

云手机PC音视频SDK文档

目录

云手机PC音视频SDK文档	
文档更新	
简介	
接口开发语言	
• 1.1 C 接口头文件CloudPhoneInterface_c.h	
• 1.2 C++接口头文件CloudPhoneInterface.h, CloudPhoneDef.h	
云机相关信息及列表等	
控制云机	
云机拉流	
运行环境	
SDK导入	
API接口类CloudPhoneInterface	
• 0. 全局初始化	
• 0.1 sdk版本号获取	
• 0.2 sdk获取本机设备id字符串	
• 0.3 设置sdk日志目录	
• 一. 云机操作部分	
• 1.1. 启动应用	
• 1.2. 停止应用	
• 1.3. 卸载应用	
• 1.4. 隐藏应用	
• 1.5. 显示应用	
• 1.6. 应用root	
• 1.7. 系统root	
• 1.8. 开启预览图	
• 1.9. 关闭预览图	
• 1.10. 位置服务	
• 1.11. 发送剪切板	
•	

1.12. 设置卡分辨率	
• 1.13. 设置卡dpi	-----
• 1.14. 应用清理	-----
• 1.15. 应用启用	-----
• 1.16. 应用停用	-----
• 1.17. 设备新机	-----
• 1.18. 消息透传	-----
• 1.19. 截图	-----
• 1.20. 上传文件	-----
• 1.21. 下载文件	-----
• 1.21.1 下载文件指定存储目录与重命名	-----
• 1.22. 吹一吹	-----
• 1.23. 摇一摇	-----
• 1.24. 获取屏幕横竖屏状态	-----
• 1.25. 单例同步操作发送键盘值	-----
• 1.26. 单例同步操作发送鼠标事件	-----
• 1.27. 设置系统导航栏显示和隐藏	-----
• 1.28. 未拉流切换输入法	-----
• 1.29. 文本透传	-----
• 1.30. 获取系统导航栏状态	-----
• 1.31. 获取分辨率	-----
• 1.32. 消息订阅	-----
• 1.33. 取消订阅	-----
• 1.34. 单例同步操作发送安卓键盘值到云机支持按下抬起	-----
• 二. 云机拉流部分	-----
• 2.1. 拉取云机音视频流	-----
• iceAddressInfo 类型说明	-----
• 2.2. 断开拉流	-----
• 2.3. 发送键盘key	-----
• 2.4. 设置最大拉流码率上限	-----
• 2.5. 暂停拉流	-----
• 2.6. 恢复拉流	-----
• 2.7. 拉流过程中设置卡分辨率及dpi	-----
• 2.8. 拉流过程中改变拉流视频宽高	-----

- 2.9. 拉流过程获取拉流状态 -----
- 2.10. 音频开 -----
- 2.11. 音频关 -----
- 2.12. 暂停云机 -----
- 2.13. 恢复云机 -----
- 2.14. 设置编码类型 -----
- 2.15. 设置画质 -----
- 2.16. 设置屏幕旋转角度 -----
- 2.17. 发送鼠标事件 -----
- 2.18. 获取视频流信息 -----
- 2.19. 切换视频渲染窗口 -----
- 2.20. 注入鼠标多点触控 -----
- 2.21. 拉流过程发送安卓键盘码到云机 -----
- 2.22. 拉流过程中切换云机键盘 -----
- 2.23. 拉流过程中键盘为真机键盘的场景发送文本或字符、键盘码 -----
- 2.24. 拉流过程中键盘为真机键盘的场景清楚输入框最后一个字符。... -----
- 2.25. 拉流过程中订阅相机麦克风，多路拉流时获得相机麦克风使用权 -----
- 2.26. 拉流控制播放声音，拉流前设置拉流成功默认不播放声音，拉... -----
- 2.27. 设置渲染缩放模式，默认是铺满渲染 -----
- 三. 监听回调部分 监听CloudPhoneObserver -----
- 3.1. 启动应用返回 -----
- 3.2. 停止应用返回 -----
- 3.3. 卸载应用返回 -----
- 3.4. 隐藏应用返回 -----
- 3.5. 隐藏应用返回 -----
- 3.6. 应用root返回 -----
- 3.7. 系统root返回 -----
- 3.8. 开启预览图返回 -----
- 3.9. 停止预览图返回 -----
- 3.10. 设置虚拟定位返回 -----
- 3.11. 发送剪切板返回 -----
- 3.12. 设置系统分辨率返回 -----
- 3.13. 设置系统dpi返回 -----
- 3.14. 预览图数据返回 -----

