

rtcSDK文档 V2.0.0 (SDK.min.js)

rtcSDK文档 V2.0.0 (SDK.min.js)

RTC h5 SDK 主要功能是为 h5 赋予拉取云桌面、操控云机，相机穿透能力，可以通过 SDK 连接云机，完成对云机的一系列操作。

一、快速开始

1.1 介绍

RTC web/h5 SDK 主要功能是为 web/h5 赋予拉取云桌面、操控云机，可以通过 SDK 连接云机，完成对云机的一系列操作。

1.2.配置

1.3.接入/初始化

1.4.获取设备唯一标识和SDK版本号

二、基本功能

2.1.MEDIASDK 初始化云机参数

返回参数说明

2.2.开始连接

2.3.主动断开连接

2.4.开始/暂停 连接

参数说明

2.5.打开/关闭 音频

参数说明

2.6.返回主页home，按键返回键，菜单键，音量+-

参数说明

2.7.设置真机/云机键盘

参数说明

2.8.修改视频格式

2.9.修改编码方式

2.10.修改帧率

2.11.设置控制模式

参数说明

2.12.文本消息透传

参数说明

2.13.键盘文本输入

参数说明

2.14.键盘文本删除

参数说明

2.15.屏蔽键盘透传功能

2.16.设置画质

- 注意:

参数说明

2.17.修改卡的分辨率 (物理分辨率)

- 说明:

2.18.修改编码器的分辨率 (推流分辨率)

- 说明:

2.19.获取截图

回调说明

2.20.文件下载

2.21.文件上传

2.22.设备新机

2.23.启用应用

2.24.停用应用

2.25.应用启动

2.26.应用停止

2.27.应用卸载

- 2.28.应用隐藏
- 2.29.取消应用隐藏
- 2.30.应用root控制
- 2.31.全局应用root
- 2.32.虚拟定位（只能使用WGS84坐标）
- 2.33.剪切板
- 2.34.群控设置分辨率
- 2.35.设置DPI
- 2.36.开启预览图推流
- 2.37.停止预览图推流（重点！退出时必须自动调用停止，不然会被占用）
- 2.38.开启文本透传键盘
- 2.39.文本透传-文本输入 V1
- 2.40.文本透传-文本删除 V1
- 2.41.关闭文本透传连接
- 2.42.自定义码率
- 2.43.开启消息订阅
 - 参数说明
- 2.44.关闭消息订阅
 - 参数说明
- 2.45.查询云机横竖屏方向（未推流之前）
 - 参数说明
- 2.46.显示云机导航栏
- 2.47.隐藏云机导航栏
- 2.48.文本透传-文本输入 V2
- 2.49.文本透传-文本删除 V2
- 2.50.查询云机导航栏 显示/隐藏状态
- 2.51.直连发送鼠标事件给群控手机
 - 参数说明
 - 说明:
- 2.52.吹一吹
 - 参数说明
- 2.53.摇一摇
 - 参数说明
- 2.54.是否允许申请摄像头
 - 参数说明
- 2.55.是否允许申请麦克风
 - 参数说明
- 2.56.查询云机分辨率信息

三、回调接口

- 备注
- 3.1.连接成功回调
 - 参数说明
- 3.2.连接异常回调
 - 参数说明
- 3.2.1.连接关闭回调
 - 参数说明
- 3.3.键盘透传响应
 - 参数说明
- 3.4.打开/关闭 音频 响应
 - 参数说明
- 3.5.开启/暂停 推流 响应
 - 参数说明
- 3.6.文本透传 响应
 - 参数说明
- 3.7.设置画质 响应
 - 参数说明

- 3.8.文件下载 响应
 - 参数说明
- 3.9.文件下载进度条 响应
 - 参数说明
- 3.10.文件上传 响应
 - 参数说明
- 3.11.文件上传进度条 响应
 - 参数说明
- 3.12.通道信息 响应
- 3.13.显示区域大小发生改变 响应
 - 参数说明
- 3.14.网络连接统计信息 响应
 - 参数说明
- 3.15.首帧解码时间 响应
 - 参数说明
- 3.15.自定义码率接口回调
 - 参数说明

