

产品概述

云平台概述

- 云屋云平台覆盖全球200多个国家和地区 and 国内几十家运营商网络，可支持百万人同时在线，平台采用集群机制，99.99%高可用。

私有化部署

- 可以在用户自有网络进行整套系统的私有化部署，适用于金融、军队、政府等安全性要求较高的场景。

技术优势

- 高保真、高还原度的语音效果，支持智能回声消除、自动增益和噪声抑制。
- 视频传输支持动态网络流控和丢包补偿，在低带宽的较差网络下也能保障视频的清晰流畅。
- 平台稳定可靠，支持大并发下的高可用性。

支持的平台

- 支持iOS、Android、Web、Windows、MacOS、Linux、小程序全平台互通，支持Chrome、Safari、360、QQ等主流浏览器，适配了5000多款不同的移动终端。详情见下表：

| 平台 | 支持版本 | |
|---------|-----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Android | 支持Android4.2及以上版本，有armv7a架构 | |
| iOS | 支持iOS8.0及以上系统版本，有armv7、arm64架构 | |
| Windows | 支持Windows xp sp2及以上版本 Windows sdk提供com接口，几乎支持windows上的任何开发语言，如：c/c++(mfc、Qt)，C#，VB，Delphi等 | |
| 微信小程序 | 微信6.6及以上 | |
| Web | Web插件 | 支持IE8、IE9、IE10、IE11，Chrome44及以下 |
| | H5 | <p>1.Windows：</p> <p>Chrome：Chrome68+</p> <p>edge：80+</p> <p>Firefox：Firefox78+</p> <p>Opera：支持最新版本</p> <p>360浏览器(极速模式): 支持最新版本</p> <p>QQ浏览器(极速模式): 支持最新版本</p> <p>其他浏览器：使用chrome内核版本68+的其他浏览器</p> <p>2.macOS 10.13+：</p> <p>Chrome：Chrome58+</p> <p>Safari: Safari11+</p> <p>3.iOS 11+：Safari11+</p> <p>3.iOS 14.3+：</p> <p>Safari浏览器</p> <p>微信内置浏览器（微信6.5+）</p> <p>其它使用WKWebView的浏览器</p> <p>4.Android 4.4+：</p> <p>Android system webview 58+</p> <p>使用chrome58+内核的安卓浏览器，如：</p> <p>Chrome：Chrome58+</p> <p>微信内置浏览器（最新版）</p> <p>QQ内置浏览器（最新版）</p> <p>QQ浏览器（最新版）</p> <p>华为浏览器（chrome内核58+）</p> <p>edge浏览器（最新版）</p> <p>宙斯浏览器（最新版）</p> |

主要功能

| | |
|------|-----------------------------|
| 音视频 | 多方同时音视频交互；在互联网即可体验会议室级的逼真效果 |
| 屏幕共享 | 屏幕、应用、浏览器标签共享 |
| | |

| | |
|------|------------------------------------------------|
| 影音共享 | 本地音视频文件远程与本地同步观看，典型场景如：远程教学中的视频案例分析、金融业务中风险播报等 |
| 文字聊天 | 可在房间内进行实时文字消息互动，既可选择房间内全部用户，也可向特定用户发起私聊 |
| 录制 | 可在房间内任意时间进行云端录制 |
| 直播 | 支持单点直播，多方互动直播，影音文件共享直播 |
| 白板 | 白板涂鸦 |

关键词

- SDK管理平台：**
 是云屋提供给SDK注册用户的 [管理平台](#) 。可以进行帐户管理、项目管理、录像管理、队列管理、通信记录查询、用量监控等操作。
[点此注册](#)  一个账号，或者联系商务代为开通，或在网站咨询客服。
- AppID：**
 用于区分不同的项目。每个项目都有属于自己唯一的AppID，不同AppID的项目完全独立，无法相互通信。
 可以通过 [管理平台](#)  中的“项目管理”来创建AppID和维护相关配置。
- AppSecret：**
 AppID对应的密码，可登录云屋管理后台修改。
- Token：**
 云屋提供的token生成器生成的授权令牌。它具有动态性、时效性，可以替换appSecret用于登录鉴权。
- 房间：**
 平台当前提供的音视频、白板、屏幕共享、IM群聊服务都是基于房间的，在使用这些服务之前，必须要先创建房间，只有加入到同一个房间的用户才能够使用这些业务互相通信。房间创建后如果不主动销毁将会长期有效。
- 房间号：**
 房间的唯一标识，用户需要先调用创建房间的api，然后在创建成功的回调通知中获取该房间的信息结构体。房间号来自该结构体。
- UID：**
 用户ID，在登录和加入房间时传入，用于标识不同的用户。同一个项目下的UID需要保证唯一。
- 发布视频流：**
 用户加入房间后，可以向房间内的其他用户发送本地采集的音视频数据流，也就是发布视频流。
- 订阅视频流：**
 用户加入房间后，可以选择接收房间内其他用户发布的音视频数据流，也就是订阅视频流。
- 云端录制：**
 在服务器上对房间内的音视频、白板、屏幕共享等通讯内容进行录制，支持自定义录制内容和布局，录制文件在服务器保存，可以通过API下载到本地。
- 透明通道：**
 用于多个客户端之间传递用户的自定义内容，支持大块数据和小块数据两种模式，透明通道可以在房间外使用。
- cookie：**

指接口cookie参数，提供给业务层的命令上下文本地缓存机制（cookie不会在网络上传输）。在命令响应回调接口里传给业务层，回调之后cookie数据就会自动销毁。

用法举例：业务层分别向A、B各发一条消息（将目标用户的id存在cookie里），在失败回调接口里可从cookie取回用户id，就能知道发给谁的消息失败了。

登录鉴权

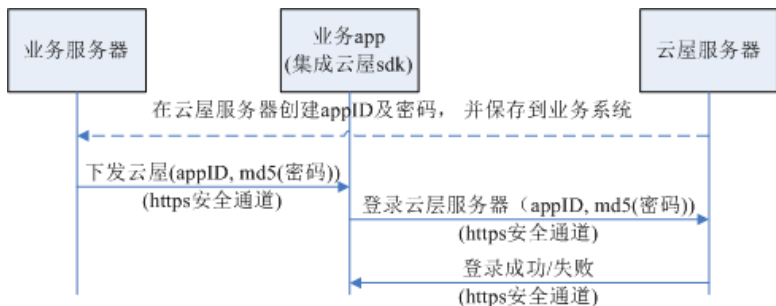
SDK登录鉴权的目的：

确保只有合法的身份才能使用sdk接入到服务器；

SDK提供两种登录鉴权方案：

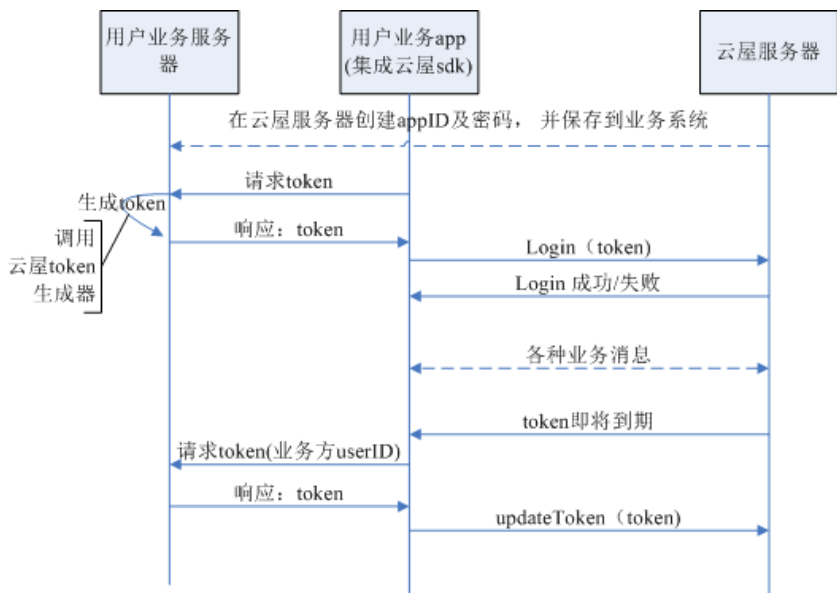
- 1、密码鉴权方案：此方案简单，配合https通信，适用于一般的安全要求；
- 2、动态token鉴权方案：此方案每次登录token都不一样，并且可自定义token的有效期，具有更高的安全性；

• 密码鉴权方案说明



- 1. 在云屋web管理页面上，创建appID及密码，然后保存到业务服务器上；
- 2. 在app连接到业务服务器时，通过安全通道将“appID及md5(密码)”下发给app并配置给sdk；
- 3. sdk通过https向云屋服务器发起登录时携带“appID及md5(密码)”；
- 4. 云屋服务器校验密码是否正确；

• 动态token鉴权方案说明



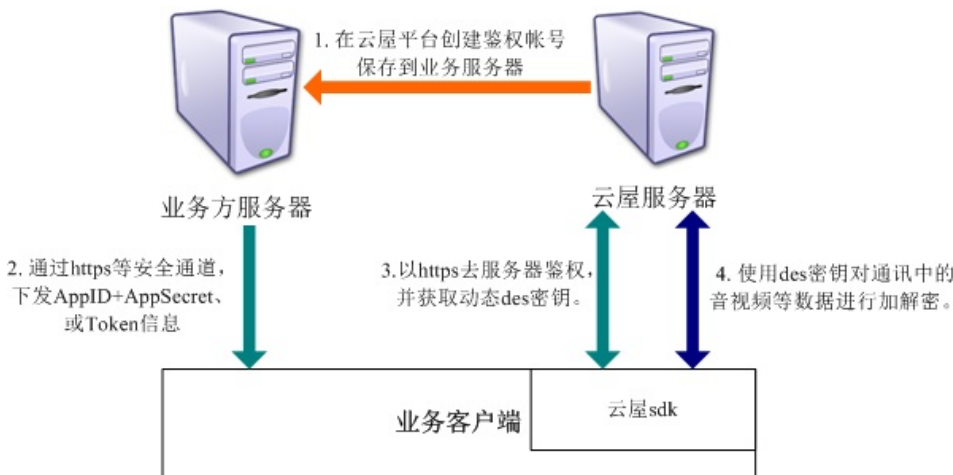
- 1. 在云屋web管理页面上，创建appID及密码，然后保存到业务服务器上；

- 2. App调用sdk登录之前，去业务服务器请求“token”；
- 3. 业务服务器调用 [云屋token生成器](#) ，生成token并返回给App；
- 4. App收到token后，调用sdk登录；
- 5. sdk通过https向云屋服务器发起登录时携带“appId及token”；
- 6. 云屋服务器校验token；
- 7. 在token即将过期前30秒，服务器通知sdk，sdk会通知app；
- 8. app应该尽快从业务服务器获取到新的token并提交给sdk，sdk将其更新到云屋服务器后，就可保证通信不中断；（如果token到期前没有更新有效token，通信将被结束）

信息安全说明

通信过程中数据安全说明：

使用云屋sdk登录鉴权和业务通信描述图：



如上图，以https通道进行鉴权，以及通过https通道获得本次通讯的des密钥，对后续tcp，udp数据进行des加解密处理。只要业务服务器不泄漏鉴权帐号信息，通信就是安全的。

Token机制说明：

云屋的token采用的是jwt这种开放标准（RFC 7519）生成的。

在token的payload中携带appId、授权有效期，并使用appSecret进行SHA256签名操作，在签名过程中会进行加盐(salt)处理。

由于appSecret是保密的，所以外界无法伪造token。再加上token设置了有效期，时间一到token就会失效，所以安全性相比采用appSecret会更高一些。

准备工作

帐号申请

[点此注册](#) 一个账号，或者联系商务代为开通，或在网站咨询客服。

创建项目

可以在 [管理平台](#) 中创建新的项目（系统有一个默认项目，可以直接使用）。项目信息页内可得到此项目的App ID及App Secret。如下图：

项目管理

用量监控

通信记录

录像管理

队列管理

房间信息

账户管理

项目管理（每个账户最多建立10个项目，每个项目分配独立的APP ID。如果需要项目数量超过10个，请联系你的销售经理）

创建新项目

请输入项目名称或APP ID

默认项目

项目名称：默认项目

APP ID：4026859001

APP Secret：*****

项目状态：开启

创建时间：2016-12-15 10:59:58

防火墙开通

在使用云屋SDK提供的相关服务之前，您需要打开下面这些特定的端口：

| 协议 | 端口 | 功能说明 | 适用SDK平台 | | | |
|-----|------|-------------------------|----------------------------|----|-----|------|
| | | | Windows、Android、iOS、web插件版 | H5 | 小程序 | 直播观看 |
| TCP | 2726 | 信令端口1 (SDK缺省使用https) | √ | √ | √ | √ |
| | 2728 | 信令端口2 | √ | | | |
| | 1935 | RTMP端口 | | | √ | √ |
| UDP | 2698 | 音视频流媒体端口 | √ | | | |
| | 2699 | 音视频流媒体端口(H5) | | √ | | |

开发及生产环境限制

- H5音视频SDK需调用浏览器的媒体设备（摄像头、麦克风等），为保护用户隐私，各浏览器会采用不同的安全限制。以谷歌浏览器来说，以下是能互通音视频的几种方案：
 - https + 域名 + SSL域名证书（推荐该方案，适用开发、生产环境）
 - http + 本地服务，localhost 或 127.0.0.1（仅用于本机浏览开发，适用开发环境）
 - https + ip + SSL自签证书（局域网互通方案，会报不安全的提示，适用开发环境）

SDK下载

- [点此下载](#) H5 SDK
- 在项目中引入SDK文件：RTCSDKV2.min.js

```
<script type="text/javascript" src="/SDK/RTCSDKV2.min.js"></script>
```

html

实现音视频通话

- 快速创建并进入房间，开始音视频通话；代码部分均为 js 代码，详细代码请参考 SDK包 目录下 Demo 源代码。
- 请先准备 [准备工作](#) 中的相关内容：1. [下载SDK](#) ， 2. [准备运行环境](#) ，以及连接相关的麦克风摄像头并确认设备工作正常。

基本步骤如下：

1. [初始化SDK](#)
2. [登录连接视频服务器](#)
3. [创建房间](#)
4. [进入房间](#)
5. [打开麦克风/摄像头](#)
6. [观看他人视频](#)
7. [退出房间](#)
8. [注销登录](#)
9. [反初始化，退出SDK](#)

1. 初始化SDK

初始化是整个SDK的使用基础，通常在程序启动的时候进行初始化([CRVideo_Init](#))，退出后进行反初始化([CRVideo_Uninit](#))，整个程序的生命周期中只进行一次初始化和反初始化。

```
// 初始化接口为Promise接口，请注意调用方式如下：  
CRVideo_Init(initObj).then(res => {  
    // 初始化成功的回调，继续设置服务器地址及登录等操作，见 第2步...  
}, err => {  
    // 初始化失败的回调，会返回对应的错误码  
});
```

相关API请参考：

- [CRVideo_Init](#)
- [CRVideo_Uninit](#)

2. 登录连接视频服务器

设置视频服务器地址，然后使用appId和md5加密后的appSecret登录。 ([获取App ID及App Secret](#))

- 调用接口：

```
// 设置服务器地址，使用云屋的云服务时填 sdk.cloudroom.com;使用私有化部署的服务器时要填部署的服务器地址:端口；此处以云屋的云服务为  
CRVideo_SetServerAddr("sdk.cloudroom.com"); // 云屋云服务器  
// CRVideo_SetServerAddr("10.8.8.220:2726"); // 私有化部署服务器  
  
//登录鉴权  
CRVideo_Login(appID, md5(appSecret), nickName, UID, userAuthCode, cookie); // 使用appId+appSecret鉴权  
// CRVideo_UpdateToken(Token, nickName, UID, userAuthCode, cookie); // 使用token鉴权
```

- 回调通知，

```
// 登录成功
CRVideo_LoginSuccess.callback = function(UID,cookie){
    // 登录成功，开始创建房间，见第3步
}

// 登录失败
CRVideo_LoginFail.callback = function(sdkErr,cookie){
    // 登录出错，可以弹出错误提示，或调用登录接口再次重试登录
}

// 登录掉线
CRVideo_LineOff.callback=function(sdkErr){
    // 系统掉线，可尝试重新登录，或弹出提示
    // token失效时，若没有及时更新token，系统将会强制掉线
}
```