- 3.15. 获取拉流状态返回 -----
- 3.16. 拉流连接状态通知 -----
- 3.17. 拉流视频宽高改变通知 -----
- 3.18. 暂停云机通知 -----
- 3.19. 恢复云机通知 -----
- 3.20. 音频开通知 -----
- 3.21. 音频关通知 -----
- 3.22. 画质设置通知 -----
- 3.23. 编码类型设置通知 -----
- 3.24. 横竖屏设置通知 -----
- 3.25. 订阅消息通知 -----
- 3.26. 取消订阅消息通知 -----
- 3.27. 消息透传通知 -----
- 3.28. 订阅app消息通知 -----
- 3.29. 截图通知 -----
- 3.30. 文件上传通知 -----
- 3.31. 文件下载通知 -----
- 3.32. 应用清理通知 -----
- 3.33. 设备新机通知 -----
- 3.34. 超时通知 -----
- 3.35. 应用停用通知 -----
- 3.36. 应用启用通知 -----
- 3.37. 获取导航栏状态通知 -----
- 3.38. 获取分辨率通知 -----
- 3.39. 触控鼠标事件回调 -----
- 3.40. 拉流鉴权通知回调 -----
- 3.41. 相机麦克风被抢占通知 -----
- 3.42. 拉流过程中真机键盘窗口隐藏通知 -----
- 3.43. 拉流过程中真机键盘窗口显示通知 -----
- 3.44. 拉流过程中真机键盘切换通知 -----

群控支持两种模式 -----

- 1. 模式一：由上层实现遍历每个从控云机通过上述的云机部分的接口... -----
 - 1.1.1 sendKeyForGroupControl 发送键盘key 按下抬起一起触发 -----
 - 1.1.2 sendKeyCodeForGroupControl 发送键盘key 按下抬起上层... -----

- 1.1.3 sendMouseForGroupControl 发送鼠标 -----
- 1.1.4 sendMultiTouchForGroupControl 发送鼠标，支持多点触控 -----
- 优势：接入灵活由上层控制 -----
- 劣势：群控效率低，群控数量可能受限，上几百台云机上层分发效... -----
- 2. 模式二：由使用后台api创建群控会话，设置主控和从控，主控的所... -----
- 2.1.1 后台openapi创建群控会话相关api -----
- 2.1.2 在拉完云机列表后，通过创建群控会话将所有云机创建一个... -----
- 2.1.3 在用户需要群控时，将用户选中的云机，调用从控生效选定... -----
- 2.1.4 上层不需要分发，操作主控主控的鼠标键盘事件会被服务器... -----
- 2.1.5 主控从控等切换操作请查阅后台openapi接口文档。 -----
- 优势：同步效率高，从控数量无限制。 -----
- 3. 与其他厂商做混合群控处理方案 -----
- 3.1 选择群控模式一简单便捷，但需要考虑劣势选是否满足贵司产品... -----
- 3.2 选群控模式二与其他厂商做混合群控，逻辑稍微复杂，下面简单... -----

[TOC]

云手机PC音视频SDK文档

云手机PC音视频SDK主要功能是为windows 桌面应用赋予云手机使用能力，可以通过SDK 连接云手机，完成对云手机的一系列操作









文档更新







日期	版本	描述	作者
2024/10/30	V1.0	初稿	---
2025/07/29	V1.0.9	新增：1.增加鉴权状态回调（3.40）。2.增加键盘按下与松开事件发送接口（1.34/2.21）。3.相机麦克风订阅接口(2.25)。4.相机麦克风被抢占通知（3.41）。5.拉流过程中真机键盘窗口隐藏（3.42）。6.拉流过程中真机键盘窗口显示（3.43）。5.拉流过程中键盘切换（3.44）。。	---
2025/08/10	V1.1.0	新增：1.拉流控制播放声音（2.26）。2.下载文件指定存储目录（1.21.1）。。	---

		修改：3.全局初始化增加sdk日志存储目录指定（0.0。4.sdk版本号获取（0.1）。5.设置sdk日志目录（0.3）。6.拉流过程中切换云机键盘接口名变动（2.22.）。7.拉流过程中键盘为真机键盘的场景发送文本或字符、键盘码，接口名变动（2.23）。8.拉流过程中键盘为真机键盘的场景清楚输入框最后一个字符。该功能可以使用sendKeyCode或sendKey发删除键的键盘码一样.接口名变动（2.24）。。	
2025/08/28	V1.1.1	接口变动：1.设置画质（2.15）增加参数控制开启关闭自动调节码率	---
2025/12/04	V1.1.2	接口变动：1.设置画质（2.15）接口名变更为setVideoQuality。	---
2026/01/04	V2.0.4d_20260104	新增：2.27 渲染模式设置	---

简介

PC音视频SDK，开发语言C/C++，主要包括头文件，lib库dll和Demo示例。32/64位release。dll主要实现对云机画面拉取以及云机的一系列操作，Demo简单的实现使用sdk开发云手机流程样例。

 include	2026/2/2 15:50	文件夹	
 win32	2026/2/2 15:50	文件夹	
 x64	2026/2/2 15:50	文件夹	
 云手机PC音视频SDK文档.md	2026/2/2 15:51	Markdown 源文件	57 KB
 云手机PC音视频SDK文档.pdf	2026/2/2 15:48	Microsoft Edge ...	2,105 KB
 CloudPhoneDef.h	2026/2/2 11:48	C Header 源文件	
 CloudPhoneInterface.h	2026/2/2 12:03	C Header 源文件	
 CloudPhoneInterface_c.h	2026/2/2 15:00	C Header 源文件	

