

模块事件

一、连麦信息改变

说明：连麦信息通过 `connectMic.getConnectMicInfo()` 获取，当连麦信息改变时会触发该事件。

Event 事件： `ConnectMicEvents.ConnectMicInfoChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>connectMicInfo</code>	连麦信息	<code>ConnectMicStoreInfo</code>

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ConnectMicInfoChange, () => {
  const connectMicInfo = watchCore.connectMic.getConnectMicInfo();
  console.log('连麦信息更新', connectMicInfo);
});
```

二、连麦网络信息改变

说明：连麦网络状态信息通过 `connectMic.getNetworkInfo()` 获取，当连麦网络状态信息改变时会触发该事件。

Event 事件： `ConnectMicEvents.NetworkInfoChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>networkInfo</code>	连麦网络信息	<code>ConnectMicNetworkInfo</code>

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.NetworkInfoChange, () => {
  const networkInfo = watchCore.connectMic.getNetworkInfo();
  console.log('网络状态改变', networkInfo);
});
```

三、讲师开启连麦功能

说明： 讲师开启连麦功能后，触发该事件，开启后页面即可显示连麦功能。

Event 事件： `ConnectMicEvents.OpenConnectMic`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>type</code>	连麦类型	<code>ConnectMicType</code>

示例：

```
import { ConnectMicType } from '@polyv/live-watch-sdk';
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.OpenConnectMic, (data) => {
  const type = data.type === ConnectMicType.Video ? '视频' : '音频';
  toast.info(`讲师开启了${type}连麦，赶快和主讲连麦互动吧~`);
});
```

四、讲师关闭连麦功能

说明： 讲师关闭连麦功能后触发该事件

Event 事件： `ConnectMicEvents.CloseConnectMic`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>type</code>	连麦类型	<code>ConnectMicType</code>

示例：

```
import { ConnectMicType } from '@polyv/live-watch-sdk';
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.CloseConnectMic, (data) => {
  const type = data.type === ConnectMicType.Video ? '视频' : '音频';
  toast.info(`讲师已结束${type}连麦`);
});
```

五、连麦列表修改

说明：当观众上下麦或用户流加入/退出后，连麦列表改变，通过该事件监听连麦用户列表更新。

Event 事件： `ConnectMicEvents.ConnectMicListChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>connectMicList</code>	连麦列表	<code>ConnectMicItem[]</code>

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ConnectMicListChange, () => {
  const connectMicList = watchCore.connectMic.getConnectMicList();
  console.log('连麦用户列表更新', connectMicList);
});
```

六、讲师挂断当前连麦者

说明：当讲师挂断当前连麦用户，连麦模块将回调该事件。

Event 事件： `ConnectMicEvents.TeacherHangUp`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.TeacherHangUp, () => {
  toast.info('讲师已挂断你的连麦');
});
```

七、邀请上麦

说明：当讲师邀请观众上麦时触发该事件。

Event 事件： `ConnectMicEvents.InviteConnectMic`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.InviteConnectMic, () => {  
  watchCore.connectMic.openInviting();  
});
```

八、邀请上麦回复

说明： 当学员对被邀请上麦是做处接受或拒绝操作时触发该事件

Event 事件： `ConnectMicEvents.InviteConnectMicAnswer`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.InviteConnectMic, (data) => {  
  console.log(`学员是否拒绝上麦: ${data.accept}, 拒绝上麦是否是因为操作超时: ${data.timeout}`)  
});
```

九、设备采集失败

说明： 媒体设备(摄像头/麦克风)自动恢复采集失败，摄像头、麦克风接口松动，或被其他应用占用。恢复采集失败时触发该事件。

Event 事件： `ConnectMicEvents.DeviceRecoverFail`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.DeviceRecoverFail, () => {  
  toast.error('自动采集失败，请检查设备正常连接后刷新页面重新进入');  
});
```

十、连麦人数到达上限

说明： 观众点击同意邀请上麦时，可能因连麦人数到达上限而连麦失败，通过该事件监听并页面提示。

Event 事件： `ConnectMicEvents.ConnectMicOverLimit`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ConnectMicOverLimit, () => {
  toast.error('连麦失败，连麦人数已到达上限');
});
```

十一、当前主讲状态改变

说明：当讲师授予/撤销当前连麦用户的主讲权限时触发该事件。

Event 事件： `ConnectMicEvents.CurrentSpeakerStatusChanged`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>currentIsSpeaker</code>	当前用户是否为主讲	<code>boolean</code>

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.CurrentSpeakerStatusChanged, (data) => {
  if (data.currentIsSpeaker) {
    toast.info('讲师已授予您主讲权限');
  } else {
    toast.info('讲师已收回您的主讲权限');
  }
});
```

十二、讲师同意连麦申请

说明：观众申请连麦，讲师同意连麦申请后触发该事件。

Event 事件： `ConnectMicEvents.AllowConnectMicApply`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.AllowConnectMicApply, (data) => {
  toast.success('讲师已同意你的连麦申请');
});
```

十三、本地流初始化成功

说明： 讲师同意连麦申请，本地流初始化完成后触发该事件，触发后即可推送本地连麦流。

Event 事件： `ConnectMicEvents.LocalStreamInited`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>micItem</code>	连麦用户节点	<code>ConnectMicItem</code>