相关API请参考：

- [CRVideo_SetServerAddr](#) （ 设置服务器地址 ）
- [CRVideo_Login](#) （ 使用appID登录 ）
- [CRVideo_LoginByToken](#) （ 使用Token登录 ）
- [CRVideo_Logout](#) （ 退出登录 ）
- [CRVideo_UpdateToken](#) （ 更新Token ）
- [CRVideo_LoginSuccess.callback](#) （ 登录成功 ）
- [CRVideo_LoginFail.callback](#) （ 登录失败 ）
- [CRVideo_NotifyTokenWillExpire.callback](#) （ 通知，Token即将失效 ）
- [CRVideo_LineOff.callback](#) （ 通知，登录掉线 ）

3. 创建房间

创建一个没有密码的，固定的房间

- 调用接口：

```
//创建房间
CRVideo_CreateMeeting()
```

- 回调通知，

```
//创建房间成功
CRVideo_CreateMeetingSuccess.callback=function(meetObj,cookie){
    //创建房间成功，可以进入房间
}

//创建房间失败
CRVideo_CreateMeetingFail.callback = function(sdkErr,cookie){
    //创建房间失败，可以弹出错误提示
}
```

相关API请参考：

- [CRVideo_CreateMeeting](#) （ 创建房间 ）
- [CRVideo_CreateMeetingSuccess](#) （ 创建房间成功 ）
- [CRVideo_CreateMeetingFail](#) （ 创建房间失败 ）

4. 进入房间

用创建成功的房间信息（房间ID）进入房间，其他用户也是利用此房间信息进入该房间。

- 调用接口：

```
//进入房间
CRVideo_EnterMeeting3(meetID, cookie);
```

js

- 回调通知，

```
//进入房间完成响应
CRVideo_EnterMeetingRslt.callback = function(sdkErr) {
    if(sdkErr === 0) {
        console.log("进入房间成功，可以继续下面第5步.....");
    } else {
        console.log("进入房间失败，错误码："+ sdkErr);
    }
}
//监控房间掉线
CRVideo_MeetingDropped.callback = function() {
    // 从房间中掉线，需要再次进入房间或退回进入房间前的状态
}
```

js

相关API请参考：

- [CRVideo_EnterMeeting3](#)（进入房间）
- [CRVideo_EnterMeetingRslt.callback](#)（进入房间的结果）
- [CRVideo_MeetingDropped.callback](#)（通知，从房间中掉线了）

5. 打开麦克风/摄像头

接第4步，进入房间成功后，打开自己的麦克风和摄像头，以便本地、远端显示自己的视频图像

```
// 打开房间内某个成员的麦克风（可以是自己）
CRVideo_OpenMic(UID)

// 打开房间内某个成员的摄像头（可以是自己）
CRVideo_OpenVideo(UID)

// 创建成员视频UI显示组件，并订阅某个成员的视频画面（可以是自己）
var myVideoUI = CRVideo_CreatVideoObj(); // 调用接口，创建视频ui组件
document.body.appendChild(myVideoUI.handler()) // 获取组件中的DOM并显示在页面上
myVideoUI.setVideo(UID); // 将用户的视频画面挂载在UI组件上，传入某个成员（自己）的UID
```

js

相关API请参考：

- [CRVideo_OpenMic](#)（打开成员麦克风）
- [CRVideo_OpenVideo](#)（打开成员摄像头）
- [CRVideo_CreatVideoObj](#)（创建成员视频UI显示组件）
- [CRVideo_GetAllVideoInfo](#)（获取成员的所有摄像头信息）
- [CRVideo_SetVideoCfg](#)（设置本地视频参数）
- [CRVideo_SetAudioCfg](#)（设置本地音频参数）

- [CRVideo_CloseVideo](#) （关闭成员摄像头）
- [CRVideo_CloseMic](#) （关闭成员麦克风）
- [CRVideo_SetDefaultVideo](#) （设置默认摄像头）
- [CRVideo_GetDefaultVideo](#) （获取当前的默认摄像头ID）

6. 观看他人视频

成功进入房间后，根据他人登录id，设置并观看他人视频图像

- 接口调用：

```
// 创建成员视频UI显示组件，并订阅某个成员的视频画面
var otherVideoUI = CRVideo_CreatVideoObj(); // 调用接口，创建视频ui组件
document.body.appendChild(otherVideoUI.handler()) // 获取组件中的DOM并显示在页面上
otherVideoUI.setVideo(UID); // 将成员的视频画面挂载在UI组件上，传入他人的UID。UID可通过接口 CRVideo_GetAllMembers 获取
```

相关API请参考：

- [CRVideo_CreatVideoObj](#) （创建成员视频UI显示组件）
- [CRVideo_GetAllMembers](#) （获取房间内所有成员的信息）

7.退出房间

```
// 退出房间
CRVideo_ExitMeeting()

// 销毁房间，房间也可以不销毁，以便一直存在，可以随时再次进入
CRVideo_DestroyMeeting()
```

相关API请参考：

- [CRVideo_ExitMeeting](#) （退出房间）
- [CRVideo_DestroyMeeting](#) （销毁房间）

8. 注销登录

```
// 注销本次登录
CRVideo_Logout()
```

相关API请参考：

- [CRVideo_Logout](#)

9.反初始化，退出SDK

执行反初始化后SDK功能不再可用。

```
// 反初始化
CRVideo_Uninit();
```

相关API请参考：

- [CRVideo_Uninit](#)

设置视频属性

功能简介：

在通话过程中可以根据实际业务场景，调整视频画面的清晰度和流畅度，提升用户体验。视频属性一般包含视频清晰度、分辨率、帧率等。

注意：成功进入房间后，才可以设置视频属性。

- 调用接口：

```
//设置全局的摄像头参数,json数据：CRVideo_VideoCfg
var cfg = {
  size: 2,
  fps: 20,
  ratio: 1
};
CRVideo_SetVideoCfg(cfg);
```

js

相关API请参考：

- [CRVideo_SetVideoCfg](#)
- [CRVideo_GetVideoCfg](#)

相关结构定义请参考：

- [CRVideo_VideoCfg](#)

设置音频属性

功能简介：

在视频通话中，用户还可以设置音频属性，获取不同的用户体验。

注意：成功进入房间后，才可以设置音频属性。

- 调用接口：

```
// 获取系统上的麦克风列表，可用于 CRVideo_SetAudioCfg
var audioMicList = CRVideo_GetAudioMicNames();

// 获取系统上的扬声器列表，可用于CRVideo_SetAudioCfg
var audioSpkList = CRVideo_GetAudioSpkNames();

//设置音频参数
var cfg = {
  micID: audioList[0].micID, // 可通过CRVideo_GetAudioMicNames获取麦克风ID
  speakerID: audioSpkList[0].speakerID // 可通过CRVideo_GetAudioSpkNames接口获取扬声器ID
};
CRVideo_SetAudioCfg(cfg);
```

相关API请参考：

- [CRVideo_GetAudioMicNames](#)
- [CRVideo_GetAudioSpkNames](#)
- [CRVideo_SetAudioCfg](#)
- [CRVideo_GetAudioCfg](#)

相关结构定义请参考：

- [CRVideo_AudioCfg](#)

多方视频通话

功能简介：

多方视频通话时，根据当前业务场景合理设置视频编码参数，可以在较低的带宽占用下实现流畅清晰的音视频沟通，下面将针对几种常见场景进行介绍。

1.一对一

视频通话中，画中画布局指视频图像布局上有重叠。只需控制ui组件的父盒子z-index属性即可



描述： 比如双人视频聊天场景，双方通常都希望看到对方比较清晰的视频，此时可以使用较高的视频编码分辨率，比如720P或480P。

- 示例代码：

```
<!--html代码-->
<div class="videoContainer">
  <div id="otherVideoContainer"></div> <!--底层，即对方的视频控件父div：-->
  <div id="myVideoContainer"></div> <!--上层，即自己的视频控件父div-->
</div>
```

html

```

/* css代码 */
.videoContainer {
    position: relative;
    width: 1280px;
    height: 720px;
}
#otherVideoContainer {
    position: absolute;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    z-index: 1;
}
#myVideoContainer {
    position: absolute;
    right: 0;
    bottom: 0;
    width: 100%;
    height: 100%;
    z-index: 2; /* 画中画需要处理层级关系 */
}

```

```

//js部分：
//创建视频控件对象
var otherVideoUI = CRVideo_CreatVideoObj();
//把创建的控件放入DOM树中
document.getElementById("otherVideoContainer").appendChlid(otherVideoUI.handler());
//otherUID为对方的登录ID
otherVideoUI.setVideo(otherUID);

//创建视频控件对象
var myVideoUI = CRVideo_CreatVideoObj();
//把创建的控件放入DOM树中
document.getElementById("myVideoContainer").appendChlid(myVideoUI.handler());
//myUID为自己的登录ID
myVideoUI.setVideo(myUID);

//3.打开用户摄像头
CRVideo_OpenVideo(myUID)

```

相关API请参考：

- [CRVideo_CreatVideoObj](#)
- [CRVideo_OpenVideo](#)
- [VideoObj.handler\(\)](#)
- [VideoObj.setVideo\(\)](#)

2.多方视频



描述： 比如在线教育场景，老师的视频画面比较大，可以使用较高的分辨率比如720P，下面学生的视频画面比较小，应采用较低的视频编码分辨率，比如360P或256P。

- 示例代码：

```
<!-- html代码 -->
<div id="videoContent">
  <div class="mainVideo"></div>
  <div class="smallVideo"></div>
</div>
```

html

```
/* css代码 */
* {
  margin: 0;
  padding: 0;
}

html,
body {
  height: 100%;
}

body {
  background-color: #0c1d25;
}

#videoContent {
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

#videoContent .mainVideo {
  display: flex;
  width: 1280px;
  height: 720px;
  margin-bottom: 10px;
  background-color: yellow;
}

#videoContent .smallVideo {
  font-size: 0;
  display: flex;
}

#videoContent .smallVideo .student {
  height: 243px;
  width: 432px;
  margin-right: 10px;
  background-color: green;
}

#videoContent .smallVideo .student:last-of-type {
  margin-right: 0;
}
```


//js代码部分：

```
CRVideo_GetAllMembers().forEach(member => {
  const videoObj = CRVideo_CreatVideoObj();
  //假设老师的userID为teacher,需要放入指定的盒子
  if (member.userID === 'teacher') {
    document.getElementsByClassName('mainVideo')[0].appendChild(videoObj.handler());
  } else {
    const div = document.createElement('div');
    div.appendChild(videoObj.handler());
    div.className = 'student';
    document.getElementsByClassName('smallVideo')[0].appendChild(div);
  }
  videoObj.setVideo(member.userID);
});
```

//3.打开自己的摄像头

CRVideo_OpenVideo(UID); //自己的登录ID

相关API请参考：

- [CRVideo_CreatVideoObj](#)
- [CRVideo_GetAllMembers](#)
- [CRVideo_OpenVideo](#)

相关结构定义请参考：

- [CRVideo_MemberInfo](#)

云端录制

功能简介：

在服务器上对房间内的音视频、白板、屏幕共享等通讯内容进行录制，支持自定义录制内容和布局，录制文件在服务器保存，可以通过API下载和删除。其中，下载需要先使用Web API [获取录像文件](#) ，以及Web API实现 [删除录像文件](#) .

1. [开始云端录制](#)
2. [云端录制文件信息变化通知](#)
3. [更新云端录制内容](#)
4. [获取云端录制状态](#)
5. [停止云端录制](#)

1.开始云端录制

- 调用接口：

```
// 云端录制推荐分辨率
var record_size_arr = [
  [640, 360, 400],
```

```

[964, 480, 600],
[1280, 720, 1200]
];

var mixerID = "1"; //生成一个混图器id
var size = 0; //使用第1个分辨率

var rw = record_size_arr[size][0]; //640
var rh = record_size_arr[size][1]; //360
var bitRate = record_size_arr[size][2] * 1000; //400 * 1000

var mw = rw / 2; //每个视频的录制宽度
var mh = mw * 9 / 16; //每个视频的录制高度
var mtop = (rh - mh) / 2; //放在录制画布的相对高度，这里是计算出视频垂直居中的位置

//配置混图器编码参数
var mutiMixerCfgs = [
  {
    "id": mixerID,
    "streamTypes": 3, //录制音视频
    "cfg": {
      "width": rw,
      "height": rh,
      "frameRate": 8,
      "bitRate": bitRate,
      "defaultQP": 26
    }
  }
];

//左视频图像的内容配置
var leftCon = {
  "type": 0; //0：录制类型为视频
  "left": 0;
  "top": mtop; //视频图像垂直居中
  "width": mw;
  "height": mh;
  //camid参数为: user1.1 ,表示该内容需要录制userID是user1的1号摄像头。后方可传入-1，表示为录制默认摄像头
  "param": {"camid": "user1.1"};
  "keepAspectRatio": 1;
};

//右视频图像的内容配置
var rightCon = {
  "type": 0; //0：录制类型为视频
  "left": mw;
  "top": mtop; //视频图像垂直居中
  "width": mw;
  "height": mh;
  "param": {"camid": "user2.1"};
  "keepAspectRatio": 1;
};

//配置混图器内容
var mutiMixerContents = [
  {
    "id": mixerID,
    "content": [ leftCon ,rightCon ]
  }
]

```

```

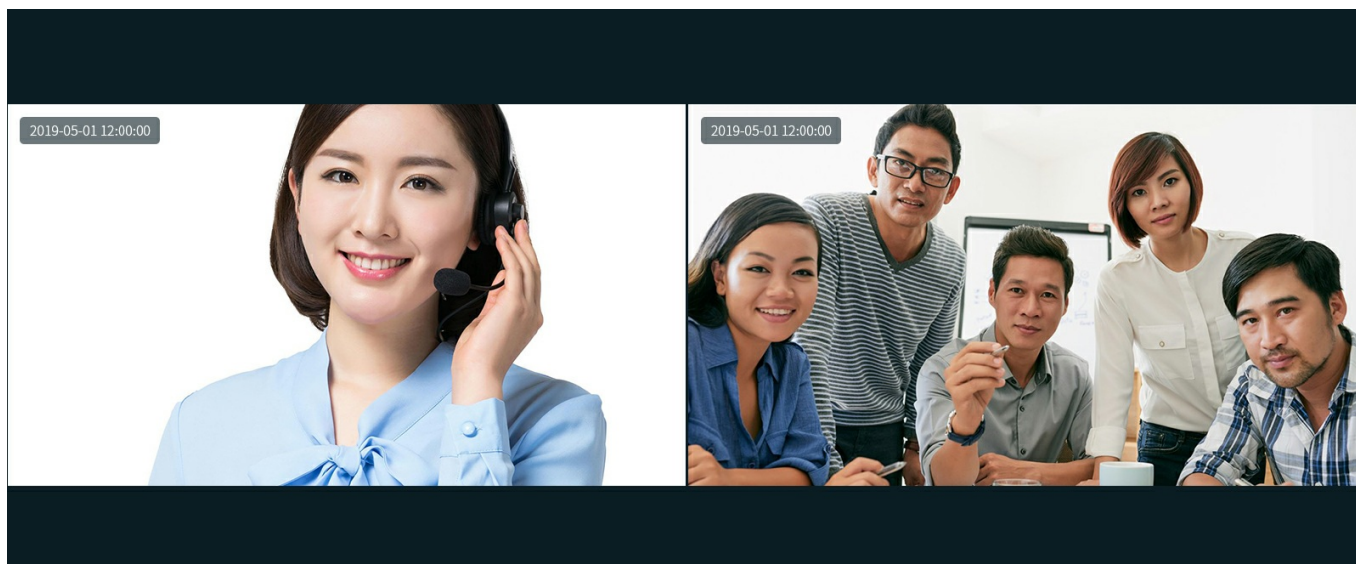
];

//配置混图器输出
var mutiMixerOutputs = [
  {
    "id": mixerID,
    "output": [
      {
        "type": 0,
        "filename": "/2019-03-19/" + "2019-03-19-10-30-30_Web.mp4" //配置录制的文件名
      }
    ]
  }
]
};

//开始录制，云端录制开启后会回调CRVideo_SvrMixerStateChanged接口
CRVideo_StartSvrMixer(mutiMixerCfgs, mutiMixerContents, mutiMixerOutputs);