 CloudPhoneToolbox.lib	2026/2/2 15:03	Object File Library	89 KB
 CloudPhoneToolbox64.dll	2026/2/2 15:43	应用程序扩展	2,214 KB
 glew32.dll	2024/12/24 16:23	应用程序扩展	454 KB
 libx265.dll	2024/11/11 19:08	应用程序扩展	1,377 KB
 mediastream.dll	2026/2/2 15:21	应用程序扩展	18,185 KB
 SDL2.dll	2025/7/12 17:24	应用程序扩展	1,624 KB

接口开发语言

1.1 C 接口头文件CloudPhoneInterface_c.h

1.2 C++接口头文件CloudPhoneInterface.h, CloudPhoneDef.h

云机相关信息及列表等

1. 根据后台服务地址,AK,SIGN参数,使用API `/api/v1/instance/list` 获取解析设备列表数据。
2. 请求云机设备控制token,使用API `/api/v1/instance/control/token/get` 。
3. 更多云机信息相关的请查阅由后台提供的openapi文档

控制云机

1. 控制云机, 需要传入参数云机设备ID, 后台服务地址, 内部IP和控制token。
2. 例如启动应用app, 则需要传入对应app的包名, 不同的控制功能需要的参数不同。
3. 更多云机操控查看目录 《云机操作部分》

云机拉流

1. 调用SDK API, startPullStream开启拉流
2. 鼠标控制云机点击滑动已sdk内部实现。如有特殊需求api也提供了上层注入。
3. 键盘发送由上层处理。sdk有对应接口发送键盘key。
4. disconnectStream停止拉流。
5. 更多云机拉流功能查看目录 《云机拉流部分》

运行环境

可运行于 win7 sp1 以上 版本 。

SDK导入



 复制代码

1. 包含头文件:

```
include\CloudPhoneDef.h
include\CloudPhoneInterface.h
```

2. 导入lib库

```
CloudPhoneToolbox.lib
```

3. 将dll与exe放在同一目录

```
CloudPhoneToolbox.dll
mediastream.dll
glew32.dll
libx265.dll
SDL2.dll
```

API接口类CloudPhoneInterface

0. 全局初始化



复制代码

```
static void init(const char* folderPath);
```

参数名	必选	类型	说明
folderPath	必须	const char*	只需要在调用sdk其它api前调用一次，folderPath 为sdk日志路径目录 如：D:/log/ 不支持中文和特殊字符路径

返回值	类型	说明
无	无	无

0.1 sdk版本号获取



复制代码

```
const char* getSdkVersion(void);
```

参数名	必选	类型	说明
无	无	无	

返回值	类型	说明
本版字符串	const char*	无

0.2 sdk获取本机设备id字符串

▽	复制代码
<pre>const char* getDeviceId(void);</pre>	

参数名	必选	类型	说明
无	无	无	

返回值	类型	说明
设备id字符串	const char*	无

0.3 设置sdk日志目录

void setLogFolderPath(const char* folderPath);

参数名	必选	类型	说明
folderPath	必须	const char*	folderPath 为sdk日志路径目录 如： D:/log/ 不支持中文和特殊字符路径

返回值	类型	说明
无	无	无

一. 云机操作部分

1.1. 启动应用

▼

复制代码

```
void startApp(const char* instanceId,
             const char* host,
             const char* internalIp,
             const char* token,
             const char*packageName);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token,来源:/api/v2/instance/control/token/get,字段:token
packageName	是	const char*	启动应用包名

返回值	类型	说明
无	无	无

1.2. 停止应用

▼

复制代码

```
void stopApp(const char* instanceId,
             const char* host,
             const char* internalIp,
             const char* token,
```

```
const char*packageName);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token,来源:/api/v2/instance/control/token/get,字段:token
packageName	是	const char*	启动应用包名

返回值	类型	说明
无	无	无

1.3. 卸载应用

[复制代码](#)

```
void unInstallApp(const char* instanceId,  
    const char* host,  
    const char* internalIp,  
    const char* token,  
    const char*packageName);
```

参数名	必选	类型	说明
-----	----	----	----

internallp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
rootSwitch	是	int	0关闭 1开启

返回值	类型	说明
无	无	无

1.8. 开启预览图

▼

复制代码

```
void sysPreviewStart(const char* instanceId,
    const char* host,
    const char* internalIp,
    const char* token,
    const char*interval,
    const char*width,
    const char*height,
    const char*quality);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip

token	是	const char*	鉴权token,来源:/api/v2/instance/control/token/get,字段:token
interval	是	const char*	每秒图片数量, 推荐值:1
width	是	const char*	预览图宽
height	是	const char*	预览图高
quality	是	const char*	预览图压缩质量 1-100 值越大质量越好

返回值	类型	说明
无	无	无

1.9. 关闭预览图

▼

复制代码

```
void sysPreviewStop(const char* instanceId,
    const char* host,
    const char* internalIp,
    const char* token);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip

1.12. 设置卡分辨率

[复制代码](#)

```
void setCardResolution(const char* instanceId,
    const char* host,
    const char* internalIp,
    const char* token,
    int width,
    int height);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
width	是	int	宽
height	是	int	高

返回值	类型	说明
无	无	无

1.13. 设置卡dpi

[复制代码](#)

