

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(): Promise<ConnectMicResult>`

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

示例：

```
// 设置连麦功能
const result = await watchCore.connectMic.setupConnectMic();

if (result.success) {
  // 打开设备设置
  watchCore.connectMic.openDeviceSetting();
} 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>supportFacingMode</code>	当前环境是否支持切换前后摄像头	<code>boolean</code>
<code>facingMode</code>	当前前后摄像头	<code>FacingMode</code>
<code>mirrorEnabled</code>	是否开启镜像	<code>boolean</code>
<code>openMicStatus</code>	连麦状态，开启或关闭	<code>boolean</code>
<code>inviteStatus</code>	邀请上麦状态	<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>autoConnect</code>	点击开始连麦是否立刻上麦	<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/主讲开启连麦功能后, 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-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! 请等待主讲同意');
    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 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
```

2.4 推送本地连麦流

讲师通过连麦申请后，通过 `publishLocalStream` 进行连麦流推送，注意该方法需要在 `ConnectMicEvents.LocalStreamInitied` 事件触发后调用。

推流成功后将触发 `ConnectMicEvents.PublishStreamSuccess` 事件。

建议通过连麦用户节点 `ConnectMicItem.publishStream` 方法进行推送。

Api 方法： `publishLocalStream(options: PublishStreamOptions): Promise<ConnectMicResult>`

参数说明：

- options：推流参数，`PublishStreamOptions` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
<code>element</code>	渲染节点	<code>HTMLDivElement</code>	是	-
<code>control</code>	控制栏	<code>boolean</code>	否	<code>true</code>
<code>fit</code>	视频裁剪模式	<code>ConnectMicFitType</code>	否	<code>ConnectMicFitType.Cover</code>
<code>profile</code>	推流属性	<code>StreamProfile</code>	否	<code>'240p'</code>

返回值说明： `Promise<ConnectMicResult>` 类型

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.LocalStreamInitied, () => {
  watchCore.connectMic.publishLocalStream({
    element: 'NodeElement',
  });
});
```

2.5 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(): 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、小班课场景连麦

3.1 以观众模式加入连麦房间, 仅订阅讲师流, 不推本地流

在小班课场景下, 观众可以通过此方法加入房间并观看讲师的流。此方法仅在小班课场景下有效, 且连麦已初始化 (setupConnectMic) 后才能调用。

Api 方法: `joinAsAudience(): Promise<joinAsAudienceResult>`

从 v2.13.0

****返回值说明:** ****** 加入结果, ``Promise<joinAsAudienceResult>`` 类型

`<a id="classmethoddock_plvconnectmicmodule_leaveaudience"></a>`

3.2 离开观众模式

****Api 方法:** ****** ``leaveAudience(): Promise<void>``

> 从 v2.13.0

`` ``` 版本开始支持

`<a id="classmethoddock_plvconnectmicmodule_stopsmallclassstat"></a>`

3.3 小班课场景停止统计

****Api 方法:** ****** ``stopSmallClassStat(): void``

> 从 v2.13.0 版本开始支持

`<a id="classmethoddock_plvconnectmicmodule_isaudiencemodejoined"></a>`

3.4 当前是否为观众模式

****Api 方法:** ****** ``isAudienceModeJoined(): boolean``

> 从 v2.13.0 版本开始支持

四、设备设置

`<a id="classmethoddock_plvconnectmicmodule_opendevicessetting"></a>`

4.1 打开设备设置界面

连麦模块提供内置的设备设置界面, 通过 ``openDeviceSetting`` 方法打开设备设置界面, 用于切换摄像头、麦克风设备等操作, 当需要关闭时可调用 `[closeDeviceSetting](/live/js/new_sdk/live_watch_sdk/articles/modules/connect-mic/function.md?id=classmethoddock_plvconnectmicmodule_closedevicessetting)` 进行关闭。

****Api 方法:** ****** ``openDeviceSetting(): void``

****示例:** ******

`` ```js

`watchCore.connectMic.openDeviceSetting();`

4.2 关闭设备设置界面

Api 方法: `closeDeviceSetting(): void`

示例:

```
watchCore.connectMic.closeDeviceSetting();
```

五、摄像头设置

5.1 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();
```

5.2 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();
```

5.3 切换前后摄像头

Api 方法： `changeFacingMode(facingMode: FacingMode): void`

从 v2.6.0 版本开始支持

参数说明：

- `facingMode`：摄像头模式，`FacingMode` 类型，必传

示例：

```
watchCore.connectMic.changeFacingMode(FacingMode.Environment);
```

六、麦克风设置

6.1 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();
```

6.2 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();
```

七、邀请连麦

7.1 打开邀请上麦界面

连麦模块提供内置的邀请上麦界面，当监听到 [ConnectMicEvents.InviteConnectMic](#) 讲师邀请上麦事件后，通过 `openInviting` 方法打开邀请上麦界面，需要关闭时可调用 `closeInviting` 进行关闭。

PS: 观众点击同意时，可能因连麦人数到达上限而连麦失败，通过 [ConnectMicEvents.ConnectMicOverLimit](#) 事件监听并页面提示

Api 方法： `openInviting(): void`

示例：

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

watchCore.connectMic.eventEmitter.on(ConnectMicEvents.InviteConnectMic, () => {
  // 打开邀请连麦窗口
  watchCore.connectMic.openInviting();
});
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ConnectMicOverLimit, () => {
  toast.error('连麦失败，连麦人数已到达上限');
});
```