```

- 左右布局示例图：



- 回调通知：

```

//云端录制状态变化通知
CRVideo_SvrMixerStateChanged.callback = function(state, err, operatorID){
  console.log("CRVideo_SvrMixerStateChanged, state:" + state + ",err:" + err + ",operatorID:" + operatorID);
}

```

js

相关API请参考：

- [CRVideo_StartSvrMixer](#)
- [CRVideo_SvrMixerStateChanged](#)

相关结构定义请参考：

- [CRVideo_MutiMixerCfg](#)
- [CRVideo_MutiMixerContent](#)
- [CRVideo_MutiMixerOutput](#)

2. 云端录制文件信息变化通知

录制过程中都会触发此事件。在此可以实时获得录制状态、录制文件当前的时长、大小，以及录制异常等信息。

- 回调通知：

```
//云端录制文件信息变化通知
CRVideo_SvrMixerOutputInfo.callback = function(outputInfo){
  console.info("CRVideo_SvrMixerOutputInfo",outputInfo);
}
```

js

相关API请参考：

- [CRVideo_SvrMixerOutputInfo](#)

相关结构定义请参考：

- [CRVideo_MixerOutputInfoObj](#)

3.更新云端录制内容

- 接口调用：

```

// 云端录制推荐分辨率
var record_size_arr = [
    [640, 360, 400],
    [964, 480, 600],
    [1280, 720, 1200]
];

var mixerID = "1"; //取之前的混图器id
var size = 0; //使用第1个分辨率

//更新录制布局为：画中画
//混图器内容：画中画布局(示例图如下， 底层640*360， 上层213*120)，底层为用户1的1号摄像头， 上层为用户2的1号摄像头

var bigW = record_size_arr[size][0]; //640， 底层视频的宽度
var bigH = record_size_arr[size][1]; //360， 底层视频的高度

var smallH = bigH / 3; //缩放3倍作为顶层视频的高度
var smallW = smallH * 16 / 9; //视频宽高比为16 : 9， 计算出顶层视频宽度

//底层视频图像的内容配置
var underCon = {
    "type": 0; //0： 录制类型为视频
    "left": 0;
    "top": 0;
    "width": bigW;
    "height": bigH;
    //camid参数为： user1.1 ,表示该内容需要录制userID是user1的1号摄像头。后方可传入-1， 表示为录制默认摄像头
    "param": {"camid": "user1.1"};
    "keepAspectRatio": 1;
};

//顶层视频图像的内容配置
var topCon = {
    "type": 0; //0： 录制类型为视频
    "left": bigW - smallW;
    "top": bigH - smallH;
    "width": smallW;
    "height": smallH;
    "param": {"camid": "user2.1"};
    "keepAspectRatio": 1;
};

//配置混图器内容
var mutiMixerContents = [
    {
        "id": mixerID,
        "content": [ underCon ,topCon ] //数组后面的画面将会覆盖前面的画面
    }
];

//更新录制内容， 录制更新后会通知CRVideo_SvrMixerCfgChanged接口
CRVideo_UpdateSvrMixerContent(mutiMixerContents);

```

- 画中画布局示例图：



相关API请参考:

- [CRVideo_UpdateSvrMixerContent](#)

4.获取云端录制状态

- 接口调用：

```
//获取云端录制状态  
var state = CRVideo_GetSvrMixerState()
```

js

相关API请参考:

- [CRVideo_GetSvrMixerState](#)

相关结构定义请参考：

- [CRVideo_MIXER_STATE](#)

5.停止云端录制

停止云端录制后，也会触发事件 [CRVideo_SvrMixerStateChanged](#)

- 接口调用：

```
CRVideo_StopSvrMixer();
```

js

相关API请参考:

- [CRVideo_StopSvrMixer](#)

屏幕共享

功能简介：

在视频会话中为了提高沟通效率，可以将自己的屏幕内容分享给其他参与方观看。还支持在屏幕上进行标注，以及授权其他参与方进行远程控制。当屏幕共享者开启标注后，控件就进入标注模式。此时，房间内所有用户都可以进行标注，并观看他人标注的内容。

使用场景如下：

- 视频会议场景中，屏幕共享可以将讲话者本地的文件、数据、网页、PPT 等画面分享给其他与会人；
- 在线课堂场景中，屏幕共享可以将老师的课件、笔记、讲课内容等画面展示给学生观看。

注意：同一个房间中，不支持多人同时开启屏幕共享。

共享端：

1. [开始共享](#)
2. [停止共享](#)

观看端：

1. [观看共享的屏幕](#)
2. [屏幕共享停止通知](#)

共享端

1.开始共享

使用接口启动共享，出现共享内容显示组件后，用组件上的功能开始标注和结束共享。

- 接口调用：

```
/*  
*开启屏幕共享分为2步：  
*/  
  
//1.开启屏幕共享  
CRVideo_StartScreenShare();  
  
//2.用户根据浏览器弹窗选择所需要的屏幕窗口共享或者应用程序
```

- 回调通知：

```
// 主动开启屏幕共享的结果  
CRVideo_StartScreenShareRslt.callback = function (sdkErr){  
    if(sdkErr == 0){ //主动开启共享成功  
        console.log("主动开启共享成功");  
    } else {  
        console.log("主动开启共享失败，错误码：" + sdkErr);  
    }  
}
```

相关API请参考：

- [CRVideo_StartScreenShare](#)
- [CRVideo_StartScreenShareRslt](#)

2.停止共享

- 接口调用：

```
//主动停止屏幕共享
CRVideo_StopScreenShare()
```

js

- 回调通知：

```
//主动停止屏幕共享的结果
CRVideo_StopScreenShareRslt.callback = function (sdkErr){
    if(sdkErr == 0){ //主动停止共享成功
        console.log("主动停止共享成功");
    } else {
        console.log("主动停止共享失败，错误码：" + sdkErr);
    }
}
```

js

相关API请参考：

- [CRVideo_StopScreenShare](#)
- [CRVideo_StopScreenShareRslt](#)

1.观看共享的屏幕

- 回调通知：

```
//收到他人开启了屏幕共享的通知，此时可以观看他人的屏幕
CRVideo_NotifyScreenShareStarted.callback = function (sharer) {
    var screenShareObj = CRVideo_CreatScreenShareObj(); //创建屏幕共享显示组件
    document.body.appendChild(screenShareObj.handler()); //添加到页面任意位置
    screenShareObj.setVideo(sharer); //需要将视频挂在在ui组件上

    var nicknameStyle = {
        display: "block"
        left: "10px",
        top: "10px",
        backgroundColor: "rgba(0, 0, 0, .7)",
        padding: "0 10px",
        borderRadius: "15px",
    }

    screenShareObj.setNickNameStyle(nicknameStyle); // 设置昵称样式
    screenShareObj.setNickNameContent(CRVideo_GetMemberNickName(sharer) + "'s screen"); //设置昵称
}
```

js

相关API请参考：

- [CRVideo_NotifyScreenShareStarted](#)

2.屏幕共享停止通知

- 回调通知：

//收到他人停止了屏幕共享的通知，即停止观看（此时可以隐藏屏幕共享控件）

```
CRVideo_NotifyScreenShareStopped.callback = function(){  
    //销毁屏幕共享控件。示例代码：  
    document.body.removeChlid(screenShareObj.handler())  
    screenShareObj.destory();  
    screenShareObj = null;  
}
```

js

相关API请参考：

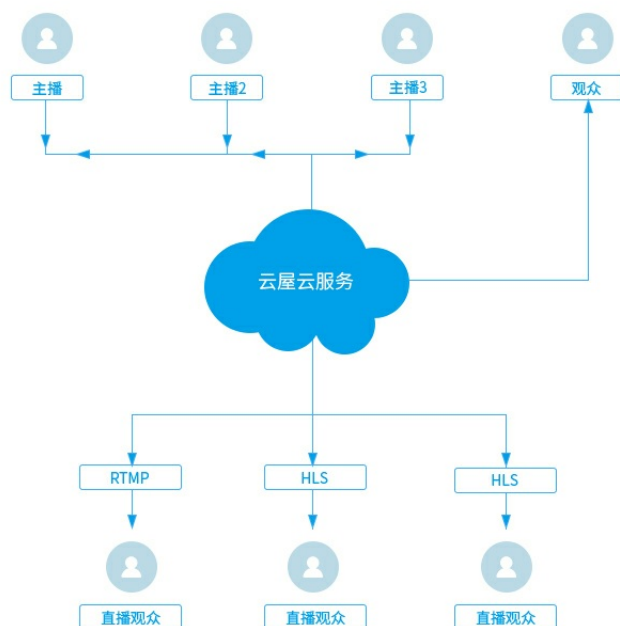
- [CRVideo_NotifyScreenShareStopped](#)

云端直播推流

功能简介：

用于多个主播实时连麦互动。技术实现上，我们会把房间里多个主播的音视频在服务器合成一路流后推流到CDN流媒体服务器，直播观众可以获取RTMP或HLS流观看直播。

云端直播推流架构图：



云端直播推流使用流程如下：

1. [创建直播间并获得推流地址](#)
2. [开始云端直播推流](#)

3. [更新云端直播推流内容](#)
4. [观众观看直播](#)
5. [停止云端直播推流](#)
6. [回放点播](#)

1.创建直播间并获得推流地址

- 创建直播间请参见：Web API [创建直播](#) 。
- 获取直播推流地址请参见：Web API [获取推流地址](#) 。

2.开始云端直播推流

- 调用接口：

```
// 云端推流推荐分辨率
var record_size_arr = [
    [640, 360, 400],
    [964, 480, 600],
    [1280, 720, 1200]
];

var mixerID = "1"; //生成一个混图器id
var size = 0; //使用第1个分辨率

var rw = record_size_arr[size][0]; //640
var rh = record_size_arr[size][1]; //360
var bitRate = record_size_arr[size][2] * 1000; //400 * 1000

var mw = rw / 2; //每个视频的宽度
var mh = mw * 9 / 16; //每个视频的高度
var mtop = (rh - mh) / 2; //放在画布的相对高度，这里是计算出视频垂直居中的位置

//配置混图器编码参数
var mutiMixerCfgs = [
    {
        "id": mixerID,
        "cfg": {
            "width": rw,
            "height": rh,
            "frameRate": 8,
            "bitRate": bitRate,
            "defaultQP": 26
        }
    }
];

//左视频图像的内容配置
var leftCon = {
    "type": 0; //0：录制类型为视频
    "left": 0;
    "top": mtop; //视频图像垂直居中
    "width": mw;
    "height": mh;
    //camid参数为: user1.1 ,表示该内容需要录制userID是user1的1号摄像头。后方可传入-1，表示为录制默认摄像头
```

js

```

    "param": {"camid": "user1.1"};
    "keepAspectRatio": 1;
};
//右视频图像的内容配置
var rightCon = {
    "type": 0; //0 : 录制类型为视频
    "left": mw;
    "top": mtop; //视频图像垂直居中
    "width": mw;
    "height": mh;
    "param": {"camid": "user2.1"};
    "keepAspectRatio": 1;
};

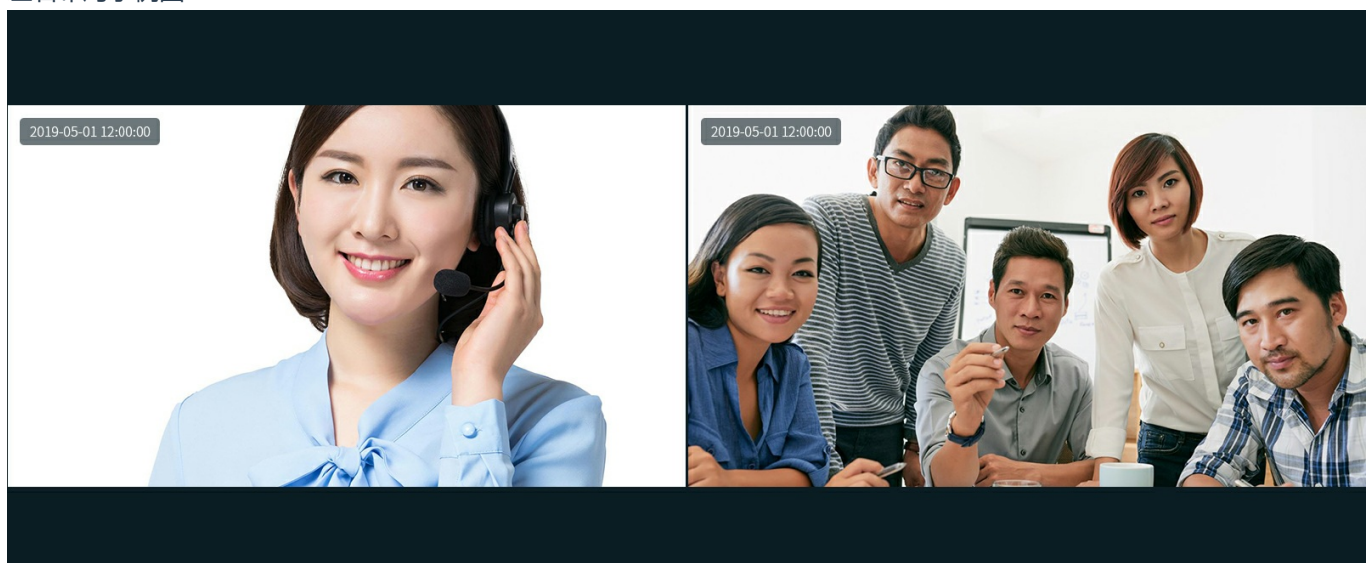
//配置混图器内容
var mutiMixerContents = [
    {
        "id": mixerID,
        "content": [ leftCon ,rightCon ]
    }
];

//配置混图器输出
var mutiMixerOutputs = [
    {
        "id": mixerID,
        "output": [
            {"type": 1, "liveUrl": "rtmp://xxx"} //配置推流地址,地址来自 步骤1 获取的推流地址
        ]
    }
];

//开始直播推流, 云端推流开启后会回调CRVideo_SvrMixerStateChanged接口
CRVideo_StartSvrMixer(mutiMixerCfgs, mutiMixerContents, mutiMixerOutputs);

```

左右布局示例图:



- 回调通知：

```
//云端直播推流状态变化通知
CRVideo_SvrMixerStateChanged.callback = function(state, err, operatorID){
    console.log("CRVideo_SvrMixerStateChanged, state:" + state + ",err:" + err + ",operatorID:" + operatorID);
}
```

相关API请参考：

- [CRVideo_StartSvrMixer](#)
- [CRVideo_SvrMixerStateChanged](#)

相关结构定义请参考：

- [CRVideo_MutiMixerCfg](#)
- [CRVideo_MutiMixerContent](#)
- [CRVideo_MutiMixerOutput](#)