```
void setCardDpi(const char* instanceId,
```

```
const char* host,
const char* internalIp,
const char* token,
int dpi);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
dpi	是	int	dpi

返回值	类型	说明
无	无	无

1.14. 应用清理



复制代码


```
bool cleanAppData(const char *instanceId, const char *host, const char
*internalIp, const char *token, const char *pkgName, int timeout = 5);
```

参数名	必选	类型	说明
-----	----	----	----

instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
pkgName	是	const char*	app包名
timeout	是	int	超时

返回值	类型	说明
无	无	无

1.15. 应用启用

 复制代码

```
bool enabelApp(const char* instanceId,
    const char* host,
    const char* internalIp,
    const char* token,
    const char* packageName, int timeout = 5);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId

host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
pkgName	是	const char*	app包名
timeout	是	int	超时

返回值	类型	说明
无	无	无

1.16. 应用停用



复制代码

```
bool disabelApp(const char* instanceId,
    const char* host,
    const char* internalIp,
    const char* token,
    const char* packageName, int timeout = 5);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip,来源:/api/v2/instance/list,字段:publicIp

internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
pkgName	是	const char*	app包名
timeout	是	int	超时

返回值	类型	说明
无	无	无

1.17. 设备新机



复制代码

```
bool setSysProp(const char *instanceId, const char *host, const char
*internalIp, const char *token, const char* brand, const char* model,
int timeout = 5);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/

			control/token/get,字段:token
brand	是	const char*	品牌
model	是	const char*	型号
timeout	是	int	超时

返回值	类型	说明
无	无	无

1.18. 消息透传



复制代码

```
bool appMessageTransmission(const char* instanceId, const char* host,
const char* internalIp, const char* token, const char* package_name,
const char* message);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
package_name	是	const char*	包名

[复制代码](#)

```
bool uploadFile(int64_t taskId, const char* filePath, int uploadType,
const char* ip, const char* host, const char* auth);
```

参数名	必选	类型	说明
taskId	是	int64_t	任务id
filePath	是	const char*	文件路径
host	是	const char*	外网ip,来源:/api/v2/instance/list,字段:publicIp
ip	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
auth	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
uploadType	是	int	1: 上传到相册, 2: 上传到SD卡

返回值	类型	说明
无	无	无

1.21. 下载文件

[复制代码](#)

```
bool downloadFile(int64_t taskId, const char* filePath, const char* ip,
const char* host, const char* auth);
```

参数名	必选	类型	说明
taskId	是	int64_t	任务id

host	是	const char*	外网ip,来源:/api/v2/instance/list,字段:publicIp
ip	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
auth	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
enableRename	是	int	重命名传1 不重命名0
rename	是	const char*	文件名utf8字符串或"" 空字符串

返回值	类型	说明
无	无	无

1.22. 吹一吹



复制代码

```
bool blow(const char* instanceId, const char* host, const char*
internalIp, const char* token, int time = 5);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp

internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/instanceId,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
time	否	int	时间 (5-10)

返回值	类型	说明
bool	布尔	无

1.23. 摇一摇



复制代码

```
bool shake(const char* instanceId, const char* host, const char* internalIp, const char* token, int time = 5);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/instanceId, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/instanceId,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/instanceId,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token

time	否	int	时间 (5-10)
------	---	-----	-----------

返回值	类型	说明
bool	布尔	无

1.24. 获取屏幕横竖屏状态

▼

复制代码

```
bool getScreenOrientation(const char *instanceId, const char *host,
const char *internalIp, const char *token);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token

返回值	类型	说明
bool	布尔	无

1.25. 单例同步操作发送键盘值

[复制代码](#)

```
bool sendKeyForGroupControl(const char *instanceId, const char *host,
const char *internalIp, const char *token, int keycode);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
keycode	是	int	键盘值

返回值	类型	说明
bool	布尔	无

1.26. 单例同步操作发送鼠标事件

[复制代码](#)

```
bool sendMouseForGroupControl(const char *instanceId, const char *host,
const char *internalIp, const char *token, int msg, int x, int y, int
width, int height, int hScroll, int vScroll);
```

参数名	必选	类型	说明
-----	----	----	----

instancetype	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instancetype
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
msg	是	int	鼠标事件类型, WM_LBUTTONDOWN, WM_LBUTTONUP, WM_MOUSEMOVE, WM_MOUSEWHEEL
x	是	int	坐标x
y	是	int	坐标y
width	是	int	窗口宽
height	是	int	窗口高
hScroll	是	int	水平方向滚动
vScroll	是	int	垂直方向滚动

返回值	类型	说明
bool	布尔	无

1.27. 设置系统导航栏显示和隐藏

[复制代码](#)

```
bool setNavBarVisable(const char *instanceId, const char *host, const char *internalIp, const char *token, bool visable);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
visable	是	int	显示/隐藏

返回值	类型	说明
bool	布尔	无

1.28. 未拉流切换输入法

[复制代码](#)

```
bool setInputMethod(const char* instanceId,const char* host,const char* internalIp,const char* token, int type);
```

参数名	必选	类型	说明
-----	----	----	----

internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/ist,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
text	是	const char*	文本数据
position	是	int	-1: 删除, 0: 增加

返回值	类型	说明
bool	布尔	无

1.30. 获取系统导航栏状态

 复制代码