RTC h5 SDK 主要功能是为 h5 赋予拉取云桌面、操控云机，相机穿透能力，可以通过 SDK 连接云机，完成对云机的一系列操作。

一、快速开始

1.1 介绍

RTC web/h5 SDK 主要功能是为 web/h5 赋予拉取云桌面、操控云机，可以通过 SDK 连接云机，完成对云机的一系列操作。

1.2.配置

```
// html引入 SDK.min.js
<script src="SDK.min.js" type="text/javascript"> </script>

// 设置容器
<div class="wrapper" ref="videoRef" id="videoRef"></div>
```

1.3.接入/初始化

```
// 引入实例
const MediaSdk = window.rtc_sdk.MediaSdk;
const MEDIASDK = new MediaSdk();
```

1.4.获取设备唯一标识和SDK版本号

```
// 获取设备唯一标识
MEDIASDK.deviceUUID()

// 获取SDK版本号
MEDIASDK.getVersion()
```

二、基本功能

2.1.MEDIASDK 初始化云机参数

```
//配置初始化参数
const videoRef = document.getElementById("videoRef");
let option ={
    mount: videoRef, // 挂载节点, id|类名|element 必填
    displaySize: {
        width: '', // 容器节点宽度
        height: '', // 容器节点高度
    },
    topic: '' // 实例SN号 必填
    url: '', //信令服务地址 必填
    ICEServerUrl: IceMap, // 三网地址 必填
    forwardServerAddress: '', // 转发服务器地址
    ip: '', // 实例ip
    controlToken: '', // 代理控制token
    width: '', // 推流视频宽度 必填
    height: '', // 推流视频高度 必填
    cardwidth: '', // 云机系统分辨率 宽 必填 （默认填0不处理）
    cardHeight: '', // 云机系统分辨率 高 必填 （默认填0不处理）
    cardDensity: '', // 云机系统显示 密度 必填 （默认填0不处理）
    authToken: '', //拉流鉴权 token 必填
    quality:'', //画质 必填
    fps:'', //帧率 必填
    videoCodec:'', // 编码方式 必填 'H264 High' || 'VP8'
    videoCodecMethod:'', // 视频格式 硬编true | 软编false
    isMuted: false, // 是否开启静音 默认false
    isAllowedOpenCamera: false, // 是否允许开启摄像头权限 必填
    sendFollow: true, // 是否允许主控转发文本到实例
    callback: statusCallback, // 回调函数 必填
}
```

返回参数说明

参数名	必选	类型	说明
mount	是	string	挂载节点
displaySize	是	Objectber	容器视频宽高 width: null, height: null
topic	是	string	云机ID
url	是	string	信令服务地址

参数名	必选	类型	说明
ICEServerUrl	是	Arr	三网地址 [{ "CMNET": \${res.data.ip}:\${res.data.port} }, { 'CHINANET-GD': \${res.data.ip}:\${res.data.port} }, { 'UNICOM-GD': \${res.data.ip}:\${res.data.port} }]
forwardServerAddress	是	string	转发服务器地址 列表接口 wss://sslForwardServer[0]?.publicIp ws://forwardServer[0]?.publicIp
ip	是	string	实例ip
controlToken	是	string	代理控制token api/v2/instance/control/token/get
width	是	number	推流视频宽度
height	是	number	推流视频高度
cardWidth	是	number	云机系统分辨率 宽
cardHeight	是	number	云机系统分辨率 高
cardDensity	是	number	云机系统显示 密度
authToken	是	number	拉流鉴权 token
quality	是	string	画质 超清/高清/标清/流畅
fps	是	number	帧率
videoCodec	是	string	编码方式
videoCodecMethod	是	string	视频格式
isMuted	是	BOOL	是否开启静音
isAllowedOpenCamera	否	BOOL	是否允许开启摄像头权限
sendFollow	是	BOOL	是否允许主控转发文本到实例
callback	是	string	回调函数