3.更新云端直播推流内容

- 接口调用：

```

// 云端录制推荐分辨率
var record_size_arr = [
    [640, 360, 400],
    [964, 480, 600],
    [1280, 720, 1200]
];

var mixerID = "1"; //取之前的混图器id
var size = 0; //使用第1个分辨率

//更新录制布局为：画中画
//混图器内容：画中画布局(示例图如下， 底层640*360， 上层213*120)，底层为用户1的1号摄像头， 上层为用户2的1号摄像头

var bigW = record_size_arr[size][0]; //640， 底层视频的宽度
var bigH = record_size_arr[size][1]; //360， 底层视频的高度

var smallH = bigH / 3; //缩放3倍作为顶层视频的高度
var smallW = smallH * 16 / 9; //视频宽高比为16 : 9， 计算出顶层视频宽度

//底层视频图像的内容配置
var underCon = {
    "type": 0; //0： 录制类型为视频
    "left": 0;
    "top": 0;
    "width": bigW;
    "height": bigH;
    //camid参数为： user1.1 ,表示该内容需要录制userID是user1的1号摄像头。后方可传入-1， 表示为录制默认摄像头
    "param": {"camid": "user1.1"};
    "keepAspectRatio": 1;
};

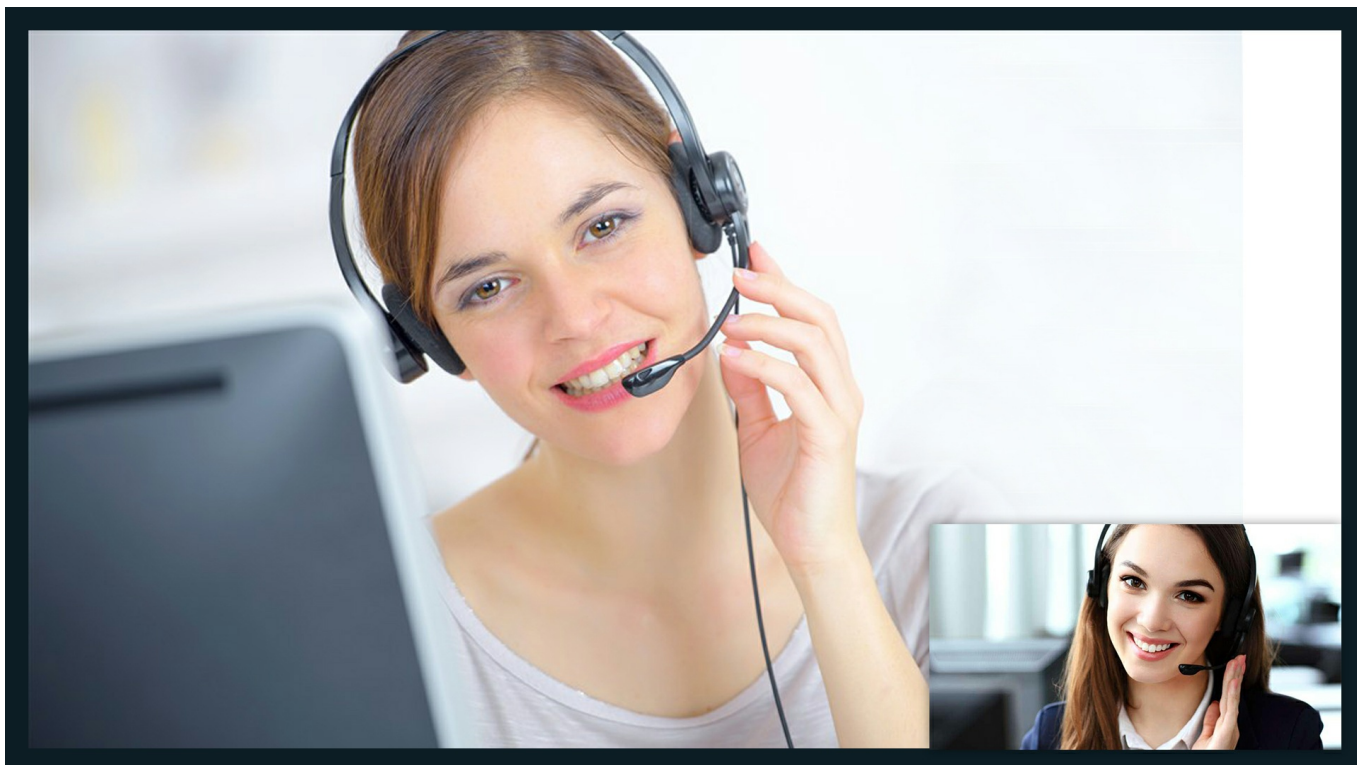
//顶层视频图像的内容配置
var topCon = {
    "type": 0; //0： 录制类型为视频
    "left": bigW - smallW;
    "top": bigH - smallH;
    "width": smallW;
    "height": smallH;
    "param": {"camid": "user2.1"};
    "keepAspectRatio": 1;
};

//配置混图器内容
var mutiMixerContents = [
    {
        "id": mixerID,
        "content": [ underCon ,topCon ] //数组后面的画面将会覆盖前面的画面
    }
];

//更新录制内容， 录制更新后会通知CRVideo_SvrMixerCfgChanged接口
CRVideo_UpdateSvrMixerContent(mutiMixerContents);

```

(2) 画中画布局示例图:



相关API请参考:

- [CRVideo_UpdateSvrMixerContent](#)

4.观众观看直播

通过 [播放器SDK](#) [🔗](#) 观看直播.

5.停止云端直播推流

停止云端直播推流后，也会触发事件 [CRVideo_SvrMixerStateChanged](#)

- 接口调用：

```
CRVideo_StopSvrMixer();
```

js

相关API请参考:

- [CRVideo_StopSvrMixer](#)

6.回放点播

通过 [云屋点播API](#) [🔗](#) 回放点播。

影音播放

功能简介：

把本地的一个视频文件播放给房间内其他用户观看，播放进度完全由播放端控制。

影音播放功能介绍分为播放端和观看端两部分。

注意：一个房间中同一时间只支持进行一个影音播放。支持的影音格式因浏览器不同而不同。

播放端：

1. [创建影音播放组件](#)
2. [开始播放并观看影音](#)
3. [设置播放进度](#)
4. [暂停、停止播放](#)

观看端：

1. [创建影音播放组件](#)
2. [观看影音](#)
3. [影音停止播放通知](#)

两端通用的功能：

1. [播放信息](#)

播放端

1.创建影音播放组件

```
//首先要创建影音播放组件
var mediaObj = CRVideo_CreatMediaObj({
  controls: true, //视频控件是否需要浏览器自带的控制条,
  loop: true, //视频是否循环播放
});
```

js

相关API请参考：

- [CRVideo_CreatMediaObj](#)

2.开始播放并观看影音

每次只能播放一个视频，停止正在播放的视频才能播放下一个视频。通过设置播放配置，还可以控制房间内其他人看到的效果。

- 接口调用：

```
// 开始播放影音：
CRVideo_StartPlayMedia(mediaObj, file, locPlay, bPauseWhenFinished)
```

js

- 观看影音需要监听以下的回调通知：

```
//观看端和播放端都会收到开始播放影音的通知。此时显示影音播放组件，即可观看影音
CRVideo_NotifyMediaStart.callback = function(userid){
  document.body.appendChild(mediaObj.handler()); //把创建的组件放入DOM树中
}
```

js

相关API请参考：

- [CRVideo_StartPlayMedia](#)
- [CRVideo_NotifyMediaStart](#)

3.设置播放进度

可以通过拖动播放组件上的工具条来调整播放进度，也可以用代码来设置播放进度。

- 调用接口：

```
// 设置播放进度，单位：毫秒.例如：设置到2秒处.  
CRVideo_SetMediaplaypos(2000)
```

js

相关API请参考：

- [CRVideo_SetMediaplaypos](#)

4.暂停、停止播放

- 调用接口：

```
// 暂停或恢复播放影音  
CRVideo_PausePlayMedia(true); // true为暂停，false为恢复  
  
//停止播放影音  
CRVideo_StopPlayMedia()
```

js

- 回调通知：

注意：主动调用CRVideo_StopPlayMedia停止播放，或者影音文件播放到结尾，都会触发事件CRVideo_NotifyMediaStop，房间内所有人都会收到。

```
//影音共享暂停播放通知  
CRVideo_NotifyMediaPause.callback = function(userid,pause){  
  
}  
  
//影音共享播放停止通知  
CRVideo_NotifyMediaStop.callback = function(){  
  
}
```

js

相关API请参考：

- [CRVideo_PausePlayMedia](#)
- [CRVideo_StopPlayMedia](#)
- [CRVideo_NotifyMediaPause](#)
- [CRVideo_NotifyMediaStop](#)

观看端：

1.创建影音播放组件

js

```
//首先要创建影音播放显示组件
var mediaObj = CRVideo_CreatMediaObj({
  controls: true, //视频控件是否需要浏览器自带的控制条,
  loop: true, //视频是否循环播放
});
```

相关API请参考：

- [CRVideo_CreatMediaObj](#)

2.观看影音

js

```
//观看端和播放端都会收到开始播放影音的通知。此时显示影音播放组件，即可观看影音
CRVideo_NotifyMediaStart.callback = function(userid){
  mediaObj.setVideo(userid); //将别人开启的共享，需要setVideo。如果是自己开启的共享，无需设置
  document.body.appendChild(mediaObj.handler()); //把创建的组件放入DOM树中
}
```

相关API请参考：

- [CRVideo_NotifyMediaStart](#)

3.影音停止播放通知

js

```
//影音停止播放通知，观看端和播放端都会收到此通知。
CRVideo_NotifyMediaStop.callback = function(){
  document.body.removeChild(mediaObj.handler());
  mediaObj = null;
}
```

两端通用的功能：

播放信息

- 调用接口：

js

```
// 取得影音文件信息
var mediaInfo = CRVideo_Getmediainfo(UID)
```

相关API请参考：

- [CRVideo_Getmediainfo](#)

相关结构定义请参考：

- [CRVideo_MediaInfoObj](#)

呼叫

功能简介：

实现了用户之间的呼叫功能，流程是：A用户先创建一个房间，然后呼叫B用户，如果B用户接受呼叫，AB进入房间进行通讯。

注意：在登录成功后才可以使用呼叫功能

呼叫功能使用流程如下：

1. [创建房间](#)
2. [发起呼叫](#)
3. [接受/拒绝呼叫](#)
4. [挂断](#)
5. [免打扰](#)

1.创建房间

输入房间标题，创建一个没有密码的房间

- 调用接口：

```
//创建房间
CRVideo_CreateMeeting()
```

js

- 回调通知：

```
//创建房间成功
CRVideo_CreateMeetingSuccess.callback = function(meetObj, cookie){
    //创建成功，获取房间信息meetObj，用于呼叫他人
}
//创建房间失败
CRVideo_CreateMeetingFail.callback = function(sdkErr, cookie){
    //创建失败，可以弹出错误提示，不能再执行 进入房间
}
```

js

相关API请参考：

- [CRVideo_CreateMeeting](#)
- [CRVideo_CreateMeetingSuccess](#)
- [CRVideo_CreateMeetingFail](#)

2.发起呼叫

注意：当用户A呼叫用户B时，只有B成功登录了，才可以收到被呼叫通知

- 调用接口：

js

```
//A发起呼叫，邀请用户B进入房间。
var callID = CRVideo_Call(UID, meetObj, usrExtDat, cookie)
```

- 回调通知：

js

```
//呼叫操作成功响应
CRVideo_CallSuccess.callback = function(callID, cookie){

}

// 呼叫操作失败响应
CRVideo_CallFail.callback = function(callID, sdkErr, cookie){
  console.log("呼叫他人失败"+ sdkErr);
}
```

呼叫他人相关API请参考:

- [CRVideo_Call](#)
- [CRVideo_CallSuccess](#)
- [CRVideo_CallFail](#)

3.接受/拒绝呼叫

- 回调通知：

js

```
// SDK通知用户 B 被他人（即callerID）呼叫
CRVideo_NotifyCallIn.callback = function(callID, meetObj, callerID, usrExtDat){
  if(/* B 接受呼叫，进入房间*/){
    CRVideo_AcceptCall(callID, meetObj);
    CRVideo_EnterMeeting3(meetObj.ID, cookie);
  }else{// B 拒绝呼叫

    CRVideo_RejectCall(callID);
  }
}
```

js

```
//当 B 接受呼叫时，A会收到如下通知：
CRVideo_NotifyCallAccepted.call = function(callID, meetObj, usrExtDat){
  //A此时可以进入房间
  CRVideo_EnterMeeting3(meetObj.ID, cookie);
}

//当 B 拒绝呼叫时，A会收到如下通知：
CRVideo_NotifyCallRejected.call = function(callID,meetObj,usrExtDat){
  console.log("客户拒绝呼叫了");
}
```

被呼叫者相关API请参考:

- [CRVideo_NotifyCallIn](#)
- [CRVideo_AcceptCall](#)
- [CRVideo_AcceptCallSuccess](#)
- [CRVideo_AcceptCallFail](#)

- [CRVideo_RejectCall](#)
- [CRVideo_RejectCallSuccess](#)
- [CRVideo_RejectCallFail](#)

呼叫者相关API请参考:

- [CRVideo_NotifyCallAccepted](#)
- [CRVideo_NotifyCallRejected](#)

4.挂断

- 调用接口：

```
//挂断呼叫
CRVideo_HungupCall(callID)
```

- 回调通知：

```
//挂断呼叫成功
CRVideo_HangupCallSuccess.callback = function(callID, cookie){
    console.log("挂断呼叫成功")
    CRVideo_ExitMeeting(); //退出房间
}

//挂断呼叫失败
CRVideo_HangupCallFail.callback = function(callID, sdkErr, cookie){
    console.log("挂断呼叫失败！错误码："+sdkErr);
}

//通知被他人挂断
CRVideo_NotifyCallHungup.callback = function(callID, usrExtDat){
    CRVideo_ExitMeeting(); //退出房间
}
```

相关API请参考:

- [CRVideo_HungupCall](#)
- [CRVideo_ExitMeeting](#)
- [CRVideo_HangupCallSuccess](#)
- [CRVideo_HangupCallFail](#)
- [CRVideo_NotifyCallHungup](#)

5.免打扰

如果用户当前不希望被呼叫，可以把自己的状态设置为免打扰，注意在免打扰状态下不会被呼叫，但是可以主动发起呼叫。

- 调用接口：

```
//开启免打扰
CRVideo_SetDNDStatus(1);

//关闭免打扰
CRVideo_SetDNDStatus(0);
```

- 回调通知：

```
//设置免打扰状态操作成功响应
CRVideo_SetDNDStatusSuccess.callback = function(cookie){

}

//设置免打扰状态操作失败响应
CRVideo_SetDNDStatusFail.callback = function(sdkErr, cookie){
    console.log("设置免打扰状态操作失败，错误码：" + sdkErr);
}
```

相关API请参考：

- [CRVideo_SetDNDStatus](#)
- [CRVideo_SetDNDStatusSuccess](#)
- [CRVideo_SetDNDStatusFail](#)

排队

功能简介：

在呼叫中心的业务场景下，有多个客户呼叫进来，有多个坐席提供服务，简单的一对一呼叫无法满足业务需求。此时可以使用我们的排队功能，客户不再直接呼叫某个坐席，而是呼叫到一个坐席队列，由系统自动给客户分配一个空闲的坐席。

- 当坐席提供的是通用型服务时，多个坐席可以服务于一个队列，客户呼叫这个队列，系统就会自动给客户分配一个空闲坐席。
- 如果是业务高峰期没有空闲的坐席了，客户将在队列中等待，一旦有坐席空闲了就会自动被分配给最先开始等待的客户。
- 坐席可以同时服务多个队列，队列有优先级，当坐席从忙变成空闲时，优先被分配给高优先级的队列里的客户。
- 同一个队列内的坐席可以配置不同的坐席优先级，客户接入时如果同时存在多个空闲坐席，会把客户分配给优先级更高的坐席。