7.2 关闭邀请上麦界面

Api 方法： `closeInviting(): void`

示例：

```
watchCore.connectMic.closeInviting();
```

7.3 接受讲师上麦邀请

用于在自定义邀请上麦 UI 中调用，触发后等同于 SDK 内置邀请窗口的"同意"操作。

Api 方法： `acceptInvite(): void`

从 v2.17.0 版本开始支持

示例：

```
watchCore.connectMic.acceptInvite();
```

7.4 拒绝讲师上麦邀请

用于在自定义邀请上麦 UI 中调用，触发后等同于 SDK 内置邀请窗口的"拒绝"操作。

Api 方法： `refuseInvite(): void`

从 v2.17.0 版本开始支持

示例：

```
watchCore.connectMic.refuseInvite();
```

7.5 获取邀请上麦倒计时

返回当前邀请上麦的剩余时间和总时间，可在 [ConnectMicEvents.InviteCountDown](#) 事件中使用。

Api 方法： `getInviteCountDown(): InviteCountDownData`

从 v2.17.0 版本开始支持

返回值说明： `InviteCountDownData` 类型

示例：

```
const { remain, total } = watchCore.connectMic.getInviteCountDown();
console.log(`剩余 ${remain}s / 总 ${total}s`);
```

八、本地预览

8.1 开启本地预览

用于在连麦面板中预览本地摄像头，未连麦时调用，连麦中无需预览。

Api 方法： `startPreview(config: PreviewConfig): Promise<PreviewHandle>`

从 v2.17.0

```
await watchCore.connectMic.startPreview({ videoEl: el, video: true });
``` 版本开始支持

参数说明:

- config: `PreviewConfig` 类型, 必传

返回值说明: ** `Promise<PreviewHandle>` 类型

示例:

```ts
```

8.2 停止本地预览

Api 方法: `stopPreview(): void`

从 v2.17.0 版本开始支持

示例:

```
watchCore.connectMic.stopPreview();
```

8.3 是否正在本地预览

Api 方法: `isPreviewing(): boolean`

从 v2.17.0 版本开始支持

8.4 获取本地预览的当前音量 (0~1)

仅在预览中可用; 非预览态返回 0。

Api 方法: `getPreviewVolume(): number`

从 v2.17.0 版本开始支持

九、本地音视频开关

9.1 获取当前本地音视频静音状态

返回 { video, audio }, true 表示已关闭/静音。未连麦（预览阶段）时返回设备预设值，已连麦时返回底层推流实时状态。

Api 方法: `getLocalMuteStatus(): Object`

从 v2.17.0 版本开始支持

返回值说明: `Object` 类型，详细类型说明如下

属性名	说明	类型
video	-	boolean
audio	-	boolean

9.2 预设/调整本地音视频开关

未推流时仅写入预设值，`acceptInvite` / `publish` 后生效；已推流时等价于调用 `enable/disable` 对应轨道。

Api 方法: `setLocalMuteStatus(option: Object): void`

从 v2.17.0 版本开始支持

参数说明:

- option: `Object` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
video	-	boolean	否	-
audio	-	boolean	否	-

十、连麦网络状态

10.1 获取连麦网络信息

Api 方法： `getNetworkInfo(): ConnectMicNetworkInfo`

返回值说明： 连麦网络信息， `ConnectMicNetworkInfo` 类型，详细类型说明如下

属性名	说明	类型
<code>uplinkNetworkQuality</code>	连麦上行网络质量	<code>UplinkNetworkQuality</code>
<code>uplinkNetworkStatus</code>	连麦上行网络状态	<code>NetworkStatus</code>
<code>downlinkNetworkQuality</code>	连麦下行网络质量	<code>DownlinkNetworkQuality</code>
<code>downlinkNetworkStatus</code>	连麦下行网络状态	<code>NetworkStatus</code>

示例：

```
const network = watchCore.connectMic.getNetworkInfo();
console.log('推流网络状态', network.uplinkNetworkStatus);
console.log('拉流网络状态', network.downlinkNetworkStatus);
```

十一、本地流镜像

11.1 获取镜像状态

获取当前连麦流的镜像状态。

Api 方法： `getMirrorEnabled(): boolean`

从 v2.10.0 版本开始支持

返回值说明： 是否开启镜像

示例：

```
const mirrorEnabled = watchCore.connectMic.getMirrorEnabled();
console.log('镜像状态: ', mirrorEnabled ? '开启' : '关闭');
```

11.2 切换镜像开关状态

通过 `toggleMirror` 方法可以切换本地连麦流的镜像状态。开启镜像后，本地流画面会水平翻转。

Api 方法： `toggleMirror(enabled?: boolean): void`

从 v2.10.0 版本开始支持

参数说明：

- `enabled`：是否开启镜像，不传则切换当前状态，`boolean` 类型，选传

示例：

```
// 开启镜像
watchCore.connectMic.toggleMirror(true);

// 关闭镜像
watchCore.connectMic.toggleMirror(false);

// 切换镜像状态
watchCore.connectMic.toggleMirror();
```

十二、其他

12.1 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

常量	枚举成员	说明
'connected'	ConnectMicStatus.Connected	已连麦
'error'	ConnectMicStatus.Error	连麦异常

12.2 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
'video'	ConnectMicType.Video	视频连麦
'audio'	ConnectMicType.Audio	音频连麦