

屏幕共享

本文档主要提供连麦模块下的屏幕共享功能的 Api 说明。

一、使用方式

1.1 获取屏幕共享信息

Api 方法：`getScreenShareInfo(): ScreenShareInfo`

返回值说明：屏幕共享信息，`ScreenShareInfo` 类型，详细类型说明如下

属性名	说明	类型
<code>supportScreenShare</code>	当前环境是否支持屏幕共享	<code>boolean</code>
<code>screenShareMode</code>	屏幕共享模式	<code>ScreenShareMode</code>
<code>audioMixEnabled</code>	屏幕共享混音开关	<code>boolean</code>
<code>isScreenSharing</code>	当前是否屏幕共享中	<code>boolean</code>
<code>supportPip</code>	当前环境是否支持画中画	<code>boolean</code>
<code>isPiping</code>	当前是否正在画中画	<code>boolean</code>

示例：

```
const screenShareInfo = watchCore.connectMic.getScreenShareInfo();
console.log('是否支持屏幕共享', screenShareInfo.supportScreenShare);
console.log('正在屏幕共享中', screenShareInfo.isScreenSharing);
```

1.2 设置屏幕共享配置

通过 `setScreenShareConfig` 设置屏幕共享配置，注意屏幕共享期间无法调用该方法进行屏幕共享设置。

Api 方法：`setScreenShareConfig(config?: ScreenShareConfig): ScreenShareResult`

参数说明：

- config：屏幕共享配置，ScreenShareConfig 类型，选传，默认 {}，详细类型说明如下

参数名	说明	类型	必须	默认值
screenShareMode	共享模式	ScreenShareMode	否	-
audioMixEnabled	混音开关	boolean	否	-

返回值说明： 设置结果， ScreenShareResult 类型

示例：

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

const result = await watchCore.connectMic.setScreenShareConfig({
  // 共享模式：动态画面为主
  screenShareMode: ScreenShareMode.Motion,
  // 开启混音
  audioMixEnabled: true,
});

if (result.success) {
  console.log('设置成功');
}
```

1.3 开始屏幕共享

通过 startScreenShare 开始屏幕共享，通过 ConnectMicEvents.ScreenShareStart 事件监听屏幕共享开始，注意开始前浏览器需要获取用户授权，如果用户拒绝授权则返回开始失败结果。

Api 方法： startScreenShare(): Promise<ScreenShareResult>

返回值说明： Promise<ScreenShareResult> 类型

示例：

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

async function startScreenShare() {
  const result = await watchCore.connectMic.startScreenShare();
  if (result.success) {
    const info = watchCore.connectMic.getScreenShareInfo();
    console.log('已进入屏幕共享', info.isScreenSharing); // true
    return;
  }

  if (result.failReason === ScreenShareError.BrowserNotSupport) {
    toast.error('屏幕共享失败，您拒绝了系统权限');
  }
}
```

1.4 停止屏幕共享

通过 `stopScreenShare` 停止屏幕共享，通过 [ConnectMicEvents.ScreenShareStop](#) 事件监听屏幕共享结束。

Api 方法： `stopScreenShare(): Promise<ScreenShareResult>`

返回值说明： `Promise<ScreenShareResult>` 类型

示例：

```
// 结束屏幕共享
watchCore.connectMic.stopScreenShare();
```

1.5 显示用户摄像头（画中画）

屏幕共享开启后，可通过 `enterPip` 显示用户的摄像头，进入画中画模式。

Api 方法： `enterPip(): Promise<void>`

示例：

```
// 显示画中画
await watchCore.connectMic.enterPip();
const info = watchCore.connectMic.getScreenShareInfo();
console.log('是否正在画中画模式', info.isPiping); // true
```

1.6 隐藏用户摄像头（画中画）

屏幕共享开启后，可通过 `exitPip` 隐藏用户的摄像头，推出画中画模式。

Api 方法： `exitPip(): Promise<void>`

示例：

```
// 隐藏画中画
await watchCore.connectMic.exitPip();
const info = watchCore.connectMic.getScreenShareInfo();
console.log('是否正在画中画模式', info.isPiping); // false
```

二、其他

2.1 屏幕共享异常原因

Enum 枚举： `ScreenShareError`

常量	枚举成员	说明
'Unknown'	<code>ScreenShareError.Unknown</code>	未知
'BrowserNotSupport'	<code>ScreenShareError.BrowserNotSupport</code>	当前环境不支持屏幕共享
'ChannelNotSupport'	<code>ScreenShareError.ChannelNotSupport</code>	当前频道不支持屏幕共享
'PermissionDenied'	<code>ScreenShareError.PermissionDenied</code>	用户权限被拒绝
'SystemPermissionDenied'	<code>ScreenShareError.SystemPermissionDenied</code>	系统权限被拒绝
'ScreenShareing'	<code>ScreenShareError.ScreenShareing</code>	屏幕共享中，无法设置屏幕共享配置