注意：在登录成功后才可以使用排队功能

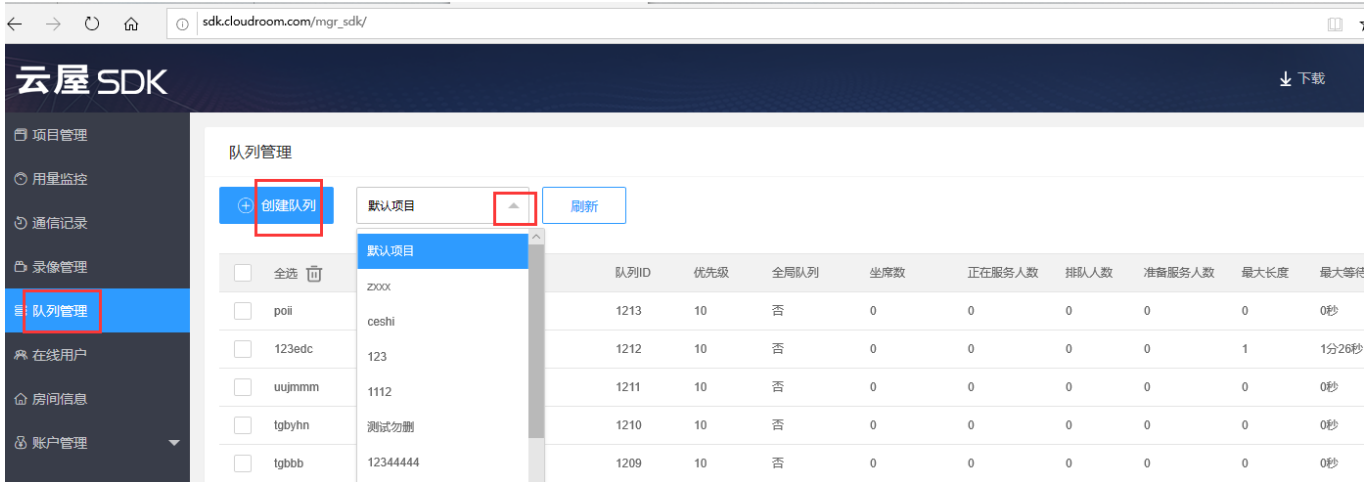
排队流程如下：

1. [创建队列](#)
2. [初始化队列](#)
3. [获取队列信息](#)
4. [坐席进入队列](#)

- 5. 客户排队
- 6. 系统给坐席分配客户
- 7. 坐席呼叫客户

1.创建队列

可以通过两种方式创建队列。第一种是 [登录云屋SDK后台](#) 并创建。如下图：



第二种是通过 [Web API](#) 创建。

2.初始化队列

在登录成功后，初始化队列信息

- 调用接口：

```
//操作完成回调CRVideo_InitQueueDatRslt
CRVideo_InitQueueDat()
```

- 回调通知：

```
//队列初始化操作结果
CRVideo_InitQueueDatRslt.callback = function(sdkErr, cookie){
    if(sdkErr == 0){
        //初始化队列成功后，才可以获取队列相关信息并展示。
    }else{
        console.log("初始化队列信息 失败，错误码： "+ sdkErr);
    }
}
```

相关API请参考：

- [CRVideo_InitQueueDat](#)
- [CRVideo_InitQueueDatRslt](#)

3.获取队列信息

在初始化队列成功后，才可以使用获取队列信息。并且可以多次获取。

- 调用接口：

```
//获取队列信息
var queueList = CRVideo_GetAllQueueInfo();
```

相关API请参考：

- [CRVideo_GetAllQueueInfo](#)

相关结构定义请参考：

- [CRVideo_QueueInfo](#)

4.坐席进入队列

- 调用接口：

```
//开始服务某个队列(可以多次调用，开启对多个队列的服务) .
CRVideo_StartService(queID);

//停止服务某个队列
CRVideo_StopService(queID);

//获取本坐席服务的所有队列
var GetServingQueues = CRVideo_GetServingQueues();
```

- 回调通知：

```
// 开始服务队列操作结果
CRVideo_StartServiceRslt.callback = function(queID, sdkErr, cookie){
    if(sdkErr == 0){
        //开始服务队列成功
    }else{
        console.log("开始服务队列 失败，错误码： "+ sdkErr);
    }
}

//停止服务队列操作结果
CRVideo_StopServiceRslt.callback = function(queID, sdkErr, cookie){
    if(sdkErr == 0){
        //停止服务队列成功
    }else{
        console.log("停止服务队列 失败，错误码： "+ sdkErr);
    }
}

// 开始/停止服务队列，会触发队列状态变化通知
CRVideo_QueueStatusChanged.callback = function(queStatus){

}
```

相关API请参考：

- [CRVideo_StartService](#)
- [CRVideo_StopService](#)
- [CRVideo_StartServiceRslt](#)

- [CRVideo_StopServiceRslt](#)
- [CRVideo_GetServingQueues](#)
- [CRVideo_QueueStatusChanged](#)

相关结构定义请参考：

- [CRVideo_QueStatus](#)

5.客户排队

客户选择一个队列进行排队，每次只能排一个队列

- 调用接口：

```
// 客户开始排队，操作完成回调CRVideo_StartQueuingRslt
CRVideo_StartQueuing(queID)

//客户停止排队，操作完成回调CRVideo_StopQueuingRslt
CRVideo_StopQueuing()
```

- 回调通知：

```
// 开始排队操作结果
CRVideo_StartQueuingRslt.callback = function(sdkErr, cookie){
    if(sdkErr == 0){ //开始排队操作成功
    }else{
        console.log("开始排队操作失败，错误码：" + sdkErr);
    }
}

// 停止排队操作结果
CRVideo_StopQueuingRslt.callback = function(sdkErr, cookie){
    if(sdkErr == 0){ //停止排队操作成功
    }else{
        console.log("停止排队操作失败，错误码：" + sdkErr);
    }
}
```

```
//开始/停止排队，会触发队列状态变化通知
CRVideo_QueueStatusChanged.callback = function(queStatus){

}

//排队信息变化通知
CRVideo_QueueingInfoChanged.callback = function(queueingInfo){

}
```

相关API请参考：

- [CRVideo_StartQueuing](#)
- [CRVideo_StopQueuing](#)
- [CRVideo_StartQueuingRslt](#)
- [CRVideo_StopQueuingRslt](#)

- [CRVideo_QueueingInfoChanged](#)

相关结构定义请参考：

- [CRVideo_QueueingInfo](#)

6.系统给坐席分配客户

客户分配模式有自动和手动两种：

- 在自动分配模式下，坐席一旦空闲系统就会立即分配当前排队的客户过来，适用于坐席可以持续提供服务的业务场景。
- 在手动分配模式下，坐席空闲后系统不自动分配客户，而是坐席准备好后手动触发分配一个当前排队的客户，适用于坐席接待完客户后需要进行信息录入之类的善后工作，或者需要在接待新客户之前进行一些准备工作的业务场景。

自动分配模式：

- 回调通知：

```
js
// 系统自动安排客户
CRVideo_AutoAssignUser.callback = function(QueUser){
    if(/接受系统分配的客户*/){
        CRVideo_AcceptAssignUser(QueUser.queID, QueUser.userID);

        //接下来做其他任务..... 例如：创建房间

    }else{//拒绝系统分配的客户
        CRVideo_RejectAssignUser(user.queID, user.userID);
    }
}

//接受或拒绝分配的客户的操作结果
CRVideo_ResponseAssignUserRslt.callback = function(sdkErr, cookie){
    if(sdkErr == 0){
        console.log("接受或拒绝系统分配的客户成功");
    }else{
        console.log("接受或拒绝系统分配的客户失败，错误码： "+ sdkErr);
    }
}

//系统取消已经安排的客户
CRVideo_CancelAssignUser.callback = function(queID, userid){
    console.log("系统取消已经安排的客户,坐席不应该再进入房间.....");
}
```

相关API请参考：

- [CRVideo_AutoAssignUser](#)
- [CRVideo_AcceptAssignUser](#)
- [CRVideo_RejectAssignUser](#)
- [CRVideo_ResponseAssignUserRslt](#)
- [CRVideo_CancelAssignUser](#)

手动分配模式：

- 调用接口：

```
//坐席请求客户分为 2步：
//1.开启免打扰状态
CRVideo_SetDNDDStatus(1,cookie)：

// 2. 请求分配一个客户
CRVideo_ReqAssignUser(cookie);
```

- 回调通知：

```
//设置免打扰成功通知
CRVideo_SetDNDDStatusSuccess.callback = function(cookie){

}

//设置免打扰失败通知
CRVideo_SetDNDDStatusFail.callback = function(sdkErr, cookie){
    console.log("设置免打扰失败，错误码：" + sdkErr);
}

//请求分配客户操作结果
CRVideo_ReqAssignUserRslt.callback = function(sdkErr, user, cookie){
    if(sdkErr == 0){
        //请求分配客户成功，接下来做其他任务..... 例如：创建房间
    }else{
        console.log("请求分配客户失败，错误码：" + sdkErr);
    }
}
```

相关API请参考：

- [CRVideo_SetDNDDStatus](#)
- [CRVideo_SetDNDDStatusSuccess](#)
- [CRVideo_SetDNDDStatusFail](#)
- [CRVideo_ReqAssignUser](#)
- [CRVideo_ReqAssignUserRslt](#)

7.坐席呼叫客户

- 调用接口：

```
//发起呼叫，邀请客户进入房间。
var callID = CRVideo_Call(UID, meetObj, usrExtDat, cookie)
```

呼叫相关流程及API请参考：

- [呼叫功能](#)

透明通道

功能简介：根据发送数据的大小，提供了发送小块数据、发送大块数据、发送文件三种方式。

1. [发送小块数据](#)
2. [发送大块数据](#)
3. [发送进度通知](#)

1.发送小块数据

注意：此接口使用后，不能使用`CRVideo_CancelSend`。一次性发送，也不会有进度通知

- 调用接口：

```
//发送小块数据,taskId为分配的任务ID
var taskId = CRVideo_SendCmd(UID, data);
```

js

- 回调通知：

```
//发送数据的结果通知
CRVideo_SendCmdRslt.callback = function(taskID, sdkErr, cookie){
  if(sdkErr != 0){
    console.log("发送失败，错误码："+sdkErr);
  }
}
```

js

相关API请参考：

- [CRVideo_SendCmd](#)
- [CRVideo_SendCmdRslt](#)

相关结构定义请参考：

- [错误码](#)

2.发送大块数据

分块发送，进度通知事件 [CRVideo_SendProgress](#)，调用 [CRVideo_CancelSend](#) 取消发送。

- 调用接口：

```
//发送大块数据,taskId为分配的任务ID
var taskId = CRVideo_SendBuffer(UID, data);
```

js

- 回调通知：

js

```
//发送数据的结果通知
CRVideo_SendBufferRslt.callback = function(taskID, sdkErr, cookie){
  if(sdkErr != 0){
    console.log("发送失败，错误码："+sdkErr);
  }
}
```

相关API请参考：

- [CRVideo_SendBuffer](#)
- [CRVideo_SendBufferRslt](#)

3.发送进度通知

- 回调通知：

js

```
//发送的进度通知事件
CRVideo_SendProgress.callback = function(taskID, sendLen, totalLen, cookie){

}
```

相关API请参考：

- [CRVideo_SendProgress](#)

API

| | |
|-------|----------|
| 房间外接口 | 基础函数 |
| | 初始化/反初始化 |
| | 登录/注销/掉线 |
| | 队列管理 |
| | 呼叫 |
| | 邀请 |
| | 房间管理 |
| | 进出房间 |
| | 房间内基础消息 |
| | 房间、用户属性 |
| | 成员管理 |
| | 音频管理 |

| | |
|-------|-----------|
| 房间内接口 | 视频管理 |
| | 屏幕共享 |
| | 影音共享 |
| | 电子白板 |
| | 云端录制/互动直播 |
| | 可视化UI组件 |
| | 白板UI组件 |
| 消息接口 | 消息通道（房间外） |
| | 广播消息（房间内） |

接口描述

基础函数

| 类型 | 接口 | 描述 |
|----|-----------------------|-------------|
| 主调 | CRVideo_GetSDKVersion | 获取当前SDK版本号 |
| | CRVideo_WriteLog | 向日志中写入自定义内容 |
| | CRVideo_SetSDKParams | 向日志中写入自定义内容 |

初始化/反初始化

| 类型 | 接口 | 描述 |
|----|----------------|---------|
| 主调 | CRVideo_Init | SDK初始化 |
| | CRVideo_Uninit | SDK反初始化 |

登录/注销/掉线

| 类型 | 接口 | 描述 |
|----|-------------------------------|--------------|
| 主调 | CRVideo_SetServerAddr | 设置服务器地址 |
| | CRVideo_Login | 登录（帐号密码鉴权） |
| | CRVideo_LoginByToken | 登录（Token鉴权） |
| | CRVideo_UpdateToken | 更新Token |
| | CRVideo_Logout | 注销登录 |
| 回调 | CRVideo_LoginSuccess | 回调，登陆成功 |
| | CRVideo_LoginFail | 回调，登录失败 |
| | CRVideo_NotifyTokenWillExpire | 通知，Token即将失效 |
| | CRVideo_LineOff | 通知，从服务器掉线了 |

队列管理

| 类型 | 接口 | 描述 |
|----|--------------------------------|--------------------|
| 主调 | CRVideo_InitQueueDat | 初始化队列数据 |
| | CRVideo_GetAllQueueInfo | 查询所有队列的信息 |
| | CRVideo_RefreshAllQueueStatus | 刷新所有队列的状态 |
| | CRVideo_GetQueueStatus | 获取某个队列的状态 |
| | CRVideo_GetQueuingInfo | 获取我的排队信息 |
| | CRVideo_GetServingQueues | 获取我当前服务的所有队列 |
| | CRVideo_GetSessionInfo | 获取我的会话信息 |
| | CRVideo_StartQueuing | 开始排队 |
| | CRVideo_StopQueuing | 停止排队 |
| | CRVideo_StartService | 坐席开始服务某个队列 |
| | CRVideo_StopService | 坐席停止服务某个队列 |
| | CRVideo_ReqAssignUser | 坐席请求分配一个客户 |
| | CRVideo_AcceptAssignUser | 坐席接受系统分配的客户 |
| | CRVideo_RejectAssignUser | 坐席拒绝系统分配的客户 |
| 回调 | CRVideo_InitQueueDatRslt | 回调，队列初始化操作的结果 |
| | CRVideo_QueueStatusChanged | 通知，队列状态变化 |
| | CRVideo_GetQueueStatusRslt | 回调，获取某个队列状态的结果 |
| | CRVideo_QueuingInfoChanged | 通知，排队信息变化（客户） |
| | CRVideo_StartQueuingRslt | 回调，开始排队的结果（客户） |
| | CRVideo_StopQueuingRslt | 回调，停止排队的结果（客户） |
| | CRVideo_StartServiceRslt | 回调，开始服务某个队列的结果（坐席） |
| | CRVideo_StopServiceRslt | 回调，停止服务某个队列的结果（坐席） |
| | CRVideo_ResponseAssignUserRslt | 回调，响应系统分配客户的结果（坐席） |
| | CRVideo_AutoAssignUser | 通知，系统自动分配了客户 |
| | CRVideo_ReqAssignUserRslt | 回调，请求系统分配客户的结果（坐席） |
| | CRVideo_CancelAssignUser | 通知，系统取消分配客户（坐席） |

呼叫

| 类型 | 接口 | 描述 |
|----|----------------------|-----------|
| | CRVideo_Call | 发起呼叫 |
| | CRVideo_AcceptCall | 接受对方发起的呼叫 |
| | CRVideo_RejectCall | 拒绝对方发起的呼叫 |
| | CRVideo_HangupCall | 挂断当前的呼叫 |
| | CRVideo_SetDNDStatus | 设置免打扰状态 |

| | | |
|----|-----------------------------------|-------------------|
| 主调 | CRVideo_GetUserStatus | 获取AppID下所有用户在线状态 |
| | CRVideo_GetOneUserStatus | 获取AppID下指定用户的在线状态 |
| | CRVideo_StartUserStatusNotify | 开启用户状态推送 |
| | CRVideo_StopUserStatusNotify | 关闭用户状态推送 |
| 回调 | CRVideo_CallSuccess | 回调，发起呼叫成功 |
| | CRVideo_CallFail | 回调，发起呼叫失败 |
| | CRVideo_AcceptCallSuccess | 回调，接受他人的呼叫成功 |
| | CRVideo_AcceptCallFail | 回调，接受他人的呼叫失败 |
| | CRVideo_RejectCallSuccess | 回调，拒绝他人的呼叫成功 |
| | CRVideo_RejectCallFail | 回调，拒绝他人的呼叫失败 |
| | CRVideo_HangupCallSuccess | 回调，挂断呼叫成功 |
| | CRVideo_HangupCallFail | 回调，挂断呼叫失败 |
| | CRVideo_NotifyCallIn | 通知，有人呼叫自己 |
| | CRVideo_NotifyCallAccepted | 通知，呼叫被对方接受 |
| | CRVideo_NotifyCallRejected | 通知，呼叫被对方拒绝 |
| | CRVideo_NotifyCallHangup | 通知，呼叫被对方挂断 |
| | CRVideo_SetDNDStatusSuccess | 回调，设置免打扰状态成功 |
| | CRVideo_SetDNDStatusFail | 回调，设置免打扰状态失败 |
| | CRVideo_GetUserStatusSuccess | 回调，获取用户在线状态成功 |
| | CRVideo_GetUserStatusFail | 回调，获取用户在线状态失败 |
| | CRVideo_StartUserStatusNotifyRslt | 回调，开启用户状态推送的结果 |
| | CRVideo_StopUserStatusNotifyRslt | 回调，停止用户状态推送的结果 |
| | CRVideo_NotifyUserStatus | 通知，用户状态变化 |