```
bool getNavBarStatus(const char *instanceId, const char *host, const char *internalIp, const char *token);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id
host	是	const char*	ws外网ip
internalIp	是	const char*	实例内网ip
token	是	const char*	鉴权token

返回值	类型	说明
bool	布尔	无

1.31. 获取分辨率

[复制代码](#)

```
bool getResolution(const char *instanceId, const char *host, const char *internalIp, const char *token);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id
host	是	const char*	ws外网ip
internalIp	是	const char*	实例内网ip
token	是	const char*	鉴权token

返回值	类型	说明
bool	布尔	无

1.32. 消息订阅

[复制代码](#)

```
bool subscribeTopic(const char* instanceId, const char* host, const char* internalIp, const char* token, const char* topic);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip,来源:/api/v2/instance/list,字段:ip

token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token
topic	是	const char*	主题, 剪切板: clipboard_transmission, 消息透传: message_transmission

返回值	类型	说明

1.33. 取消订阅



复制代码

```
bool unsubscribeTopic(const char* instanceId, const char* host, const char* internalIp, const char* token, const char* topic);
```

参数名	必选	类型	说明
instanceId	是	const char*	实例id ,来源:/api/v2/instance/list, 字段:instanceId
host	是	const char*	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char*	实例内网ip ,来源:/api/v2/instance/list,字段:ip
token	是	const char*	鉴权token ,来源:/api/v2/instance/control/token/get,字段:token

action	是	int	0为按下事件、1为抬起事件
--------	---	-----	---------------

返回值	类型	说明
bool	布尔	无

二. 云机拉流部分

2.1. 拉取云机音视频流

复制代码

```
int startPullStream(const char* uri,
    const iceAddressInfo* iceInfo,
    int iceCount,
    const char* instanceId,
    int fmt,
    int videoWidth,
    int videoHeight,
    int fps,
    int bitrate,
    int cardWidth,
    int cardHeight,
    int cardDensity,
    const char* authToken,
    HWND hwnd,
    const char *host,
    const char *internalIp,
    const char *ctrlToken
);
```

参数名	必选	类型	说明
uri	是	const char*	拉流信令地址，来源:/api/v2/instance/stream/token/get,字段:signalTcp
iceInfo	是	const iceAddressInfo*	获取拉流token协议返回, 来源: /api/v2/instanc

			e/stream/token/get ， 字段： coturnHttp,再用 api/v2/net/forward 请求，得到ip和port, 然后用冒号连接起来
iceCount	是	int	iceInfo 个数
instanceId	是	const char*	实例id ,来 源:/api/v2/instance/ id, 字段:instanceId
fmt	是	int	拉流格式 1 h264 5 H265
videoWidth	是	int	视频宽 云手机场景 推荐720
videoHeight	是	int	视频高 云手机场景 推荐1280
fps	是	int	拉流帧率 30-60
cardWidth	是	int	云机分辨率宽 无需 改动传0
cardHeight	是	int	云机分辨率高 无需 改动传0
cardDensity	是	int	云机分dpi 无需改动 传0
authToken	是	const char*	鉴权码 协议请求返 回的token， 来 源： /api/v2/instanc e/stream/token/get ， 字段： token
hwnd	是	HWND	渲染窗口 窗口句 柄， 来源： 本地 windows界面窗口句 柄

host	是	const char *	ws外网ip ,来源:/api/v2/instance/list,字段:publicIp
internalIp	是	const char *	实例内网ip,来源:/api/v2/instance/list,字段:ip
ctrlToken	是	const char *	控制token, 来源:/api/v2/instance/control/token/get,字段:token

iceAddressInfo 类型说明

参数名	必选	类型	说明
address	是	const char*	地址
type	是	int	0移动 1联通 2电信

返回值	类型	说明
0或小于0	int	0成功 小于0失败

2.2. 断开拉流



复制代码

```
void disconnectStream();
```

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
无	无	无

2.3. 发送键盘key

[复制代码](#)

```
void sendKey(int keyCode);
```

参数名	必选	类型	说明
keyCode	是	int	键盘转换成安卓key值

返回值	类型	说明
无	无	无

2.4. 设置最大拉流码率上限

[复制代码](#)

```
void setMaxBitrate(int bitrate);
```

参数名	必选	类型	说明
bitrate	是	int	设置最大拉流码率上限码率值，用于控制清晰度。可以通过此接口控制各个清晰的码率上限值

返回值	类型	说明
无	无	无

2.5. 暂停拉流

[复制代码](#)

```
void pauseStream();
```

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
无	无	无

2.6. 恢复拉流

▼	复制代码
<pre>void resumeStream();</pre>	

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
无	无	无

2.7. 拉流过程中设置卡分辨率及dpi

▼	复制代码
<pre>void setCardSize(int cardWidth, int cardHeight, int cardDensity);</pre>	

参数名	必选	类型	说明
cardWidth	是	int	云机分辨率宽
cardHeight	是	int	云机分辨率高
cardDensity	是	int	云机dpi