2.2.开始连接

```
// 开始推流
MEDIASDK.RtcEngine(option)
```

2.3.主动断开连接

```
// 断开函数
MEDIASDK.disconnect();
```

2.4.开始/暂停 连接

```
// 暂停接收流信息
MEDIASDK.setControlStream(data);
```

参数说明

字段	必填	类型	说明
data	是	BOOL	true: 恢复接收流信息 false: 暂停接收流信息

2.5.打开/关闭 音频

```
// 默认推流自动打开音频
MEDIASDK.setControlAudio(data);
```

参数说明

字段	必填	类型	说明
data	是	BOOL	true: 打开音频 false: 关闭音频

2.6.返回主页home， 按键返回键， 菜单键， 音量+-

```
// 187:虚拟按键菜单键,3:虚拟按键home键,4:虚拟按键返回键,24:音量增大,25:音量减小

MEDIASDK.sendKey(3);
```

参数说明

字段	必填	类型	说明
data	是	number	187:虚拟按键菜单键,3:虚拟按键home键,4:虚拟按键返回键,24:音量增大,25:音量减小

2.7.设置真机/云机键盘

```
// 示例
MEDIASDK.setCloudKeyboard(data);
```

参数说明

字段	必填	类型	说明
data	是	number	1：真机键盘 2：云机键盘

2.8.修改视频格式

```
/* 备注：修改初始化参数，重连生效，客户端决定是否重连 */
初始化 videoCodecMethod: bool // 硬编true | 软编false
重连 MEDIASDK.create(option)
```

2.9.修改编码方式

```
/* 备注：修改初始化参数，重连生效，客户端决定是否重连 */
初始化 videoCodec: string // 'H264 High' || 'VP8'
重连 MEDIASDK.create(option)
```

2.10.修改帧率

```
/* 备注：修改初始化参数，重连生效，客户端决定是否重连 */
初始化 fps: number // 60, 40, 30, 20
重连 MEDIASDK.create(option)
```

2.11.设置控制模式

```
// 手机端：触控 PC端：键鼠 PC端移动模式：触控
MEDIASDK.changeTouchMode(data);
```

参数说明

字段	必填	类型	说明
data	是	BOOL	true: 触控 false: 键鼠

2.12.文本消息透传

```
// 文本消息
MEDIASDK.setMessageTransmission(data);
```

参数说明

字段	必填	类型	说明
data	是	string	透传文本

2.13.键盘文本输入

```
// 文本消息
MEDIASDK.KeyboardAddText(data);
```

参数说明

字段	必填	类型	说明
data	是	string	input文本输入

2.14.键盘文本删除

```
// 文本消息
MEDIASDK.KeyboardDeleteText(data);
```

参数说明

字段	必填	类型	说明
data	是	string	input文本删除

2.15.屏蔽键盘透传功能

```
// 删除键盘透传功能
MEDIASDK.removeKeyBoard();
```

2.16.设置画质

- 注意:

自动：提高流畅性,降低延迟，画质会跟随网络变化而变化,网络差时会降低画质保流畅度，网络恢复时画质提升。

手动：保证推流过程中画质稳定性，可以有效控制推流流量,网络差时会出现卡顿，丢包，高延迟，声音断续等。

```
/* 示例 */
MEDIASDK.setMaxBitrate(data);
```


参数说明

字段	必填	类型	说明
data	是	string	'自动', '流畅', '标清', '高清', '超清'

2.17.修改卡的分辨率 (物理分辨率)

- 说明:

该接口只能在拉流中的时候才能修改卡的分辨率

```
const data = {
  'width': 1080,
  'height': 1920,
  'density': 480
}
MEDIASDK.setCardSize(data);
```

2.18.修改编码器的分辨率 (推流分辨率)

- 说明:

该接口只能在拉流中的时候才能修改卡的分辨率

```
// 备注: fps 修改的上限为 调用start时传入的fps值。
const data = {
  'width': 1080,
  'height': 1920,
  'fps': 30,
}
MEDIASDK.setEncoderSize(data);
```

2.19.获取截图

```
const data ={
  instanceId: String, //板卡实例id
  url:String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备唯一标识
  quality: number, // 图片质量
  listener: OperatorListener(event) // 回调函数
}
MEDIASDK.sysScreenshot(data);
```

回调说明

字段	必填	类型	说明
OperatorListener	是	Obj	event.content.data : 图片地址（云机图片路径）

2.20.文件下载

```
const data = {  
  
  ip: String, //板卡ip  
  token: String, //ws 连接鉴权token  
  filePath: string, // 文件路径  
  httpPath: string, // http请求地址 http://888.888.888  
  
}  
MEDIASDK.fileDownload(data);
```