邀请

| 类型 | 接口 | 描述 |
|----|-----------------------------|----------------|
| 主调 | CRVideo_Invite | 发送邀请 |
| | CRVideo_AcceptInvite | 接受对方发送的邀请 |
| | CRVideo_RejectInvite | 拒绝对方发送的邀请 |
| | CRVideo_CancelInvite | 取消自己发送的邀请 |
| | CRVideo_InviteSuccess | 回调，发送邀请成功 |
| | CRVideo_InviteFail | 回调，发送邀请失败 |
| | CRVideo_AcceptInviteSuccess | 回调，接受对方发送的邀请成功 |
| | CRVideo_AcceptInviteFail | 回调，接受对方发送的邀请失败 |
| | CRVideo_RejectInviteSuccess | 回调，拒绝对方发送的邀请成功 |

| | | |
|----|------------------------------|----------------|
| 回调 | CRVideo_RejectInviteFail | 回调，拒绝对方发送的邀请失败 |
| | CRVideo_CancelInviteSuccess | 回调，取消自己发送的邀请成功 |
| | CRVideo_CancelInviteFail | 回调，取消自己发送的邀请失败 |
| | CRVideo_NotifyInviteIn | 通知，收到别人的邀请 |
| | CRVideo_NotifyInviteAccepted | 通知，我发送的邀请被对方接受 |
| | CRVideo_NotifyInviteRejected | 通知，我发送的邀请被对方拒绝 |
| | CRVideo_NotifyInviteCanceled | 通知，邀请已经被取消了 |

房间管理

| 类型 | 接口 | 描述 |
|----|------------------------------|-------------|
| 主调 | CRVideo_CreateMeeting | 创建房间 |
| | CRVideo_DestroyMeeting | 销毁房间 |
| 回调 | CRVideo_CreateMeetingSuccess | 回调，创建房间成功 |
| | CRVideo_CreateMeetingFail | 回调，创建房间失败 |
| | CRVideo_DestroyMeetingRslt | 回调，销毁房间的结果 |
| | CRVideo_MeetingStopped | 通知，当前房间已被结束 |

进出房间

| 类型 | 接口 | 描述 |
|----|--------------------------|--------------|
| 主调 | CRVideo_EnterMeeting3 | 进入房间 |
| | CRVideo_ExitMeeting | 离开房间 |
| | CRVideo_Kickout | 将房间内指定成员踢出房间 |
| 回调 | CRVideo_EnterMeetingRslt | 回调，进入房间的结果 |
| | CRVideo_UserEnterMeeting | 通知，有人进入房间 |
| | CRVideo_UserLeftMeeting | 通知，有人离开房间 |
| | CRVideo_MeetingDropped | 通知，从房间中掉线了 |

房间内基础消息

| 类型 | 接口 | 描述 |
|----|----------------------------|--------------|
| 主调 | CRVideo_GetMainVideo | 获取当前哪个用户为主视频 |
| | CRVideo_SetMainVideo | 设置哪个用户为主视频 |
| | CRVideo_SetMainVideoMode | 设置主视频模式 |
| | CRVideo_SwitchToPage | 功能区切换 |
| | CRVideo_GetCurrentMainPage | 获取当前主功能区 |

| | | |
|----|------------------------------------|---------------|
| | CRVideo_GetCurrentSubPage | 获取当前子页面 |
| 回调 | CRVideo_NotifyMainVideoChanged | 通知，会话内主视频变化 |
| | CRVideo_NotifyMainVideoModeChanged | 通知，会话内主视频模式变化 |
| | CRVideo_NotifySwitchToPage | 通知，会话内功能区变化 |

房间、用户属性

| 类型 | 接口 | 描述 |
|----|-------------------------------------|-----------------|
| 主调 | CRVideo_GetMeetingAttrs | 获取房间部份属性 |
| | CRVideo_GetMeetingAllAttrs | 获取房间所有属性 |
| | CRVideo_SetMeetingAttrs | 重置房间属性集 |
| | CRVideo_AddOrUpdateMeetingAttrs | 增加或者更新房间属性 |
| | CRVideo_DelMeetingAttrs | 删除房间属性 |
| | CRVideo_ClearMeetingAttrs | 清除房间所有属性 |
| | CRVideo_GetUserAttrs | 获取指定用户的指定属性 |
| | CRVideo_SetUserAttrs | 重置用户属性 |
| | CRVideo_AddOrUpdateUserAttrs | 增加或更新用户属性 |
| | CRVideo_DelUserAttrs | 删除用户的特定属性 |
| | CRVideo_ClearUserAttrs | 清除特定用户的所有属性 |
| | CRVideo_ClearAllUserAttrs | 清除所有用户的所有属性 |
| 回调 | CRVideo_GetMeetingAttrsSuccess | 回调，成功获取到房间指定属性 |
| | CRVideo_GetMeetingAttrsFail | 回调，获取房间指定属性失败 |
| | CRVideo_GetMeetingAllAttrsSuccess | 回调，成功获取到房间所有属性 |
| | CRVideo_GetMeetingAllAttrsFail | 回调，获取房间属性失败 |
| | CRVideo_SetMeetingAttrsRsIt | 回调，设置用户属性结果 |
| | CRVideo_AddOrUpdateMeetingAttrsRsIt | 回调，增加或更新房间属性结果 |
| | CRVideo_DelMeetingAttrsRsIt | 回调，删除房间属性结果 |
| | CRVideo_ClearMeetingAttrsRsIt | 回调，清除房间属性结果 |
| | CRVideo_GetUserAttrsSuccess | 回调，获取房间内用户属性成功 |
| | CRVideo_GetUserAttrsFail | 回调，获取房间内用户属性失败 |
| | CRVideo_SetUserAttrsRsIt | 回调，设置用户属性结果 |
| | CRVideo_AddOrUpdateUserAttrsRsIt | 回调，增加或者更新用户属性结果 |
| | CRVideo_DelUserAttrsRsIt | 回调，删除特定用户属性结果 |
| | CRVideo_ClearUserAttrsRsIt | 回调，清除特定用户属性结果 |
| | CRVideo_ClearAllUserAttrsRsIt | 回调，清除所有用户属性结果 |
| | CRVideo_NotifyMeetingAttrsChanged | 通知，房间属性发生改变 |

| | | |
|--|--------------------------------|---------------|
| | CRVideo_NotifyUserAttrsChanged | 通知，某个用户属性发生改变 |
|--|--------------------------------|---------------|

房间成员管理

| 类型 | 接口 | 描述 |
|----|-------------------------------|----------------|
| 主调 | CRVideo_GetAllMembers | 获取房间内所有成员的信息 |
| | CRVideo_GetMemberInfo | 获取房间内指定成员的信息 |
| | CRVideo_GetMemberNickName | 获取房间内指定成员的昵称 |
| | CRVideo_IsUserInMeeting | 判断某个用户是否在房间中 |
| | CRVideo_SetNickName | 设置房间内某个成员的昵称 |
| 回调 | CRVideo_NotifyNickNameChanged | 通知，房间内某个成员昵称变化 |

音频管理

| 类型 | 接口 | 描述 |
|----|----------------------------|-----------------|
| 主调 | CRVideo_GetAudioMicNames | 获取本地麦克风设备列表 |
| | CRVideo_GetAudioSpkNames | 获取本地扬声器设备列表 |
| | CRVideo_SetAudioCfg | 设置本地音频参数 |
| | CRVideo_GetAudioCfg | 获取本地音频参数 |
| | CRVideo_GetMicEnergy | 获取指定成员的麦克风声音大小 |
| | CRVideo_OpenMic | 打开指定成员的麦克风 |
| | CRVideo_CloseMic | 关闭指定成员的麦克风 |
| | CRVideo_GetAudioStatus | 获取指定成员的麦克风状态 |
| | CRVideo_SetAllAudioClose | 关闭房间内所有人麦克风 |
| | CRVideo_PauseSpeaker | 暂停播放房间内通话声音（静音） |
| | CRVideo_PlaySpeaker | 恢复播放房间内通话声音 |
| 回调 | CRVideo_AudioStatusChanged | 通知，成员麦克风状态变化 |
| | CRVideo_MicEnergyUpdate | 通知，成员麦克风声音能量变化 |

视频管理

| 类型 | 接口 | 描述 |
|----|-------------------------|--------------|
| | CRVideo_CreatVideoObj | 创建成员视频UI显示对象 |
| | CRVideo_GetVideoCfg | 获取当前视频参数 |
| | CRVideo_SetVideoCfg | 设置视频参数 |
| | CRVideo_GetVideoStatus | 获取指定成员的视频状态 |
| | CRVideo_GetDefaultVideo | 获取指定成员的默认摄像头 |

| | | |
|----|----------------------------|----------------------|
| 主调 | CRVideo_SetDefaultVideo | 设置指定成员的默认摄像头 |
| | CRVideo_GetAllVideoInfo | 获取指定成员的所有摄像头信息 |
| | CRVideo_GetOpenedVideoIDs | 获取指定成员当前打开的摄像头ID列表 |
| | CRVideo_OpenVideo | 打开指定成员的摄像头 |
| | CRVideo_CloseVideo | 关闭指定成员的摄像头 |
| | CRVideo_SetEnableMutiVideo | 设置是否启用多摄像头 |
| | CRVideo_GetEnableMutiVideo | 获取当前是否启用多摄像头 |
| | CRVideo_AddStreamCam | 添加WebRTC媒体流摄像头 |
| | CRVideo_RemoveStreamVideo | 移除WebRTC媒体流摄像头 |
| | CRVideo_AddCanvasVCam | 添加canvas虚拟摄像头 |
| | CRVideo_RemoveCanvasVCam | 移除canvas摄像头 |
| 回调 | CRVideo_VideoDevChanged | 通知，某个成员的摄像头设备发生了变化 |
| | CRVideo_DefVideoChanged | 通知，某个成员的默认摄像头发生了变化 |
| | CRVideo_VideoStatusChanged | 通知，某个成员的视频状态发生了变化 |
| | CRVideo_NotifyVideoStream | 通知，获取到某个成员的WebRTC视频流 |

屏幕共享

| 类型 | 接口 | 描述 |
|----|----------------------------------|-------------------|
| 主调 | CRVideo_CreatScreenShareObj | 创建屏幕共享显示UI组件 |
| | CRVideo_SetScreenShareCfg | 配置屏幕共享参数 |
| | CRVideo_StartScreenShare | 开启本地屏幕共享 |
| | CRVideo_StopScreenShare | 结束本地屏幕共享 |
| | CRVideo_StartScreenMark | 开启房间内屏幕共享标注 |
| | CRVideo_StopScreenMark | 停止房间内屏幕共享标注 |
| 回调 | CRVideo_StartScreenShareRslt | 回调，开启屏幕共享的结果 |
| | CRVideo_StopScreenShareRslt | 回调，停止屏幕共享的结果 |
| | CRVideo_StartScreenMarkRslt | 回调，开启房间内屏幕共享标注的结果 |
| | CRVideo_StopScreenMarkRslt | 回调，停止房间内屏幕共享标注的结果 |
| | CRVideo_NotifyScreenShareStarted | 通知，他人开启了屏幕共享 |
| | CRVideo_NotifyScreenShareStopped | 通知，他人停止了屏幕共享 |
| | CRVideo_NotifyScreenMarkStarted | 通知，房间内开启了屏幕共享标注 |
| | CRVideo_NotifyScreenMarkStopped | 通知，房间内停止了屏幕共享标注 |

影音共享

| 类型 | 接口 | 描述 |
|----|-----------------------------|----------------|
| 主调 | CRVideo_CreatMediaObj | 创建影音共享显示UI组件 |
| | CRVideo_StartPlayMedia | 开始播放影音 |
| | CRVideo_PausePlayMedia | 暂停或恢复播放影音 |
| | CRVideo_StopPlayMedia | 停止播放影音及共享 |
| | CRVideo_SetMediaplaypos | 设置播放进度 |
| | CRVideo_SetMediaVolume | 设置本地媒体音量 |
| | CRVideo_Getmediainfo | 获取影音播放信息 |
| | CRVideo_GetMediaImg | 获取影音图像数据 |
| 回调 | CRVideo_NotifyMediaOpened | 通知，影音文件已打开 |
| | CRVideo_NotifyMediaStart | 通知，影音开始播放 |
| | CRVideo_NotifyMediaStop | 通知，影音播放停止 |
| | CRVideo_NotifyMediaPause | 通知，影音播放暂停/取消暂停 |
| | CRVideo_NotifyPlayPosSetted | 通知，影音播放位置更新 |

电子白板

| 类型 | 接口 | 描述 |
|----|------------------------------|------------|
| 主调 | CRVideo_Creatboard | 创建白板 |
| | CRVideo_Closeboard | 关闭白板 |
| | CRVideo_SetBoardCurPageNo | 白板翻页 |
| | CRVideo_CreateBoardUIObj | 白板翻页 |
| 回调 | CRVideo_NotifyInitBoards | 通知，初始化白板列表 |
| | CRVideo_NotifyCreateBoard | 通知，创建了新的白板 |
| | CRVideo_NotifyCloseBoard | 通知，关闭某个白板 |
| | CRVideo_NotifyBoardCurPageNo | 通知，白板翻页 |

云端录制/互动直播

| 类型 | 接口 | 描述 |
|----|-------------------------------|--------------------|
| 主调 | CRVideo_StartSvrMixer | 开始云端录制、云端直播推流 |
| | CRVideo_UpdateSvrMixerContent | 更新云端录制、云端直播推流内容 |
| | CRVideo_StopSvrMixer | 停止云端录制、云端直播推流 |
| | CRVideo_GetSvrMixerState | 获取云端录制、云端直播推流状态 |
| | CRVideo_SvrMixerStateChanged | 通知，云端录制、云端直播推流状态变化 |
| | | |

| | | |
|----|----------------------------|----------------------|
| 回调 | | |
| | CRVideo_SvrMixerOutputInfo | 通知，云端录制文件、云端直播推流信息变化 |

可视化UI组件

| 组件 | 方法 | 描述 |
|-----------------------------------------------------------------------|--------------------|------------------------|
| CRVideo_VideoObj、 CRVideo_ScreenShareObj、 CRVideo_MediaObj 公共方法 | id | 设置/查询组件的独立ID |
| | height | 设置/查询组件的高度 |
| | width | 设置/查询组件的宽度 |
| | left | 设置/查询组件在父元素中的left位置 |
| | top | 设置/查询组件在父元素中的top位置 |
| | show | 显示组件 |
| | hide | 隐藏组件 |
| | handler | 获取该组件的DOM对象 |
| | setVideo | 设置组件上要显示的视频画面，订阅视频 |
| | clear | 清除组件中的图像 |
| | isPicEmpty | 检查组件当前图像是否为空 |
| | setNickNameStyle | 设置组件上昵称的显示样式 |
| | setNickNameContent | 设置组件上昵称的显示内容 |
| | setVisibleNickName | 设置组件上是否显示昵称 |
| | getVisibleNickName | 获取组件当前是否显示昵称 |
| | dblClickFullScreen | 设置/查询组件是否支持双击全屏 |
| CRVideo_VideoObj | savePic | 保存组件当前显示的一帧画面为图片文件 |
| | savePicToBase64 | 保存组件当前显示的一帧画面为base64数据 |
| | | |
| CRVideo_VideoObj | setVideo | 订阅成员摄像头画面 |
| | getVideoID | 查询组件当前订阅的摄像头ID |
| | destroy | 销毁组件，退订视频 |
| CRVideo_ScreenShareObj | setMark | 设置屏幕共享标注样式 |