示例：

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

十四、本地流推送成功

说明： 本地连麦流推送成功后触发该事件，触发后连麦状态为 `ConnectMicStatus.Connected`。

Event 事件： `ConnectMicEvents.PublishStreamSuccess`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.PublishStreamSuccess, () => {
  toast.success('上麦成功');
  const connectMicInfo = watchCore.connectMic.getConnectMicInfo();
  console.log('连麦状态', connectMicInfo.connectMicStatus); // ConnectMicStatus.Connected
});
```

十五、离开连麦成功

说明： 本地连麦流离开后触发该事件，触发后连麦状态为 `ConnectMicStatus.NotConnect`。

Event 事件： `ConnectMicEvents.LeaveConnectMicSuccess`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.LeaveConnectMicSuccess, () => {
  toast.success('你已下麦');
  const connectMicInfo = watchCore.connectMic.getConnectMicInfo();
  console.log('连麦状态', connectMicInfo.connectMicStatus); // ConnectMicStatus.NotConnect
});
```

十六、远端连麦用户麦克风开关改变

Event 事件： `ConnectMicEvents.RemoteAudioMuteChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>micItem</code>	连麦用户节点	<code>ConnectMicItem</code>
<code>isAudioMuted</code>	是否关闭麦克风	<code>boolean</code>

十七、远端连麦用户摄像头开关改变

Event 事件： `ConnectMicEvents.RemoteVideoMuteChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>micItem</code>	连麦用户节点	<code>ConnectMicItem</code>
<code>isVideoMuted</code>	是否关闭摄像头	<code>boolean</code>

十八、本地麦克风开关改变

Event 事件： `ConnectMicEvents.LocalAudioMuteChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>micItem</code>	连麦用户节点	<code>ConnectMicItem</code>

属性名	说明	类型
isAudioMuted	是否关闭麦克风	boolean

十九、本地摄像头开关改变

Event 事件： `ConnectMicEvents.LocalVideoMuteChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
micItem	连麦用户节点	ConnectMicItem
isVideoMuted	是否关闭摄像头	boolean

二十、切换第一画面

说明： 讲师切换第一画面时触发该事件

Event 事件： `ConnectMicEvents.SwitchMaster`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
currentMaster	当前第一画面的连麦用户节点	ConnectMicItem
previousMaster	上一次的连麦用户节点	ConnectMicItem

二十一、屏幕共享信息改变

说明： 屏幕恭喜信息改变时触发该事件

Event 事件： `ConnectMicEvents.ScreenShareInfoChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
screenInfoChange	屏幕共享信息	ScreenShareInfo

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ScreenShareInfoChange, () => {  
  const screenShareInfo = watchCore.connectMic.getScreenShareInfo();  
  console.log('屏幕共享信息改变', screenShareInfo);  
});
```

二十二、屏幕共享开始

说明：观众开始屏幕共享时触发该事件。

Event 事件： ConnectMicEvents.ScreenShareStart

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ScreenShareStart, () => {  
  toast.info('你已进入屏幕共享');  
});
```

二十三、屏幕共享结束

说明：观众结束屏幕共享时触发该事件。

Event 事件： ConnectMicEvents.ScreenShareStop

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ScreenShareStart, () => {  
  toast.info('你已退出屏幕共享');  
});
```

二十四、举手申请连麦超时

说明：观众举手申请连麦超时时触发该事件。

Event 事件： ConnectMicEvents.ApplyTimeOut

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.ApplyTimeOut, () => {
  toast.info('你已申请超时');
});
```

二十五、前后摄像头切换

说明： 前后摄像头切换时触发该事件

Event 事件： `ConnectMicEvents.FacingModeChange`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.FacingModeChange, () => {
  toast.info(watchCore.connectMic.getConnectMicInfo().facingMode);
});
```

二十六、用户流播放失败

说明： 用户流播放失败时触发该事件

Event 事件： `ConnectMicEvents.UserStreamPlayFailed`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>streamId</code>	流ID	<code>string</code>

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.UserStreamPlayFailed, (data) => {
  toast.error('用户流播放失败');
});
```

二十七、邀请上麦倒计时

说明：收到讲师邀请后，距离自动拒绝的剩余时间发生变化时触发该事件，开发者可用于自定义邀请上麦 UI 的倒计时显示。

Event 事件： `ConnectMicEvents.InviteCountDown`

回调参数： `InviteCountDownData`

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.InviteCountDown, (data) => {
  console.log(`剩余 ${data.remain}s / 总 ${data.total}s`);
});
```

二十八、本地音视频开关变更（预设 / 预览阶段）

说明：未连麦时（预设 / 预览阶段）本地摄像头、麦克风开关发生变化时触发，例如在设备检测面板中修改了设备开关。开发者可用于同步自定义连麦面板的设备按钮状态。

Event 事件： `ConnectMicEvents.LocalMuteStatusChange`

回调参数： `Object` 对象，详细类型说明如下

属性名	说明	类型
<code>video</code>	是否关闭摄像头	<code>boolean</code>
<code>audio</code>	是否关闭麦克风	<code>boolean</code>

示例：

```
watchCore.connectMic.eventEmitter.on(ConnectMicEvents.LocalMuteStatusChange, (data) => {
  console.log('摄像头是否关闭: ', data.video, '麦克风是否关闭: ', data.audio);
});
```