2.21.文件上传

```
// 1: 相册 2: SD卡  
const data = {  
  
  ip: String, //板卡ip  
  token: String, //ws 连接鉴权token  
  file: file, // 选中文件  
  uploadType: string, // 1 相册 2 sdcard  
  httpPath: string, // http请求地址 http://888.888.888  
  
}  
MEDIASDK.fileupload(data);
```

2.22.设备新机

```
const data = {  
  instanceId: String, //板卡实例id  
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912  
  ip: String, //板卡ip  
  token: String, //ws 连接鉴权token  
  deviceId: string, // 设备ID  
  brand: String, // 手机品牌  
  model: String, // 手机型号  
  listener: OperatorListener  
}  
MEDIASDK.sysNewDevice(data);
```

2.23.启用应用

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  packageName: String, // 应用包名
  listener: OperatorListener
}
MEDIASDK.enableApp(data);
```

2.24.停用应用

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  packageName: String, // 应用包名
  listener: OperatorListener
}
MEDIASDK.disableApp(data);
```

2.25.应用启动

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  packageName: String, // 应用包名
  listener: OperatorListener
}
MEDIASDK.startApp(data);
```

2.26.应用停止

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  packageName: String,
  listener: OperatorListener
}
MEDIASDK.stopApp(data);
```

2.27.应用卸载

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  packageName: String,
  listener: OperatorListener
}
MEDIASDK.unInstallApp(data);
```

2.28.应用隐藏

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  listener: OperatorListener
}
MEDIASDK.hideApp(data);
```

2.29.取消应用隐藏

```
const data = {
  instanceId: String, //板卡实例id
  url: String, //ws连接地址, 格式: ws://123.138.156.4:44912
  ip: String, //板卡ip
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  listener: OperatorListener
}
MEDIASDK.displayApp(data);
```

2.30.应用root控制

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  rootSwitch: Int, //root开关 0禁用, 1启用
  packageList: List<String>, //应用包名集合
  listener: OperatorListener
}
MEDIASDK.rootApp(data);
```

2.31.全局应用root

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  rootSwitch: Int, //root开关 0禁用, 1启用
  listener: OperatorListener
}
MEDIASDK.systemRoot(data);
```

2.32.虚拟定位（只能使用WGS84坐标）

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  longitude: String,
  latitude: String,
  listener: OperatorListener
}
MEDIASDK.virtualLocation(data);
```

2.33.剪切板

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  deviceId: string, // 设备ID
  token: String, //ws 连接鉴权token
  content: String, //需要剪切的内容 最大长度5000
  listener: OperatorListener
}
MEDIASDK.clipBoard(data);
```

2.34.群控设置分辨率

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  width: String,
  height: String,
  listener: OperatorListener
}
MEDIASDK.sysResolution(data);
```

2.35.设置DPI

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  dpi: String,
  listener: OperatorListener
}
MEDIASDK.sysDpi(data);
```

2.36.开启预览图推流

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  interval: String, //预览图推流间隔时间, 类型为 string。单位为秒
  width: String, // 预览图宽, 类型为 string
  height: String, // 预览图高, 类型为 string
  quality: String, // 预览图质量, 类型为 string值范围: 1-100
  listener: : function (event) {
    switch (event.code) {
      case 0:
        if (event.type === 'sys_preview_stop') {
          // 关闭预览图
        } else if (event.type === 'sys_preview_open') {
          event.url // 返回的预览图二进制流
        }
        break;
      }
    }
  }
}
MEDIASDK.sysPreviewStart(data);
```

2.37.停止预览图推流（重点！退出时必须自动调用停止，不然会被占用）

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  listener: OperatorListener
}
MEDIASDK.sysPreviewStop(data);
```

2.38.开启文本透传键盘

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: string, // 设备ID
  type: int, // 1 真机键盘 2 云机键盘
  listener: OperatorListener
}
MEDIASDK.EnableTextKeyboard(data);
```

2.39.文本透传-文本输入 V1

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: String, // 设备ID
  content: String, // 文本内容
  listener: OperatorListener
}
MEDIASDK.addTextTransmission(data);
```