白板UI组件

| 组件 | 方法 | 描述 |
|----|------------------|--------------------|
| | handler | 获取白板UI组件的Dom对象 |
| | setCurBoardPage | 设置该组件上要显示的白板页 |
| | isScaledContents | 设置（查询）当前组件是否内容允许缩放 |
| | setTitleStyle | 设置当前组件的标题样式 |
| | setBoardStyle | 设置当前组件的canvas样式 |

| | | |
|--------------------|---------------|---------------|
| CRVideo_BoardUIObj | setPen | 设置当前组件的画笔样式 |
| | undo | 撤销一笔 |
| | redo | 恢复一笔 |
| | clear | 清空组件 |
| | addElement | 添加自定义图元（暂不支持） |
| | removeElement | 移除指定图元（暂不支持） |

房间外点对点消息通道

| 类型 | 接口 | 描述 |
|----|--------------------------|---------------|
| 主调 | CRVideo_SendCmd | 发送点对点消息 |
| | CRVideo_SendBuffer | 发送大块数据 |
| | CRVideo_SendFile | 发送文件数据 |
| | CRVideo_CancelSend | 取消发送 |
| 回调 | CRVideo_SendCmdRslt | 回调，发送点对点消息的结果 |
| | CRVideo_SendBufferRslt | 回调，发送大块数据的结果 |
| | CRVideo_SendFileRslt | 回调，发送文件的结果 |
| | CRVideo_SendProgress | 通知，当前发送进度 |
| | CRVideo_NotifyCmdData | 通知，收到点对点消息 |
| | CRVideo_NotifyBufferData | 通知，收到大块数据 |
| | CRVideo_NotifyFileData | 通知，收到文件数据 |

房间内广播消息

| 类型 | 接口 | 描述 |
|----|----------------------------------|-----------------|
| 主调 | CRVideo_SendMeetingCustomMsg | 发送房间内广播消息 |
| 回调 | CRVideo_SendMeetingCustomMsgRslt | 回调，发送房间内广播消息的结果 |
| | CRVideo_NotifyMeetingCustomMsg | 通知，收到房间内广播消息 |

接口详情

CRVideo_GetSDKVersion()

- 描述：获取当前SDK版本号
- 返回值：当前sdk版本，如："1.8.0.3"
- 回调：无

CRVideo_WriteLog(strLog)

- 描述：向日志中写入自定义内容

- 返回值：无
- 回调：无

CRVideo_SetSDKParams(SDKParams)

- 描述：向日志中写入自定义内容
- 返回值：无
- 回调：无
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|--------------------------------------------------------|
| SDKParams | Object | 否 | SDK参数对象，详见： CRVideo_SDKParams ，不传则为默认值 |

CRVideo_Init(SDKParams).then(res=>{},err=>{})

- 描述：SDK初始化
- 返回值：返回一个promise对象，成功返回resolve，失败返回reject [错误码](#)
- 回调：无
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|-------------------------------------------------------------|
| SDKParams | Object | 否 | SDK参数对象，详见： CRVideo_SDKParams ，可以不传，则全部为默认值 |

示例代码：

```
// 传一个对象，对象内设置具体参数（只传需要修改的参数，不传的参数为缺省值）
CRVideo_Init({
  isCallSer: false,
  MSPProtocol: 1,
  videoSendBW: 1024,
  //...
}).then(res => {
  //初始化成功后的处理...
}, err => {
  //初始化失败后的处理...
})

//或者不传这个对象，全部为缺省值
CRVideo_Init().then(res => {
  //初始化成功后的处理...
}, err => {
  //初始化失败后的处理...
})

// 注：.then(callback1,callback2)方法中的参数为两个回调函数
// 第一个参数为初始化成功的回调函数，第二个参数为初始化失败的回调函数
// 可以通过错误码判断初始化失败的原因
```

注意:

程序开始时init，程序结束时uninit，整个程序的生命周期中只进行一次初始化和反初始化。

webRTC默认不支持http协议（chrome78+已支持file:///协议），只能通过配置127.0.0.1或localhost访问，或通过配置Chrome调试模式绕过此限制。

CRVideo_Uninit()

- **描述：** SDK反初始化
- **返回值：** 无
- **回调：** 无

CRVideo_SetServerAddr(ServerAddr)

- **描述：** 设置服务器地址
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|------------------------------------------------|
| ServerAddr | String | 是 | 登录的服务器地址，如：10.8.8.122:2726 或 sdk.cloudroom.com |

CRVideo_Login(appID,appSecret,nickname,UID,userAuthCode,cookie)

- **描述：** 登录（帐号密码鉴权）
- **返回值：** 无
- **回调：** 登陆成功则回调 [CRVideo_LoginSuccess.callback](#)，登陆失败则回调 [CRVideo_LoginFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------------|--------|----|--------------------------------------------------------------------------------------------------------|
| appID | String | 是 | 云屋鉴权帐号。用于区分不同的项目，每个项目都有属于自己的唯一的appID，不同appID的项目完全独立，无法相互通信。appID请联系云屋销售获取，并开通相关套餐和功能权限。 |
| appSecret | String | 是 | md5(appSecret)。(appSecret: 云屋鉴权密码, 可登录云屋管理后台修改) |
| nickname | String | 是 | 昵称 |
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| userAuthCode | String | 否 | 用户验证信息，为空代表不需要sdk验证（一般由业务方先对UID认证完了，再来使用云屋sdk)。如果不为空，云屋sdk服务器将连接提前配置好的业务方服务器进行实时验证。（ SDK管理后台-项目管理可以配置） |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_LoginByToken(Token,nickname,UID,userAuthCode,cookie)

- **描述：** 登录（Token鉴权）
- **返回值：** 无
- **回调：** 登陆成功则回调 [CRVideo_LoginSuccess.callback](#)，登陆失败则回调 [CRVideo_LoginFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|----------|
| Token | String | 是 | Token鉴权码 |

| | | | |
|--------------|--------|---|------------------------------------------------------------------------------------------------------|
| nickname | String | 是 | 昵称 |
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| userAuthCode | String | 否 | 用户验证信息，为空代表不需要sdk验证（一般由业务方先对UID认证完了，再来使用云屋sdk）。如果不为空，云屋sdk服务器将连接提前配好的业务方服务器进行实时验证。（SDK管理后台-项目管理可以配置） |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_UpdateToken(Token)

- **描述：**更新Token
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|------------|
| Token | String | 是 | 新的Token鉴权码 |

CRVideo_Logout()

- **描述：**注销登录
- **返回值：**无
- **回调：**无

退出程序时，必须注销本次登录，然后再进行SDK反初始化操作。

CRVideo_LoginSuccess.callback=function(UID,cookie) {}

- **描述：**回调，登陆成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_LoginFail.callback=function(sdkErr,cookie) {}

- **描述：**回调，登录失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |

| | | |
|--------|--------|----------------------------------|
| cookie | String | 详细介绍见 关键词 ，可以传空串 |
|--------|--------|----------------------------------|

CRVideo_NotifyTokenWillExpire.callback=function(){}

- **描述：** 通知，Token即将失效

此回调将在token失效前30s通知，此时应重新获取新的Token后调用 [CRVideo_UpdateToken](#) 接口及时更新Token，否则将被强制下线

CRVideo_LineOff.callback=function(sdkErr){}

- **描述：** 通知，从服务器掉线了
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |

从服务器掉线后，应再次调用登录接口重新登录服务器，可以用 [cookie](#) 来区分是掉线后的重登还是第一次登录，需要注意的是，登录服务器和登录房间是不同的接口，掉线后的通知也是不同的回调，重新登录也应该分别做不同的处理。

CRVideo_InitQueueDat(cookie)

- **描述：** 初始化队列数据
- **返回值：** 无
- **回调：** [CRVideo_InitQueueDatRsIt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

队列初始化成功后才可获取队列相关信息

CRVideo_GetAllQueueInfo()

- **描述：** 查询所有队列的信息
- **返回值：** 所有队列的信息集合（数组），其中每个队列信息：[CRVideo_QueueInfo](#)
- **回调：** 无

CRVideo_RefreshAllQueueStatus(nolog)

- **描述：** 刷新所有队列的状态
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|------|----|--------------------------------------------------------------|
| nolog | Bool | 否 | 是否禁止SDK打印此接口的日志，缺省为空，不禁止。当需要频繁调用此接口时，可以传true，SDK将不会打印此接口的日志。 |

刷新所有队列的状态会触发多次 [CRVideo_GetQueueStatusRsIt.callback](#) 回调，从该回调中获取每个队列的状态

CRVideo_GetQueueStatus(queID,cookie)

- **描述：** 获取某个队列的状态
- **返回值：** 无
- **回调：** [CRVideo_GetQueueStatusRsIt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| queID | Number | 是 | 队列ID |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetQueuingInfo()

- **描述：** 获取我的排队信息
- **返回值：** 返回排队信息 [CRVideo_QueuingInfo](#)
- **回调：** 无

CRVideo_GetServingQueues()

- **描述：** 获取我当前服务的所有队列
返回我当前服务的队列ID列表，Array: 如
- **返回值：** [1,2,3]
- **回调：** 无

CRVideo_GetSessionInfo()

- **描述：** 获取我的会话信息
- **返回值：** 返回会话信息 [CRVideo_SessionInfo](#)
- **回调：** 无

此接口用来在会话过程中掉线并重登后，获取上一次会话的信息以便重新进入之前的会话
此会话信息只会在初始化队列成功时更新一次，会话过程中不会更新

CRVideo_StartQueuing(queID,usrExtDat,cookie)

- **描述：** 开始排队
- **返回值：** 无
- **回调：** [CRVideo_StartQueuingRsIt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|----------------------------------|
| queID | Number | 是 | 队列ID |
| usrExtDat | String | 否 | 用户排队携带的扩展数据, 座席在服务时可获得 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

客户只允许同时在一个队列排队，需要更换队列排队时，应该先 [停止排队](#)，再重新开始排队
客户开始排队时会触发排队信息变化的通知 [CRVideo_QueuingInfoChanged.callback](#)

CRVideo_StopQueueing(cookie)

- **描述：** 停止排队
- **返回值：** 无
- **回调：** [CRVideo_StopQueueingRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StartService(queID,priority,cookie)

- **描述：** 坐席开始服务某个队列
- **返回值：** 无
- **回调：** [CRVideo_StartServiceRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------|--------|----|----------------------------------|
| queID | Number | 是 | 队列ID |
| priority | Number | 是 | 坐席优先级 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

可以多次调用，开启对多个队列的服务。

开始服务成功后，如果没有开启免打扰，那么有客户在此队列排队时系统会自动分配客户：[CRVideo_AutoAssignUser.callback](#)；如果开启免打扰，系统就不会自动分配客户，如果需要手动分配客户，可调[用 CRVideo_ReqAssignUser](#)。

坐席优先级描述：

- 1.坐席优先级只在服务的队列内部有效；
- 2.客户优先分配给服务此队列优先级最高的，且空闲的座席；
- 3.优先级相同时，则分配给最先空闲的座席；
- 4.优先级高的座席变空闲时，不抢夺已分配的客户；

CRVideo_StopService(queID,cookie)

- **描述：** 坐席停止服务某个队列
- **返回值：** 无
- **回调：** [CRVideo_StopServiceRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| queID | Number | 是 | 队列ID |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ReqAssignUser(cookie)

- **描述：** 坐席请求分配一个客户
- **返回值：** 无
- **回调：** [CRVideo_ReqAssignUserRslt.callback](#)

• 参数：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

当坐席关闭免打扰时，系统将自动分配客户，无需调用此函数；当开启免打扰时，系统不再自动分配客户，座席如需服务客户可调用此接口请求系统分配。

CRVideo_AcceptAssignUser(queID,UID,cookie)

- 描述：坐席接受系统分配的客户
- 返回值：无
- 回调：[CRVideo_ResponseAssignUserRslt.callback](#)
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| queID | Number | 是 | 队列ID |
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_RejectAssignUser(queID,UID,cookie)

- 描述：坐席拒绝系统分配的客户
- 返回值：无
- 回调：[CRVideo_ResponseAssignUserRslt.callback](#)
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| queID | Number | 是 | 队列ID |
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_InitQueueDatRslt.callback = function(sdErr,cookie){}

- 描述：回调，队列初始化操作的结果
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_QueueStatusChanged.callback = function(QueStatus){}

- 描述：通知，队列状态变化

- 回调参数：

| 参数 | 类型 | 含义 |
|-----------|--------|------------------------------------------------|
| QueStatus | Object | 用户队列状态，详见： CRVideo_QueueStatus |

当用户服务某个队列，相关队列状态变化时，会收到此通知。

CRVideo_GetQueueStatusRslt.callback = function(QueStatus,sdkErr,cookie){}

- 描述：回调，获取某个队列状态的结果
- 回调参数：

| 参数 | 类型 | 含义 |
|-----------|--------|------------------------------------------------|
| QueStatus | Object | 用户队列状态，详见： CRVideo_QueueStatus |
| sdkErr | Number | 错误码，详细见 错误码定义 |

CRVideo_QueueingInfoChanged.callback = function(CRVideo_QueueingInfo){}

- 描述：通知，排队信息变化（客户）
- 回调参数：

| 参数 | 类型 | 含义 |
|----------------------|--------|-------------------------------------------|
| CRVideo_QueueingInfo | Object | 排队信息 CRVideo_QueueingInfo |

CRVideo_StartQueueingRslt.callback = function(sdkErr,cookie){}

- 描述：回调，开始排队的结果（客户）
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详细见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StopQueueingRslt.callback = function(sdkErr,cookie){}

- 描述：回调，停止排队的结果（客户）
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详细见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StartServiceRslt.callback = function(queID,sdkErr,cookie){}

- **描述：** 回调，开始服务某个队列的结果（坐席）
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| queID | Number | 队列ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StopServiceRslt.callback = function(queID,sdkErr,cookie){}

- **描述：** 回调，停止服务某个队列的结果（坐席）
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| queID | Number | 队列ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ResponseAssignUserRslt.callback = function(sdkErr,cookie){}

- **描述：** 回调，响应系统分配客户的结果（坐席）
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AutoAssignUser.callback = function(CRVideo_QueueUser){}

- **描述：** 通知，系统自动分配了客户
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------------------|--------|------------------------------------|
| CRVideo_QueueUser | Object | 队列用户信息, 详细介绍见 类型定义 |

如果想停止系统的自动分配，请调用 [CRVideo_SetDNDStatus](#) 设置为免打扰状态

CRVideo_ReqAssignUserRslt.callback = function(sdkErr,CRVideo_QueueUser,cookie){}

- **描述：** 回调，请求系统分配客户的结果（坐席）
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------------------|--------|------------------------------------|
| sdkErr | Number | 错误码，详细见 错误码定义 |
| CRVideo_QueueUser | Object | 队列用户信息, 详细介绍见 类型定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_CancelAssignUser.callback = function(queID,UID){}

- **描述：**通知，系统取消分配客户（坐席）
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------|--------|---------------------------------|
| queID | Number | 队列ID |
| UID | String | 用户ID, 详细介绍见 关键词 |

系统通过接口 [CRVideo_AutoAssignUser](#) 给队列服务者自动推送客户（或队列服务者通过 [CRVideo_ReqAssignUser](#) 请求分配客户），收到系统分配的客户后，如果队列服务者30秒之内不接受 ([CRVideo_AcceptAssignUser](#)) 也不拒绝 ([CRVideo_RejectAssignUser](#)) 推送的客户，系统将通过本接口取消分配，然后将客户分配给其他空闲的服务者。

CRVideo_Call(calledUID,MeetInfoObj,userExtDat,cookie)

- **描述：**发起呼叫
- **返回值：**无
- **回调：**呼叫成功回调 [CRVideo_CallSuccess.callback](#)，呼叫失败回调 [CRVideo_CallFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------------|--------|----|------------------------------------------------|
| calledUID | String | 是 | 被叫用户的 UID |
| MeetInfoObj | Object | 是 | 房间信息对象，详见： CRVideo_RoomInfoObj |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

房间信息对象可以在创建房间成功的回调接口 [CRVideo_CreateMeetingSuccess.callback](#) 获得
userExtDat参数为自定义参数，对方在收到呼叫时可以拿到此参数的数据

