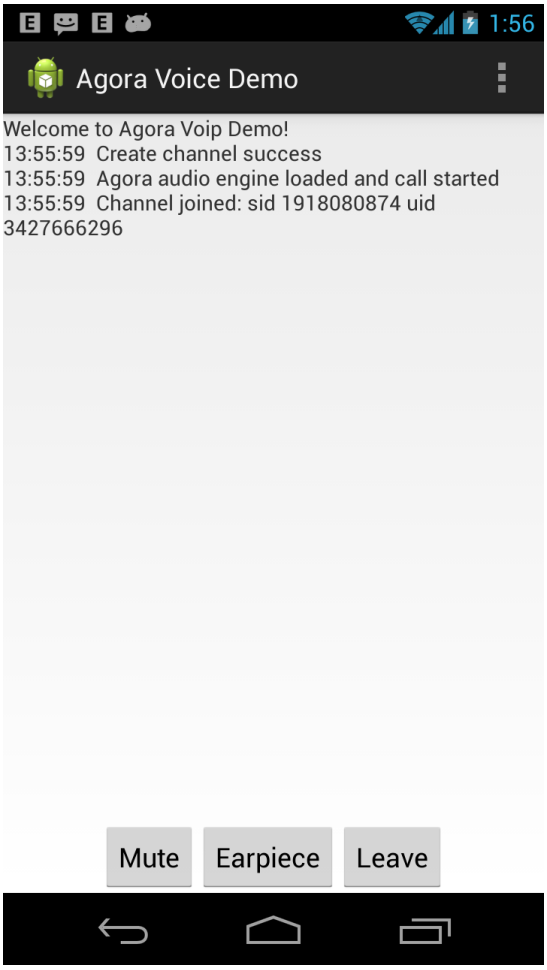
channelName—this is the name of the channel you can join. In the example below we use "1," but it could be something like "conference call" or "game XYZ."

optionalInfo—you can pass in any optional information here.

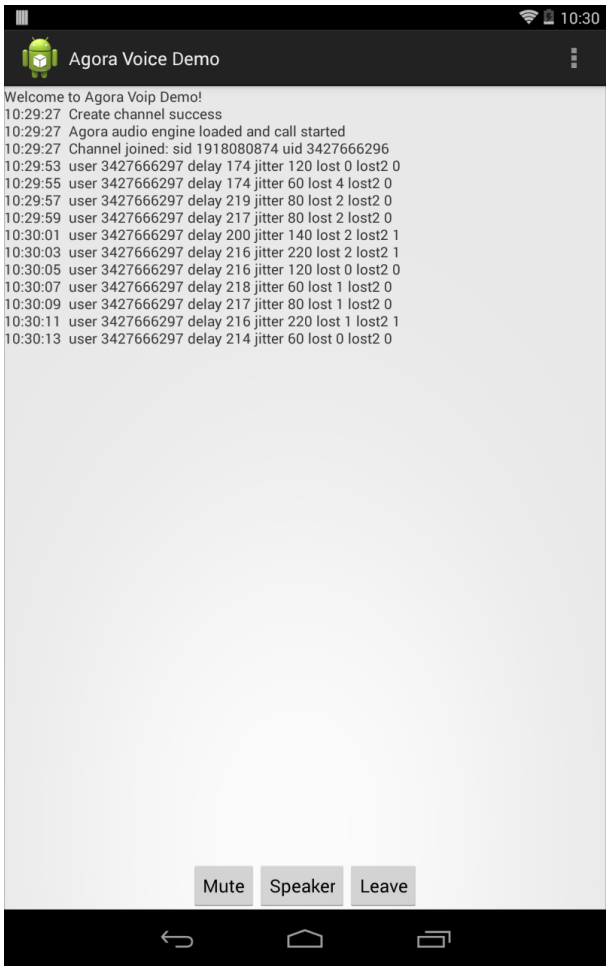
optionalUid—if you leave this blank, the AgoraVoice object will create an ID for you. This uniquely identifies the parties to the call. For this demo leave it blank.



When you press Join, the screen appears like this:



Then if you repeat the same procedure on another Android device (i.e., someone else joins the call) the screen looks like this and both parties can speak to each other.



The fields on the screen are:

Sid—this is an integer that represents the channel that you joined. The SDK generates this number.

uid—this is the userid created for the user.