2.40.文本透传-文本删除 V1

```
const data = {
  instanceId: String,
  url: String, //ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String, //ws 连接鉴权token
  deviceId: String, // 设备ID
  content: String, // ★: 文本删除内容只能为空''
  listener: OperatorListener
}
MEDIASDK.deleteTextTransmission(data);
```


2.41.关闭文本透传连接

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  listener: OperatorListener
}
MEDIASDK.disableTextKeyboard(data);
```

2.42.自定义码率

```
// 自定义码率
MEDIASDK.maxbitrateSetting(data);
```

2.43.开启消息订阅

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  topic:array,// 订阅主题
  listener: OperatorListener
}
MEDIASDK.openSubscribeTopic(data);
```

参数说明

字段	必填	类型	说明
topic	是	array	'message_transmission':消息透传, 'clipboard_transmission': 剪贴板透传, 'screen_orientation':云机横竖屏消息订阅, 'sys_resolution': 分辨率订阅

2.44.关闭消息订阅

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  topic:string,// 订阅主题
  listener: OperatorListener
}
MEDIASDK.unsubscribeTopic(data);
```

参数说明

字段	必填	类型	说明
topic	是	string	'message_transmission':消息透传

2.45.查询云机横竖屏方向（未推流之前）

```
const data ={
  instanceId: String,
  url:String,
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: string, // 设备ID
  listener: OperatorListener
}
MEDIASDK.sysGetScreenOrientation(data);
```

参数说明

字段	必填	类型	说明
orientation	是	int	0竖屏, 1横屏

2.46.显示云机导航栏

```
const data ={
  instanceId: String,
  url:String,
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: string, // 设备ID
  listener: OperatorListener
}
MEDIASDK.sysDisplayBar(data);
```

2.47.隐藏云机导航栏

```
const data ={
  instanceId: String,
  url:String,
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: string, // 设备ID
  listener: OperatorListener
}
MEDIASDK.sysHideBar(data);
```

2.48.文本透传-文本输入 V2

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  content: String, // 文本内容
  listener: OperatorListener
}
MEDIASDK.addInputTransmission(data);
```

2.49.文本透传-文本删除 V2

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  content: String, // ★: 文本删除内容只能为空''
  listener: OperatorListener
}
MEDIASDK.deleteInputTransmission(data);
```

2.50.查询云机导航栏 显示/隐藏状态

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  listener: OperatorListener
}
MEDIASDK.sysGetBarStatus(data);
```

2.51.直连发送鼠标事件给群控手机

```
MEDIASDK.sendMouseForGroupControl(type,x,y,width,height,vScroll);
```

参数说明

字段	必填	类型	说明
type	是	int	2:mousemove鼠标移动/0:mousedown鼠标按下/1:mouseup鼠标抬起/4:mouseleave鼠标离开/3:wheel 鼠标滚动
x	是	int	鼠标在当前视频窗口中的x, 如offsetX
y	是	int	鼠标在当前视频窗口中的y, 如offsetY
width	是	string	当前视频窗口的width, 如clientWidth
height	是	string	当前视频窗口的height, 如clientHeight
vScroll	是	int	0:向上滚动 1:向下滚动

- 说明:

鼠标滚动情况下 X,Y传0

2.52.吹一吹

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  time: int, //
  listener: OperatorListener
}
MEDIASDK.sysBlow(data);
```

参数说明

字段	必填	类型	说明
time	是	int	吹一吹秒数(5-10)

2.53.摇一摇

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  time: int, //
  listener: OperatorListener
}
MEDIASDK.sysShake(data);
```

参数说明

字段	必填	类型	说明
time	是	int	摇一摇秒数(3-10)

2.54.是否允许申请摄像头

```
MEDIASDK.SetupCameraLive(data);
```

参数说明

字段	必填	类型	说明
data	是	BOOL	true: 允许 false: 不允许

2.55.是否允许申请麦克风

```
MEDIASDK.SetupMicrophoneLive(data);
```

参数说明

字段	必填	类型	说明
data	是	BOOL	true: 允许 false: 不允许

2.56.查询云机分辨率信息

```
const data ={
  instanceId: String,
  url:String,//ws连接地址 格式: ws://123.138.156.4:44912
  ip: String,
  token: String,//ws 连接鉴权token
  deviceId: String, // 设备ID
  listener: OperatorListener
}
MEDIASDK.sysGetResolution(data);
```

