混音

1 功能简介

混音是指 SDK 从 App 获取一路音频数据，将 App 提供的音频数据与 SDK 采集的音频数据整合为一路音频数据；为了实现在通话或直播过程中，需要播放自定义的声音或者音乐文件并且让房间内的其他人也听到的需求。

混音功能常用于以下场景：

1. 直播过程中需要有掌声、口哨等音效，或者需要播放背景音乐。

2 使用步骤

2.1 初始化 SDK

如何初始化请查看文档：快速开始-初始化

2.2 启用混音功能

调用 `zego-api-audio-aux-oc.h` 下的 `enableAux` 来打开混音功能开关，才能在后续塞音频数据给 SDK。 **注意：此接口必须在初始化 SDK 之后调用，可在需要混音的任一时间调用。**

- 接口原型：

  ```
  - (bool)enableAux:(BOOL)enable
  ```

- 参数：

  `enable`：true-开启混音功能，false-关闭混音功能，默认为 false。

- 备注： 当开启混音后，SDK 会在 onAuxCallback:dataLen:sampleRate:channelCount: 中获取用于混音的音频数据，即需要开发者在此接口中塞音频数据给 SDK。

2.3 设置混音音量

在启用混音功能后，开发者可以根据需要调用 `zego-api-audio-aux-oc.h` 下的 `setAuxVolume` 调整混音后的音频的音量；或者调用 `muteAux` 设置混音为静音，静音后主播端将听不到混音声音，观众端还能听到混音声音。 **注意：设置混音音量的两个 API 可以在混音之前或者混完音之后调用，取决于用户需求。**

2.3.1 设置混音音量

> 1. `setAuxVolume` 的工作逻辑都是基于对引擎的输入输出数据进行处理，即对输入 SDK 的音频数据的音量大小进行设置，与进行混音的推流设备的系统音量没有关系。
> 2. 采集混音需要的音频数据时采用 general 模式，一般是媒体音量；采用 communication 模式，一般是通话音量；采用 auto 模式，连麦时会变成通话音量。`setAuxVolume` 是对采集的音频数据的音量进行设置。

- 接口原型：

  `- (void)setAuxVolume:(int)volume`

- 参数：

  `volume`：音量值范围 0 ~ 100，默认为 50。

- 备注：

  此接口设置的是本地播放的音量以及混音推出去的音量。

2.3.2 设置混音本地播放音量

- 接口原型：

  `- (void)setAuxPlayVolume:(int)volume`

- 参数：

  `volume`：音量值范围 0 ~ 100，默认为 50。

2.3.3 设置混音推流音量

- 接口原型：

  `- (void)setAuxPublishVolume:(int)volume`

- 参数：

  `volume`：音量值范围 0 ~ 100，默认为 50。

2.3.4 静音混音

当开启静音后，主播将听不到混音内容，观众端依然能听到混音声音。

- 接口原型：

  `- (bool)muteAux:(bool)bMute`

- 参数：

  `bMute`：true-静音，false-恢复音量。

2.4 将混音数据传给 SDK

在启用混音功能后，SDK 将通过 `onAuxCallback:dataLen:sampleRate:channelCount:` 获取待传递的音频数据来完成混音功能。App 在设置推流的回调监听并实现推流回调之后处理传递音频数据操作，推流请查看步骤 2.5 推流。

> 请注意：
>
> 1. 目前 SDK 仅支持位深为 16bit，16k、32k、44.1k、48k 采样率，单声道或者双声道的 PCM 音频数据格式，用户根据实际的 PCM 音频填写采样率及声道数。
> 2. 为确保混音效果，请不要在此 API 中执行耗时操作。

- 接口原型：

  `- (void)onAuxCallback:(void *)pData dataLen:(int *)pDataLen sampleRate:(int *)pSampleRate channelCount:(int *)pChannelCount`

- 参数：

`pData`：混音数据

`pDataLen`：pDataLen 既是输入参数也是输出参数； 作为输入参数，SDK 会提供好长度值，用户按照这个长度写入数据即可，数据充足的情况下，无需更改 pDataLen的值。作为输出参数，如果填写的数据不足 SDK 提供的长度值，则pDataLen = 0，或者最后的尾音不足 SDK 提供的长度值，可以用静音数据补齐。

`pSampleRate`：混音数据采样率，支持16k、32k、44.1k、48k。

`pChannelCount`：混音数据声道数，支持1、2。

2.5 推流

推流的调用说明请查看文档：快速开始-推流

3 API 参考列表

方法	描述
- enableAux:	开启混音功能
- setAuxVolume:	设置混音音量
- setAuxPublishVolume:	设置混音推流音量
- setAuxPlayVolume:	设置混音本地播放音量
- muteAux:	设置混音禁音
- setDelegate:	设置混音代理
- onAuxCallback:dataLen:sampleRate:channelCount:	混音数据输入回调

- 示例代码片段如下： 示例代码中pcm文件采样率为:44100 声道数为: 2

```
ZegoLiveViewController.m

- (void)onAuxCallback:(void *)pData dataLen:(int *)pDataLen sampleRate:(int *)pSampleR
ate channelCount:(int *)pChannelCount
```

```
{
    if (self.auxData == nil)
    {
        //初始化auxData
        NSURL *auxURL = [[NSBundle mainBundle] URLForResource:@"a.pcm" withExtension:n
il];
        if (auxURL)
        {
            self.auxData = [NSData dataWithContentsOfURL:auxURL options:0 error:nil];
            self.pPos = (void *)[self.auxData bytes];
        }
    }

    if (self.auxData)
    {
        int nLen = (int)[self.auxData length];
        if (self.pPos == 0)
            self.pPos = (void *)[self.auxData bytes];

        const void *pAuxData = [self.auxData bytes];
        if (pAuxData == NULL)
            return;

        *pSampleRate = 44100;
        *pChannelCount = 2;

        int nLeftLen = (int)(pAuxData + nLen - self.pPos);
        if (nLeftLen < *pDataLen) {
            self.pPos = (void *)pAuxData;
            *pDataLen = 0;
            return;
        }

        memcpy(pData, self.pPos, *pDataLen);
        self.pPos = self.pPos + *pDataLen;
    }
}
```

请注意：

1. Demo 中演示的是循环播放音频，**请开发者按照各自的需求实现该方法，不要直接复制。**
2. 如果数据足够，则copy *pDataLen 长度的数据到pData，如果不足要补齐静音达到SDK指定的 *pDataLen长度，或者设置 *pDataLen = 0。