

连麦功能

连麦模块(connectMic) 提供连麦功能的集成，详细使用方式见下文。

一、设置连麦及连麦信息

1.1 设置连麦

观看页 SDK 默认不会设置连麦功能，如果需要连麦功能，开发者需要手动设置连麦功能，设置完成后即可调用连麦模块的 Api。

Api 方法： `setupConnectMic(pageInstance: ConnectMicPageInstance): Promise<ConnectMicResult>`

参数说明：

- pageInstance：页面或组件实例，`ConnectMicPageInstance` 类型，必传

返回值说明： 设置结果，`Promise<ConnectMicResult>` 类型

示例：

```
// 在小程序页面或组件 attached 中设置连麦并传入页面或组件实例
Component({
  lifetimes: {
    async attached() {
      // 设置连麦功能
      const result = await watchCore.connectMic.setupConnectMic(this);
      if (result.success) {
        console.log('设置成功');
      } else {
        console.log('设置失败', result.failReason);
      }
    },
  },
});
```

1.2 是否支持连麦功能

用于判断当前环境是否支持连麦功能。

Api 方法： `supportConnectMic(): SupportResult`

返回值说明： `SupportResult` 类型

示例：

```
const result = watchCore.connectMic.supportConnectMic();
console.log('是否支持连麦：', result.support ? '支持' : '不支持');
```

1.3 获取连麦信息

连麦模块的状态，数据均通过 `connectMicInfo` 保存，开发者可通过 `getConnectMicInfo` 方法获取连麦信息。

PS：通过 `ConnectMicEvents.ConnectMicInfoChange` 事件监听连麦事件改变。

Api 方法： `getConnectMicInfo(): ConnectMicStoreInfo`

返回值说明： 连麦信息， `ConnectMicStoreInfo` 类型，详细类型说明如下

属性名	说明	类型
<code>supportConnectMic</code>	当前环境是否支持连麦	<code>boolean</code>
<code>openMicStatus</code>	连麦状态，开启或关闭	<code>boolean</code>
<code>connectMicType</code>	连麦类型	<code>ConnectMicType</code>
<code>connectMicStatus</code>	用户连麦状态	<code>ConnectMicStatus</code>
<code>showJoinQueueNumberEnabled</code>	连麦排序显示开关	<code>boolean</code>
<code>currentMicIndex</code>	连麦顺序索引值(-1 表示不在队列)	<code>number</code>
<code>currentIsSpeaker</code>	当前用户是否为主讲	<code>boolean</code>
<code>localIsVideoMuted</code>	本地流是否关闭摄像头	<code>boolean</code>
<code>localIsAudioMuted</code>	本地流是否关闭麦克风	<code>boolean</code>

示例：

```
const info = watchCore.connectMic.getConnectMicInfo();
console.log('当前是否支持连麦功能', data.supportConnectMic);
console.log('当前是否开启连麦', data.openMicStatus);
```

1.4 判断连麦状态是否处于连麦中

通过 `isConnectMicing` 判断用户或传入的状态是否处于连麦中 (`ConnectMicStatus.Publishing` or `ConnectMicStatus.Connected`)。

Api 方法: `isConnectMicing(status?: ConnectMicStatus): boolean`

参数说明:

- status: 连麦状态, 不传则用当前状态, `ConnectMicStatus` 类型, 选传

返回值说明: 是否连麦中

示例:

```
const res = watchCore.connectMic.isConnectMicing();
if (res) {
  console.log('用户连麦中');
} else {
  console.log('用户没有连麦');
}
```

二、观众上麦

2.1 观众申请连麦

当讲师/主播开启连麦功能后, 用户即可申请连麦, 开发者可调用 `applyConnectMic` 方法申请连麦, 在申请时可调用 `cancelApplyConnectMic` 取消连麦申请。

PS: 通过 `ConnectMicEvents.AllowConnectMicApply` 监听连麦申请通过

Api 方法: `applyConnectMic(): Promise<ConnectMicResult>`

返回值说明: `Promise<ConnectMicResult>` 类型

示例:

```
import { ConnectMicError, ConnectMicEvents } from '@polyv/live-watch-miniprogram-sdk';

// 申请连麦
async function applyConnectMic() {
  const result = await watchCore.connectMic.applyConnectMic();
  if (result.success) {
    toast.success('连麦申请成功! 请等待主播同意');
    const info = watchCore.connectMic.getConnectMicInfo();
    console.log('当前连麦状态: ', info.connectMicStatus); // ConnectMicStatus.Applying
    return;
  }

  if (result.failReason === ConnectMicError.GetDevicePermissionFail) {
    toast.error('连麦申请失败! 未获取设备权限');
  }
}

watchCore.connectMic.eventEmitter.on(ConnectMicEvents.AllowConnectMicApply, () => {
  toast.success('讲师已通过你的连麦申请');
});
```