三、回调接口

备注

3.1.连接成功回调

```
/* 示例 */
MEDIASDK.on(CONNECT_SUCCESS, (r) => {
  // 连接成功,响应
  console.log("Player => CONNECT_SUCCESS", r);
});
```

参数说明

type类型	state描述	code
nats_connect_status	success: nats服务连接成功	code: 10010
rtc_connect_status	success: rtc连接成功	code: 1001
Authentication_status	success : 鉴权成功	code: 1005

3.2.连接异常回调

```
/* 示例 */
MEDIASDK.on(CONNECT_ERROR, (r) => {
  // 连接异常,响应
  console.log("Player => CONNECT_ERROR", r);
});
```

参数说明

| type类型 | state描述 | code
|:---- |:---
| nats_connect_status | TIMEOUT: nats服务连接超时 | code: 10011 |
| nats_connect_status | failed: nats服务失败 | code: 10012 |
| rtc_connect_status | failed: rtc连接失败 | code: 1002 |
| rtc_connect_status | disconnected: rtc异常断开 | code: 1003 |

|rtc_connect_status | closed: rtc连接失败 | code: 1004 |
|Authentication_status | failed : 鉴权失败 | code: 1006

3.2.1.连接关闭回调

```
/* 示例 */
MEDIASDK.on(CONNECT_CLOSE, (r) => {
  // 连接成功,响应
  console.log("Player => CONNECT_CLOSE", r);
});
```

参数说明

type类型	state描述	code
close_Peerconnection	success: 关闭RTC通道成功	code: 10014
REMOVE_NODEEL	success: 删除容器节点	code: 10015

3.3.键盘透传响应

```
/* 示例 */
MEDIASDK.on(keyboardFeedbackBean, (r) => {
  // 接收到的键盘透传响应

  console.log("Player => keyboardFeedbackBean", r);
});
```

参数说明

type类型	state描述	code
keyboardFeedbackBean	open: 真机键盘打开	code: 10040
keyboardFeedbackBean	close: 真机键盘关闭	code: 10041

3.4.打开/关闭 音频 响应

```
/* 示例 */
MEDIASDK.on(RECEIVE_AUDIO_STATUS, (r) => {
  // 打开/关闭 音频 响应

  console.log("Player => RECEIVE_AUDIO_STATUS", r);
});
```

参数说明

type类型	state描述	code
audio_Status	success: 打开/关闭 音频成功	code: 10023
audio_Status	failed: 打开/关闭 音频失败	code: 10024

3.5.开启/暂停 推流 响应

```
/* 示例 */
MEDIASDK.on(RECEIVE_STREAM_STATUS, (r) => {
    // 开启/暂停 推流 响应

    console.log("Player => RECEIVE_STREAM_STATUS", r);
});
```

参数说明

type类型	state描述	code
stream_Status	success: 开启/暂停 推流成功	code: 10021
stream_Status	failed: 开启/暂停 推流失败	code: 10022

3.6.文本透传 响应

```
/* 示例 */
MEDIASDK.on(RECEIVE_MESSAGE_TRANSMISSION_STATUS, (r) => {
    // 文本透传 响应

    console.log("Player => RECEIVE_MESSAGE_TRANSMISSION_STATUS", r);
});
```

参数说明

type类型	state描述	code
MessageTransmission_Status	success: 文本透传成功	code: 10027
MessageTransmission_Status	failed: 文本透传失败	code: 10028

3.7.设置画质 响应

```
/* 示例 */
MEDIASDK.on(RECEIVE_BITRATE_STATUS, (r) => {
    // 文本透传 响应

    console.log("Player => RECEIVE_BITRATE_STATUS", r);
});
```

参数说明

type类型	state描述	code
bitrate_Status	success: 画质设置成功	code: 10025
bitrate_Status	failed: 画质设置失败	code: 10026