返回值	类型	说明
无	无	无

2.8. 拉流过程中改变拉流视频宽高

▼

复制代码

```
void setVideoSize(int videoWidth, int videoHeight, int fps);
```

参数名	必选	类型	说明
videoWidth	是	int	视频宽
videoHeight	是	int	视频高
fps	是	int	帧率

返回值	类型	说明
无	无	无

2.9. 拉流过程获取拉流状态

▼

复制代码

```
int getMediaConnectionStatus();
```

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
由回调接口返回	无	无

2.10. 音频开

[复制代码](#)

```
bool audioOpen();
```

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
无	无	无

2.11. 音频关

[复制代码](#)

```
bool audioClose();
```

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
无	无	无

2.12. 暂停云机

[复制代码](#)

```
bool pauseStream();
```

参数名	必选	类型	说明

[复制代码](#)

```
bool sendMouse(int msg, int x, int y, int width, int height,
int hScroll, int vScroll);
```

参数名	必选	类型	说明
msg	是	int	WM_LBUTTONDOWN WN/WM_LBUTTON UP/WM_MOUSEM OVE/WM_MOUSE WHEEL
x	是	int	鼠标在当前视频窗口 中的x
y	是	int	鼠标在当前视频窗口 中的y
width	是	int	当前视频窗口的 width
height	是	int	当前视频窗口的 height
hScroll	是	int	鼠标滚轮水平方向 值，这里一般为0
vScroll	是	int	鼠标滚轮垂直方向 值，监听windows消 息得到

返回值	类型	说明
true/false	bool	成功true,失败false

2.18. 获取视频流信息

[复制代码](#)

```
bool getStreamStats(bool enable);
```


y	是	int	y 事件在渲染窗口的x坐标.
w	是	int	w 渲染窗口的渲染区域宽.
h	是	int	h 渲染窗口的渲染区域高.
id	是	int	id 每个手指或按键映射的id值, 由上层自定义值依次递增, id最小值1.下层用于识别同一按键或手指的action事件。

返回值	类型	说明
bool	布尔	无

2.21. 拉流过程发送安卓键盘码到云机

▼

复制代码

```
bool sendKeyCode(int keyCode, int action);
```

参数名	必选	类型	说明
keyCode	是	int	安卓键盘码.
action	是	int	0为按下事件、1为抬起事件

返回值	类型	说明
bool	布尔	无

2.22. 拉流过程中切换云机键盘

[复制代码](#)

```
bool switchKeyboardPassthrough(int action);
```

参数名	必选	类型	说明
action	是	int	1: 真机键盘, 2: 云机键盘

返回值	类型	说明
bool	布尔	无

2.23. 拉流过程中键盘为真机键盘的场景发送文本或字符、键盘码

[复制代码](#)

```
bool sendKeyboardPassthroughValue(const char* value, int keyCode, int action);
```

参数名	必选	类型	说明
value	是	const char*	文本数据
keyCode	是	int	安卓键盘码. 只发文本时该值传0, 如果value是字母数字等、而且传了该值的键盘码, 云机会判断是否打开输入框, 打开输入框就填充value, 如果没有打开输入框就注入keyCode。客户端如果使用该方法注入键盘码需要action成对
action	是	int	0为按下事件、1为抬起事件

返回值	类型	说明
bool	布尔	无

2.24. 拉流过程中键盘为真机键盘的场景清楚输入框最后一个字符。该功能可以使用 `sendKeyCode` 或 `sendKey` 发删除键的键盘码一样

▼复制代码

```
bool clearKeyboardPassthroughValue();
```

参数名	必选	类型	说明
无	无	无	无

返回值	类型	说明
bool	布尔	无

2.25. 拉流过程中订阅相机麦克风，多路拉流时获得相机麦克风使用权

▼复制代码

```
bool subscribeVirtualMicrophoneCamera(bool microphone,
bool camera,
const char* userData);
```

参数名	必选	类型	说明
microphone	是	目前支持传true	
camera	是	目前支持传true	
userData	const char*	用户自定义数据，会通过 virtualMicrophoneCameraOccupiedNotify 相机麦克风占用通	

	知给所有人包括自己，自己用userData区分是否是自己	
--	------------------------------	--

返回值	类型	说明
bool	布尔	无

2.26. 拉流控制播放声音，拉流前设置拉流成功默认不播放声音，拉流过程中设置播放或停止播放声音

▼

复制代码

```
void setAudioPlayout(bool playout);
```

参数名	必选	类型	说明
playout	是	true 播放 false 不播放	

返回值	类型	说明
无	无	无

2.27. 设置渲染缩放模式，默认是铺满渲染

▼

复制代码

```
void setScale(bool sacle);
```

参数名	必选	类型	说明
sacle	布尔	true 等比缩放 false 铺满渲染	

返回值	类型	说明
无	无	无

三. 监听回调部分 监听CloudPhoneObserver

3.1. 启动应用返回

[复制代码](#)

```
virtual void startAppResponse(const char* instanceId, int code, const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.2. 停止应用返回

[复制代码](#)

```
virtual void stopAppResponse(const char* instanceId, int code, const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.3. 卸载应用返回

[复制代码](#)

```
virtual void unInstallAppResponse(const char* instanceId, int code, const char* message) {}
```