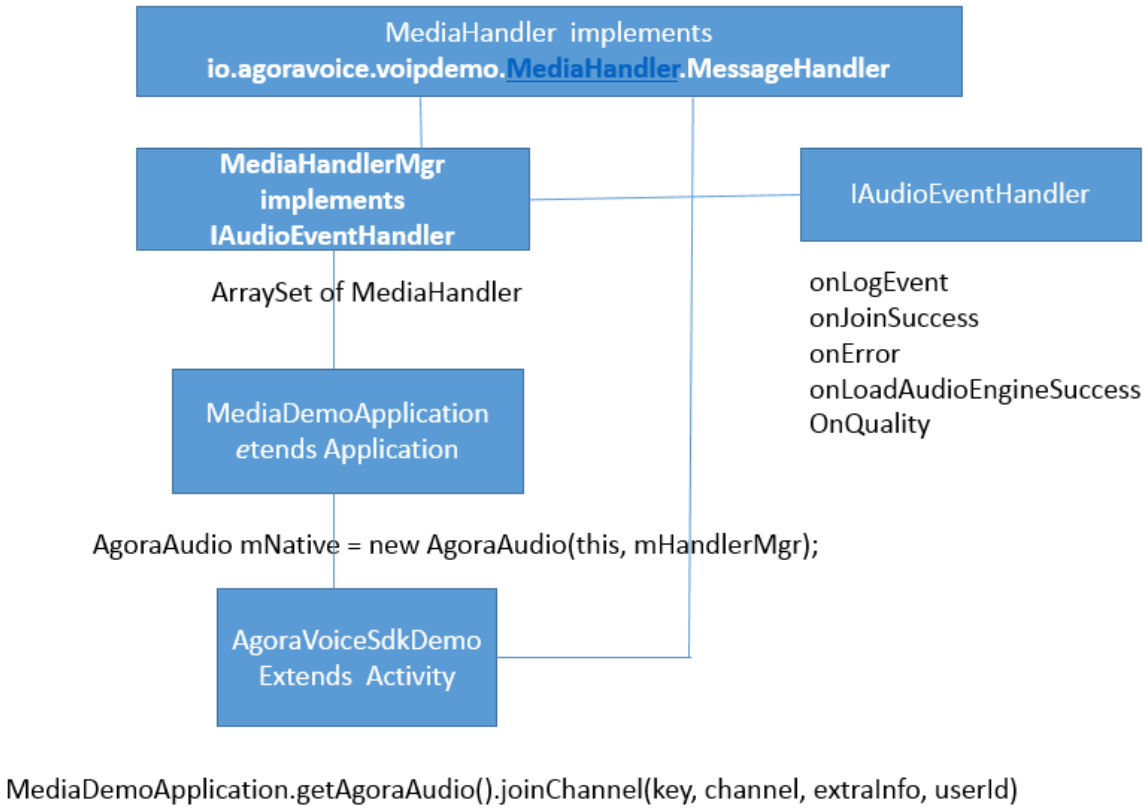
Delay—voice delay in ms.

Jitter—means variation in the delay of received packets due to network congestion or queuing issues.

Lost—lost packet ratio in percentage, from 0 to 99.

Diagram of Program

Here is a diagram of the classes in the sample app:



The classes shown above are:

- AgoraVoiceSdkDemo—this is the Android activity. It presents the buttons that the user presses to join the call.
- MediaDemoApplication—this is the Android Application. It initializes the AgoraAudio object.
- MediaHandlerMgr—this implements the IAudioEventHandler interface to handle callback methods when the call is joined to report status. It handles status message with an ArraySet of MediaHandlers.
- MediaHandler—this passes errors and other messages back to AgoraVoiceSdkDemo. It is an implementation of MessageHandler.

The four basic operations initialize, join, leave, and mute are called as below:

Initialize AgoraAudio object:

```
AgoraAudio  mNative = new AgoraAudio(this, mHandlerMgr);
```

Join Channel:

```
MediaDemoApplication.getAgoraAudio().joinChannel(key, channel, extraInfo, userId)
```

Leave Channel:

```
MediaDemoApplication.getAgoraAudio().leaveChannel()
```

Mute Call:

```
MediaDemoApplication..getAgoraAudio().mute(true);
```

IAudioEventHandler Interface

In the example, the MediaHandlerMgr implements IAudioEventHandler. It makes an ArraySet of MediaHandlers to process status messages.

The callback methods in IAudioEventHandler are called when the user joins a call to report on errors, success, and call quality. The developer implements the methods shown below:

MediaHandlerMgr implements IAudioEventHandler in the example. You implement the methods below to update the user interface with status.

```
@Override
public void onError(int arg0) {
 }
@Override
public void onJoinSuccess(int sid, int uid) {
}
@Override
public void onLeaveChannel(SessionStats stats) {
}
```