3.8.文件下载 响应

```
/* 示例 */
MEDIASDK.on(FILEDOWNMLOAD_STATUS, (r) => {
    // 文件下载 响应

    console.log("Player => FILEDOWNMLOAD_STATUS", r);
});
```

参数说明

type类型	state描述	code
fileDonwload_Status	start: 文件开始 下载	code: 10030
fileDonwload_Status	success: 文件下载 成功	code: 10031
fileDonwload_Status	failed: 文件下载 失败	code: 10032

3.9.文件下载进度条 响应

```
/* 示例 */
MEDIASDK.on(FILEDOWNMLOAD_PROGRESS, (r) => {
    // 文件下载进度条 响应

    console.log("Player => FILEDOWNMLOAD_PROGRESS", r);
});
```

参数说明

type类型	返回值	描述
fileDownload_Progress	0-100	进度条数据

3.10.文件上传 响应

```
/* 示例 */
MEDIASDK.on(FILEUPLOAD_STATUS, (r) => {
    // 文件上传 响应

    console.log("Player => FILEUPLOAD_STATUS", r);
});
```

参数说明

type类型	state描述	code
fileUpload_Status	start: 文件开始 上传	code: 10035
fileUpload_Status	success: 文件上传 成功	code: 10036
fileUpload_Status	failed: 文件上传 失败	code: 10037

3.11.文件上传进度条 响应

```
/* 示例 */
MEDIASDK.on(FILEUPLOAD_PROGRESS, (r) => {
    // 文件下载进度条 响应

    console.log("Player => FILEUPLOAD_PROGRESS", r);
});
```

参数说明

type类型	返回值	描述
fileUpload_Progress	0-100	进度条数据

3.12.通道信息 响应

```
/* 示例 */
MEDIASDK.on(RECEIVE_CHANNEL, (r) => {
    // 接收到 CHANNEL通道,响应
    console.log("Control => RECEIVE_CHANNEL", r);
});
```

3.13.显示区域大小发生改变 响应

```
/* 示例 */
MEDIASDK.on(RECEIVE_RESOLUTION, (r) => {
    // 分辨率大小发生改变,响应
    console.log("Player => RECEIVE_RESOLUTION", r);
});
```

参数说明

返回值说明
width: 编码器分辨率宽度
height: 编码器分辨率高度

3.14.网络连接统计信息 响应

```
/* 示例 */
MEDIASDK.on(NETWORK_STATS, (r) => {
    // 网络连接统计信息,响应 1s/1

});
```

参数说明

属性	返回值描述
averageDecodeTime	解码时间
framesPerSecond	帧率
lostRate	丢包率
secondsBytes	带宽 (b)
secondsKBytes	带宽 (kb)
secondsMBytes	带宽 (Mb)
currentRoundTripTime	网络延迟

3.15.首帧解码时间 响应

```
/* 示例 */
MEDIASDK.on("DECODING_STATUS", (r) => {
    // 首帧解码时间
});
```

参数说明

type类型	state描述	code
FirstDecoding_status	time: rtc连接成功时间	code: 10071
FirstDecoding_status	time: 鉴权成功时间	code: 10073
FirstDecoding_status	time: 首帧解码时间	code: 1007
FirstDecoding_status	time: 信令连接成功时间	code: 10074
FirstDecoding_status	time: 获取清晰度配置成功时间	code: 10077
FirstDecoding_status	time: 获取清晰度配置失败时间	code: 10078
FirstDecoding_status	time: 获取清晰度配置超时时间	code: 10080

3.15.自定义码率接口回调

```
/* 示例 */
MEDIASDK.on('MAXRECEIVE_BITRATE_STATUS', (r) => {
  if (r.code === 10053) {
    // 不支持自定义码率修改
  } else if (r.code === 10054) {
    // 超出可设置最大码率，最大码率只支持xxx
  } else if (r.code === 10051) {
    // 自定义码率设置成功
  }
}
```

参数说明

type类型	state描述	code
maxbitrate_Status	success: 自定义码率设置成功	code: 10051
maxbitrate_Status	failed: 不支持自定义码率修改	code: 10053
maxbitrate_Status	failed: 超出可设置最大码率，最大码率只支持xxx	code: 10054