2.2 取消连麦申请

当观众申请连麦后，调用 `cancelApplyConnectMic` 即可取消连麦申请。

Api 方法： `cancelApplyConnectMic(): void`

示例：

```
watchCore.connectMic.cancelApplyConnectMic();

const info = watchCore.connectMic.getConnectMicInfo();
console.log('当前连麦状态: ', info.connectMicStatus); // ConnectMicStatus.NotConnect
```

2.3 开始推送本地流

讲师通过连麦申请后，连麦模块会触发 `ConnectMicEvents.ConnectMicDataChange` 事件回调新的连麦数据列表，此时列表中会存在一个 `isLocal` 为 `true` 的连麦对象。在获取新的列表后创建连麦播放器，并在连麦播放器组件的 `attached` 生命周期中调用 `startStream` 方法推流，详细可参考 demo 中的 `<connect-mic-player>` 组件。

Api 方法： `startStream(): void`

示例：

```

Component({
  lifetimes: {
    attached() {
      // 数据为本地流，调用方法推流
      if (this.data.isLocal) {
        watchCore.connectMic.startStream();
      }
    }
  },
});

```

2.4 结束连麦

当讲师通过观众的连麦申请并连麦成功后，通过 `endConnectMic` 可手动结束观众的连麦。

PS：通过 [ConnectMicEvents.LeaveConnectMicSuccess](#) 监听离开成功。

Api 方法： `endConnectMic(): Promise<void>`

示例：

```

// 结束连麦
watchCore.connectMic.endConnectMic();

watchCore.connectMic.eventEmitter(ConnectMicEvents.LeaveConnectMicSuccess, () => {
  toast.success('结束连麦成功');
  const info = watchCore.connectMic.getConnectMicInfo();
  console.log('当前连麦状态：', info.connectMicStatus); // ConnectMicStatus.NotConnect
});

```

三、摄像头设置

3.1 开启本地摄像头

通过 `enabledVideo` 方法开启本地摄像头，可通过 [ConnectMicEvents.LocalVideoMuteChange](#) 事件监听本地摄像头的开关。

Api 方法： `enabledVideo(): void`

示例：

```

// 开启本地摄像头
watchCore.connectMic.enabledVideo();

```

3.2 关闭本地摄像头

通过 `disabledVideo` 方法关闭本地摄像头，可通过 [ConnectMicEvents.LocalVideoMuteChange](#) 事件监听本地摄像头的开关。

Api 方法： `disabledVideo(): void`

示例：

```
// 关闭本地摄像头
watchCore.connectMic.disabledVideo();
```

3.3 切换前后置摄像头

Api 方法： `switchCamera(): void`

四、麦克风设置

4.1 开启本地麦克风

通过 `enabledAudio` 方法开启本地麦克风，可通过 [ConnectMicEvents.LocalAudioMuteChange](#) 事件监听本地麦克风的开关。

Api 方法： `enabledAudio(): void`

示例：

```
// 开启本地麦克风
watchCore.connectMic.enabledAudio();
```

4.2 关闭本地麦克风

通过 `disabledAudio` 方法关闭本地麦克风，可通过 [ConnectMicEvents.LocalAudioMuteChange](#) 事件监听本地麦克风的开关。

Api 方法： `disabledAudio(): void`

示例：

```
// 关闭本地麦克风
watchCore.connectMic.disabledAudio();
```

五、其他

5.1 用户连麦状态

Enum 枚举： `ConnectMicStatus`

常量	枚举成员	说明
<code>'notConnect'</code>	<code>ConnectMicStatus.NotConnect</code>	未连麦
<code>'applying'</code>	<code>ConnectMicStatus.Applying</code>	连麦申请中
<code>'publishing'</code>	<code>ConnectMicStatus.Publishing</code>	推流中
<code>'connected'</code>	<code>ConnectMicStatus.Connected</code>	已连麦
<code>'error'</code>	<code>ConnectMicStatus.Error</code>	连麦异常

5.2 连麦类型

用于区分讲师开启的是视频连麦或音频连麦

Enum 枚举： `ConnectMicType`

常量	枚举成员	说明
<code>'video'</code>	<code>ConnectMicType.Video</code>	视频连麦
<code>'audio'</code>	<code>ConnectMicType.Audio</code>	音频连麦