混音

1 功能简介

ZegoAudioRoom SDK 为开发者提供了混音功能。

混音是指，SDK 从 App 获取一路音频数据，将其与采集的音频数据，整合为一路混音数据，进而推流。

直播过程中的**掌声，口哨，背景音等音效**均可通过混音实现。

2 步骤

混音的使用流程如下：

1. App 启用混音功能
2. App 设置混音音量
3. App 将混音数据传递给 SDK

2.1 启用混音功能

调用此 API 启用混音功能。

```
zego-api-audio-aux-oc.h

/**
 混音开关

 * 注意：
 * 1. 必须在初始化 SDK 后调用，可在需要混音的任一时间开启混音开关。
 * 2.当开启混音后，SDK 会在 -onAuxCallback:dataLen:sampleRate:channelCount:sideInfo:sideInfoLen:packet: 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。

 @param enable 开启/关闭混音开关，YES 表示开启混音，NO 表示关闭混音；默认为 NO（关闭混音）。
```

```
     @return YES 表示调用成功，能收到混音回调；NO 表示调用失败，不能收到混音回调。
     */
    - (BOOL)enableAux:(BOOL)enable;
```

> 请注意，后续操作均基于开启混音功能的基础上。

2.2 设置混音音量

启用混音后，调用此 API 调整混音音量。

```
zego-api-audio-aux-oc.h

/**
 设置混音音量

 @param volume 0~100，默认为 50
 */
- (void)setAuxVolume:(int)volume;
```

也可调用此 API 设置混音静音。

```
zego-api-audio-aux-oc.h

/**
 混音静音开关

 * 该接口调用时机无要求，开发者按需调用即可。

 @param mute YES: aux 输入播放静音，NO: 不静音。默认 NO
 @return YES 成功，NO 失败
 */
- (BOOL)muteAux:(BOOL)mute;
```

> 请注意，SDK 对上述两个混音音量相关的 API 的调用时机无要求，混音前或混音后调用均可，取决于用户需求。

2.3 App 传递数据给 SDK

启用混音后，SDK 通过此 API 获取待传递的混音数据。

zego-api-audio-aux-oc.h

```
/**
 混音音频数据的输入回调，当开启混音后，用户调用该 API 将混音数据传递给 SDK。

 * 注意：
 * 1.针对混音数据，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。
 * 2.用户根据实际的 PCM 音频填写采样率及声道数。
 * 3.为确保混音效果，请不要在此 API 中执行耗时操作。

 @param pData 待混音的音频数据
 @param pDataLen 一次传入的音频数据长度；SDK会提供好长度值，用户按照这个长度写入音频数据即可；如果填写的音频数据长度大于等于 pDataLen，则无需更改 pDataLen 的值;如果填写的音频数据长度小于 pDataLen，将 pDataLen 的值更改为0；当音频最后的尾音不足 SDK 提供的长度值时，又需要向 SDK 传入完整的音频数据，可以用静音数据补齐后再传给 SDK。
 @param pSampleRate 混音数据采样率，支持16k、32k、44.1k、48k
 @param pChannelCount 混音数据声道数，支持1、2
 @see -enableAux:
 */
- (void)onAuxCallback:(void *)pData dataLen:(int *)pDataLen sampleRate:(int *)pSampleRate channelCount:(int *)pChannelCount;
```

示例代码片段如下：

```
- (void)onAuxCallback:(void *)pData dataLen:(int *)pDataLen sampleRate:(int *)pSampleRate channelCount:(int *)pChannelCount
{
    if (self.auxData == nil)
    {
        //初始化 auxData，此处待混音的是 a.pcm 音频数据
        NSURL *auxURL = [[NSBundle mainBundle] URLForResource:@"a.pcm" withExtension:nil];
        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;

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

        if (pSampleRate)
            *pSampleRate = 44100;

        if (pChannelCount)
            *pChannelCount = 2;

        // 将 self.pPos 中的数据复制给 pData
        memcpy(pData, self.pPos, *pDataLen);

        // 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。
        self.pPos = self.pPos + *pDataLen;
    }
}
```

请注意：

1. 示例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. 由于 SDK 每次会取 20ms 的数据。如果开发者传入的 pDataLen 不为 0，则必须传递一个固定值。该值由数据的采样率、声道和 20ms 的时间计算得来。否则可能导致混音结果异常。