[复制代码](#)

```
virtual void rootAppResponse(const char* instanceId, int code, const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.7. 系统root返回

[复制代码](#)

```
virtual void systemRootResponse(const char* instanceId, int code, const char* message) {}
```

3.8. 开启预览图返回

[复制代码](#)

```
virtual void sysPreviewStartResponse(const char* instanceId, int code, const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.9. 停止预览图返回

[复制代码](#)

```
virtual void sysPreviewStopResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.10. 设置虚拟定位返回

[复制代码](#)

```
virtual void virtualLocationResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.11. 发送剪切板返回

[复制代码](#)

```
virtual void clipBoardResponse(const char* instanceId, int code, const
char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.12. 设置系统分辨率返回

▽	复制代码
<pre>virtual void sysResolutionResponse(const char* instanceId, int code, const char* message) {}</pre>	

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.13. 设置系统dpi返回

▽	复制代码
<pre>virtual void sysDpiResponse(const char* instanceId, int code, const char* message) {}</pre>	

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.14. 预览图数据返回

[复制代码](#)

```
virtual void previewResponse(const char* instanceId, int code, const char* message,
    const char* pImag, int size) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	
pImag	const char*	jpg 图片内存数据	
size	int	长度	

3.15. 获取拉流状态返回

[复制代码](#)

```
virtual void onStreamStats(int videoWidth, int videoHeight, float lostRate,
    int decodeFps, int preReceive, int currentRoundTripTime) {}
```

返回值	类型	说明	
videoWidth	int	视频宽	
videoHeight	int	视频高	
lostRate	float	丢包率 百分比	
decodeFps	int	解码帧率	
preReceive	int	每秒字节数 除1024 为KB/s	
currentRoundTripTime	int	网络往返时延 ms	

3.16. 拉流连接状态通知

[复制代码](#)

```
virtual void onStreamConnectionStatus(int code, const char*
descriptions) {}
```

返回值	类型	说明	
code	int	状态值	
descriptions	const char*	描述	

3.17. 拉流视频宽高改变通知

[复制代码](#)

```
virtual void onFrameResultionChanned(int width, int height) {}
```

返回值	类型	说明	
width	int	视频宽	
height	int	视频高	

3.18. 暂停云机通知

[复制代码](#)

```
virtual void pauseStreamResponse(bool ret) {}
```

返回值	类型	说明	
ret	bool	操作结果	

3.19. 恢复云机通知

[复制代码](#)

```
virtual void resumeStreamResponse(bool ret) {}
```

返回值	类型	说明	
ret	bool	操作结果	

3.20. 音频开通知

[复制代码](#)

```
virtual void setAudioOpenResponse(bool ret) {}
```

返回值	类型	说明	
ret	bool	操作结果	

3.21. 音频关通知

[复制代码](#)

```
virtual void setAudioCloseResponse(bool ret) {}
```

返回值	类型	说明	
ret	是	bool	操作结果

3.22. 画质设置通知

[复制代码](#)

```
virtual void setPictureQualityResponse(bool ret) {}
```

返回值	类型	说明	
ret	是	bool	操作结果

3.23. 编码类型设置通知

▼

复制代码

```
virtual void setVideoCodecHwResponse(bool ret) {}
```

返回值	类型	说明	
ret	bool	操作结果	

3.24. 横竖屏设置通知

▼

复制代码

```
virtual void setScreenOrientationResponse(bool ret) {}
```

返回值	类型	说明	
ret	是	bool	操作结果

3.25. 订阅消息通知

▼

复制代码

```
virtual void subscribeTopicResponse(const char* instanceId, int code, const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.26. 取消订阅消息通知

[复制代码](#)

```
virtual void unsubscribeTopicResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.27. 消息透传通知

[复制代码](#)

```
virtual void appMessageTransmissionResponse(const char* instanceId, int
code,
const char* message) {}
```

返回值	类型	说明	
instanceId	是	const char*	实例id
code	是	int	0成 非0 失败
message	是	const char*	描述

3.28. 订阅app消息通知

[复制代码](#)

```
virtual void subscribeAppMessageResponse(const char* instanceId, int
code,
const char* message, const char* pkgName) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	

code	int	0成 非0 失败	
message	const char*	描述	
pkgName	const char*	包名	

3.29. 截图通知

<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>
<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>

返回值	类型	说明	
instanceId	是	const char*	实例id
code	是	int	0成 非0 失败
message	是	const char*	描述
imagePath	是	const char*	文件路径

3.30. 文件上传通知

<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>
<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>

返回值	类型	说明	
taskId	int64_t	任务id	
filename	const char*	文件名称	
type	int	0: 相册, 1: SD卡	
progress	int	进度值	

3.31. 文件下载通知

[复制代码](#)

```
virtual void downloadResponse(int64_t taskId, const char* filename,
int type, int progress) {}
```

返回值	类型	说明	
taskId	int64_t	任务id	
filename	const char*	文件名称	
progress	int	进度值	

3.32. 应用清理通知

[复制代码](#)

```
virtual void cleanAppDataResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.33. 设备新机通知

[复制代码](#)

```
virtual void setSysPropResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	是	const char*	实例id
code	是	int	0成 非0 失败
message	是	const char*	描述

3.34. 超时通知



复制代码

```
virtual void onTimeOutResponse(const char* instanceId, int code,
const char* message) {};
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.35. 应用停用通知



复制代码

```
virtual void disableAppResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	是	const char*	实例id
code	是	int	0成 非0 失败
message	是	const char*	描述

3.36. 应用启用通知

[复制代码](#)

```
virtual void disableAppResponse(const char* instanceId, int code,
const char* message) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	

3.37. 获取导航栏状态通知

[复制代码](#)

```
virtual void onGetNavBarStatusResponse(const char* instanceId, int code,
const char* message, int status) {}
```

返回值	类型	说明	
instanceId	const char*	实例id	
code	int	0成 非0 失败	
message	const char*	描述	
status	int	0: 显示, 1:隐藏	

3.38. 获取分辨率通知

[复制代码](#)

```
virtual void onGetResolutionResponse(const char* instanceId, int code,
const char* message, int width, int height) {}
```

返回值	类型	说明
instanceId	const char*	实例id

code	int	0成 非0 失败
message	const char*	描述
width	int	分辨率宽
height	int	分辨率高

3.39. 触控鼠标事件回调


[复制代码](#)

```
virtual void onMouseChanned(int type, int x, int y, int w, int h,
int hScroll, int vScroll) {}
```

返回值	类型	说明
msg	int	WM_LBUTTONDOWN/WMLBUTTONDOWN/WMLBUTTONUP/WMLMOUSEMOVE/WMLMOUSEWHEEL
x	int	鼠标在当前视频窗口中的x
y	int	鼠标在当前视频窗口中的y
width	int	当前视频窗口的width
height	int	当前视频窗口的height
hScroll	int	鼠标滚轮水平方向值，这里一般为0
vScroll	int	鼠标滚轮垂直方向值，监听windows消息得到

3.40. 拉流鉴权通知回调


[复制代码](#)

```
virtual void onAuthResult(int code, const char* descriptions) {}
```

返回值	类型	说明
code	int	0成 非0 失败
descriptions	const char*	描述

3.41. 相机麦克风被抢占通知

<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>
<pre>virtual void virtualMicrophoneCameraOccupiedNotify(bool microphone, bool camera, const char* userData) {}</pre>	

返回值	类型	说明
microphone	布尔	目前支持传true
camera	布尔	目前支持传true
userData	const char*	用户自定义数据，会通过 virtualMicrophoneCameraOccupiedNotify 相机麦克风占用通知给所有人包括自己，自己用userData区分是否是自己

3.42. 拉流过程中真机键盘窗口隐藏通知

<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>
<pre>virtual void onRealKeyboardWindowHidden() {}</pre>	

返回值	类型	说明
-----	----	----

3.43. 拉流过程中真机键盘窗口显示通知

<div> <div> </div> <div> </div> </div>	<div> <div> </div> <div> </div> </div>
<pre>virtual void onRealKeyboardWindowShow(float y) {}</pre>	

返回值	类型	说明
y	float	键盘弹出的Y坐标

3.44. 拉流过程中真机键盘切换通知

▼复制代码

```
virtual void onRealKeyboardStaus(int action) {}
```

返回值	类型	说明
action	int	1真机键盘打开，2云机键盘打开

群控支持两种模式

1. 模式一：由上层实现遍历每个从控云机通过上述的云机部分的接口注入鼠标事件与及键盘事件实现群控。

1.1.1 *sendKeyForGroupControl* 发送键盘key 按下抬起一起触发

1.1.2 *sendKeyCodeForGroupControl* 发送键盘key 按下抬起上层控制

1.1.3 *sendMouseForGroupControl* 发送鼠标

1.1.4 *sendMultiTouchForGroupControl* 发送鼠标，支持多点触控

优势：接入灵活由上层控制

劣势：群控效率低，群控数量可能受限，上几百台云机上层分发效率低可能导致同步不同步。

2. 模式二：由使用后台api创建群控会话，设置主控和从控，主控的所有控制操作由服务转发给所有从控实现群控同步

2.1.1 后台openapi创建群控会话相关api

▼ 同屏推流

创建同屏推流会话

同屏推流会话续期

注销同屏推流会话

主屏控制权切换

从控设备切换

从控设备生效选定

从控设备生效选定详情

查看同屏推流会话详情

查看同屏推流会话列表

2.1.2 在拉完云机列表后，通过创建群控会话将所有云机创建一个群控会话，把从控选定为0.

2.1.3 在用户需要群控时，将用户选中的云机，调用从控生效选定即可。

2.1.4 上层不需要分发，操作主控主控的鼠标键盘事件会被服务器转发给从控。

2.1.5 主控从控等切换操作请查阅后台openapi接口文档。

优势：同步效率高，从控数量无限制。

3. 与其他厂商做混合群控处理方案

3.1 选择群控模式一简单便捷，但需要考虑劣势选是否满足贵司产品需求。

3.2 选群控模式二与其他厂商做混合群控，逻辑稍微复杂，下面简单梳理一个流程图：