CRVideo_AcceptCall(callID,MeetInfoObj,userExtDat,cookie)

- **描述：**接受对方发起的呼叫
- **返回值：**无
- **回调：**成功回调 [CRVideo_AcceptCallSuccess.callback](#)，失败回调 [CRVideo_AcceptCallFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------|
| callID | String | 是 | 呼叫全局标识ID |

| | | | |
|-------------|--------|---|------------------------------------------------|
| MeetInfoObj | Object | 是 | 房间信息对象，详见： CRVideo_RoomInfoObj |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_RejectCall(callID,userExtDat,cookie)

- **描述：**拒绝对方发起的呼叫
- **返回值：**无
- **回调：**成功回调 [CRVideo_RejectCallSuccess.callback](#)，失败回调 [CRVideo_RejectCallFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|----------------------------------|
| callID | String | 是 | 呼叫全局标识ID |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_HangupCall(callID,userExtDat,cookie)

- **描述：**挂断当前的呼叫
- **返回值：**无
- **回调：**成功回调 [CRVideo_HangupCallSuccess.callback](#)，失败回调 [CRVideo_HangupCallFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|----------------------------------|
| callID | String | 是 | 呼叫全局标识ID |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SetDNDStatus(DNDStatus,cookie)

- **描述：**设置免打扰状态
- **返回值：**无
- **回调：**操作成功回调 [CRVideo_SetDNDStatusSuccess.callback](#)，失败回调 [CRVideo_SetDNDStatusFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|----------------------------------|
| DNDStatus | Number | 是 | 0代表关闭免打扰，其它值代表开启免打扰，含义自由定义 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

如果是使用到了SDK的队列功能，则入会后需调用此接口，设置自己的为免打扰状态，防止系统再次推送自己

CRVideo_GetUserStatus(cookie)

- **描述**：获取AppID下所有用户在线状态
- **返回值**：无
- **回调**：成功回调， [CRVideo_GetUserStatusSuccess.callback](#) ，失败回调， [CRVideo_GetUserStatusFail.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetOneUserStatus(cookie)

- **描述**：获取AppID下指定用户的在线状态
- **返回值**：无
- **回调**：成功回调， [CRVideo_GetUserStatusSuccess.callback](#) ，失败回调， [CRVideo_GetUserStatusFail.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StartUserStatusNotify(cookie)

- **描述**：开启用户的状态推送
- **返回值**：无
- **回调**：[CRVideo_StartUserStatusNotifyRslt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

AppID下所有用户状态有变化时（包括在线状态、空闲状态、免打扰状态），都会收到 [CRVideo_NotifyUserStatus.callback](#) 通知。开启后，用户量越大消息量越大，所以请按需开启。在 [CRVideo_StartUserStatusNotify](#) 前，应该先通过 [CRVideo_GetUserStatus](#) 获取所有用户状态。

CRVideo_StopUserStatusNotify(cookie)

- **描述**：关闭用户的状态推送
- **返回值**：无
- **回调**：[CRVideo_StopUserStatusNotifyRslt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_CallSuccess.callback = function(callID,cookie){}

- **描述**：回调，发起呼叫成功

- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_CallFail.callback = function(callID,sdkErr,cookie){}

- 描述：回调，发起呼叫失败
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AcceptCallSuccess.callback = function(callID,cookie){}

- 描述：回调，接受他人的呼叫成功
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AcceptCallFail.callback = function(callID,sdkErr,cookie){}

- 描述：回调，接受他人的呼叫失败
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_RejectCallSuccess.callback = function(callID,cookie){}

- 描述：回调，拒绝他人的呼叫成功
- 回调参数：

| 参数 | 类型 | 含义 |
|----|----|----|
|----|----|----|

| | | |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_RejectCallFail.callback = function(callID,sdkErr,cookie){}

- **描述：** 回调，拒绝他人的呼叫失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_HangupCallSuccess.callback = function(callID,cookie){}

- **描述：** 回调，挂断呼叫成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_HangupCallFail.callback = function(callID,sdkErr,cookie){}

- **描述：** 回调，挂断呼叫失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| callID | String | 呼叫全局标识ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_NotifyCallIn.callback = function(callID,MeetInfoObj,callerID,userExtDat){}

- **描述：** 通知，有人呼叫自己
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------|
| callID | String | 呼叫全局标识ID |

| | | |
|-------------|--------|------------------------------------------------|
| MeetInfoObj | Object | 房间信息对象，详见： CRVideo_RoomInfoObj |
| callerID | String | 对方呼叫人员的标识ID |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_NotifyCallAccepted.callback = function(callID,MeetInfoObj,userExtDat){}

- **描述：**通知，呼叫被对方接受
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------------|--------|------------------------------------------------|
| callID | String | 呼叫全局标识ID |
| MeetInfoObj | Object | 房间信息对象，详见： CRVideo_RoomInfoObj |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_NotifyCallRejected.callback = function(callID,sdkErr,userExtDat){}

- **描述：**通知，呼叫被对方拒绝
- **回调参数：**

| 参数 | 类型 | 含义 |
|------------|--------|-------------------------------|
| callID | String | 呼叫全局标识ID |
| sdkErr | Number | 错误码，详细见 错误码定义 |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_NotifyCallHangup.callback = function(callID,userExtDat){}

- **描述：**通知，呼叫被对方挂断
- **回调参数：**

| 参数 | 类型 | 含义 |
|------------|--------|----------------|
| callID | String | 呼叫全局标识ID |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_SetDNDDStatusSuccess.callback = function(cookie){}

- **描述：**回调，设置免打扰状态成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|----|----|----|
| | | |

| | | |
|--------|--------|----------------------------------|
| cookie | String | 详细介绍见 关键词 ，可以传空串 |
|--------|--------|----------------------------------|

CRVideo_SetDNDStatusFail.callback = function(sdkErr,cookie){}

- **描述：** 回调，设置免打扰状态失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetUserStatusSuccess.callback = function(userStatusArr,cookie){}

- **描述：** 回调，获取用户在线状态成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------------|--------|--------------------------------------------------------|
| userStatusArr | Array | 用户在线状态信息列表，[CRVideo_UserStatus , ...] |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetUserStatusFail.callback = function(sdkErr,cookie){}

- **描述：** 回调，获取用户在线状态失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StartUserStatusNotifyRslt.callback = function(sdkErr,cookie){}

- **描述：** 回调，开启用户状态推送的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_StopUserStatusNotifyRslt.callback = function(sdkErr,cookie){}

- **描述：** 回调，停止用户状态推送的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_NotifyUserStatus.callback = function(CRVideo_UserStatus){}

- **描述：**通知，用户的状态变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------------------|--------|-----------------------------------------------------|
| CRVideo_UserStatus | Object | 单个用户在线状态信息，类型定义： CRVideo_UserStatus |

CRVideo_Invite(invitedUserID,userExtDat,cookie)

- **描述：**发送邀请
- **返回值：**返回inviteID(每个邀请对应的唯一ID) (string)
- **回调：**成功回调 [CRVideo_InviteSuccess.callback](#)，失败回调 [CRVideo_InviteFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------------|--------|----|----------------------------------|
| invitedUserID | String | 是 | 受邀者用户的 UID |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

需要邀请对方加入房间时，userExtDat可以是房间信息 [CRVideo_RoomInfoObj](#)

CRVideo_AcceptInvite(inviteID,userExtDat,cookie)

- **描述：**接受对方发送的邀请
- **返回值：**无
- **回调：**成功回调 [CRVideo_AcceptInviteSuccess.callback](#)，失败回调 [CRVideo_AcceptInviteFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|----------------------------------|
| inviteID | String | 是 | 邀请ID，在收到邀请时获取 |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

收到邀请时，需要及时接受邀请或拒绝邀请，超时回应服务端将会自动拒绝本地邀请

CRVideo_RejectInvite(inviteID,userExtDat,cookie)

- **描述：**拒绝对方发送的邀请
- **返回值：**无

- **回调：** 成功回调 [CRVideo_RejectInviteSuccess.callback](#)，失败回调 [CRVideo_RejectInviteFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|----------------------------------|
| inviteID | String | 是 | 邀请ID，在收到邀请时获取 |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

收到邀请时，需要及时接受邀请或拒绝邀请，超时回应服务端将会自动拒绝本地邀请

CRVideo_CancelInvite(inviteID,userExtDat,cookie)

- **描述：** 取消自己发送的邀请
- **返回值：** 无
- **回调：** 成功回调 [CRVideo_CancelInviteSuccess.callback](#)，失败回调 [CRVideo_CancelInviteFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|----------------------------------|
| inviteID | String | 是 | 邀请ID，在发送邀请时接口返回 |
| userExtDat | String | 否 | 自定义扩展参数，没有则传空串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_InviteSuccess.callback = function(inviteID,cookie){}

- **描述：** 回调，发送邀请成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_InviteFail.callback = function(inviteID, sdkErr, cookie){}

- **描述：** 回调，发送邀请失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AcceptInviteSuccess.callback = function(inviteID,cookie){}

- **描述：** 回调，接受对方发送的邀请成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AcceptInviteFail.callback = function(inviteID,sdkErr,cookie){}

- **描述：** 回调，接受对方发送的邀请失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_RejectInviteSuccess.callback = function(inviteID,cookie){}

- **描述：** 回调，拒绝对方发送的邀请成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_RejectInviteFail.callback = function(inviteID,sdkErr,cookie){}

- **描述：** 回调，拒绝对方发送的邀请失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_CancelInviteSuccess.callback = function(inviteID,cookie){}

- **描述：** 回调，取消自己发送的邀请成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|----|----|----|
|----|----|----|

| | | |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_CancelInviteFail.callback = function(inviteID,sdkErr,cookie){}

- **描述：** 回调，取消自己发送的邀请失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_NotifyInviteIn.callback = function(inviteID,inviterUserID,userExtDat){}

- **描述：** 通知，收到别人的邀请
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------------|--------|----------------------------|
| inviteID | String | 邀请ID |
| inviterUserID | String | 发送邀请的 用户ID |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_NotifyInviteAccepted.callback = function(inviteID,userExtDat){}

- **描述：** 通知，我发送的邀请被对方接受
- **回调参数：**

| 参数 | 类型 | 含义 |
|------------|--------|----------------|
| inviteID | String | 邀请ID |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_NotifyInviteRejected.callback = function(inviteID,reason,userExtDat){}

- **描述：** 通知，我发送的邀请被对方拒绝
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|------|
| inviteID | String | 邀请ID |

| | | |
|------------|--------|----------------------------------|
| reason | String | 邀请被拒绝的原因，详见： 错误码 |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_NotifyInviteCanceled.callback = function(inviteID,reason,userExtDat){}

- **描述：**通知，邀请已经被取消了
- **回调参数：**

| 参数 | 类型 | 含义 |
|------------|--------|----------------------------------|
| inviteID | String | 邀请ID |
| reason | String | 邀请被取消的原因，详见： 错误码 |
| userExtDat | String | 自定义扩展参数，没有则传空串 |

CRVideo_CreateMeeting(CreateParams,cookie)

- **描述：**创建房间
- **返回值：**无
- **回调：**成功回调 [CRVideo_CreateMeetingSuccess.callback](#)，失败回调 [CRVideo_CreateMeetingFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------------|--------|----|--------------------------------------------------------|
| CreateParams | Object | 否 | 创建房间的参数对象，非必传。详见： CRVideo_CreateParams |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_DestroyMeeting(meetID,cookie)

- **描述：**销毁房间
- **返回值：**无
- **回调：**[CRVideo_DestroyMeetingRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| meetID | Number | 是 | 房间号 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

SDK创建的房间为固定房间，固定房间不会自动销毁，当该房间今后不再使用时，应当调用此接口销毁该房间
调用此接口后，房间内成员将收到 [CRVideo_MeetingStopped.callback](#) 通知

CRVideo_CreateMeetingSuccess.callback = function(MeetInfoObj,cookie){}

- **描述：**回调，创建房间成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------------|--------|------------------------------------------------|
| MeetInfoObj | Object | 房间信息对象，详见： CRVideo_RoomInfoObj |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_CreateMeetingFail.callback = function(sdkErr,cookie){}

- **描述：**回调，创建房间失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_DestroyMeetingRslt.callback = function(sdkErr,cookie){}

- **描述：**回调，销毁房间的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_MeetingStopped.callback = function(){}

- **描述：**通知，当前房间已被结束

CRVideo_EnterMeeting3(meetID,cookie)

- **描述：**进入房间
- **返回值：**无
- **回调：**[CRVideo_EnterMeetingRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------|--------|----|----------------------------------|
| meetID | Number | 是 | 房间号 |
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| nickName | String | 是 | 昵称 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ExitMeeting()

- **描述：**离开房间

- 返回值：无
- 回调：无

CRVideo_Kickout(UID).then(resolveCallback).catch(rejectCallback)

- 描述：将房间内指定成员踢出房间
- 返回值：返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

- resolveCallback 参数：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| userID | String | 是 | 用户UID, 详细介绍见 关键词 |

- rejectCallback 参数：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

示例代码：

```
CRVideo_Kickout(UID).then(resObj =>{
  // 成功
  const errCode = resObj.errCode;
  const userID = resObj.userID;
}).catch(errObj =>{
  // 失败
  const errCode = errObj.errCode;
  const errDesc = errObj.errDesc;
})
```

CRVideo_EnterMeetingRslt.callback = function(sdkErr,cookie){}

- 描述：回调，进入房间的结果
- 回调参数：

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_UserEnterMeeting.callback = function(UID){}()

- **描述：** 通知，有人进入房间
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_UserLeftMeeting.callback = function(UID){}()

- **描述：** 通知，有人离开房间
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_MeetingDropped.callback = function(){}()

- **描述：** 通知，从房间中掉线了

当收到此通知后，如果要继续会话，需再次调用 [CRVideo_EnterMeeting3](#) 接口重新进入房间，并可通过传cookie参数在进入房间成功的回调中来区分是第一次进入房间还是重新进入房间，以此来不同的处理，详见Demo。

如果是Token鉴权登录，当Token过期时，也会强制从房间中退出并收到此通知，此时应先更新Token，再重新进入房间。

CRVideo_GetMainVideo()

- **描述：** 获取当前哪个用户为主视频
- **返回值：** 当前主视频用户ID
- **回调：** 无

CRVideo_SetMainVideo(UID)

- **描述：** 设置哪个用户为主视频
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_SetMainVideoMode(mode,UID)

- **描述：** 设置主视频模式
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----|----|----|----|
| | | | |

| | | | |
|------|--------|---|--------------------------------------------------|
| mode | Number | 是 | 主视频模式，详见 CRVideo_MAIN_VIDEO_MODE |
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_SwitchToPage(mainPage,subPage).then(resolveCallback).catch(rejectCallback)

- **描述：** 功能区切换
- **返回值：** 返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------|--------|----|-------------------------------------------------|
| mainPage | Number | 是 | 功能区类型，详见 CRVideo_MAIN_PAGE_TYPE |
| subPage | String | 否 | 子页面标识（如创建白板时返回的boardID："2.58"），非必传 |

- **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |

- **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

CRVideo_GetCurrentMainPage()

- **描述：** 获取当前主功能区
- **返回值：** 返回当前的主功能区。功能区类型详见 [CRVideo_MAIN_PAGE_TYPE](#)

CRVideo_GetCurrentSubPage()

- **描述：** 获取当前子页面
返回当前的子页面标识，如白板的boardID。没有子页面标识的功能区，则返回
- **返回值：** 0.0

CRVideo_NotifyMainVideoChanged.callback = function(UID){}

- **描述：** 通知，会话内主视频变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_NotifyMainVideoModeChanged.callback = function(mode){}

- **描述：** 通知，会话内主视频模式变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|------|--------|--------------------------------------------------|
| mode | Number | 主视频模式，详见 CRVideo_MAIN_VIDEO_MODE |

CRVideo_NotifySwitchToPage.callback = function(mainPage,subPage){}

- **描述：** 通知，会话内功能区变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|---------------------------------------------------------|
| mainPage | Number | 功能区类型，详见 CRVideo_MAIN_PAGE_TYPE |
| subPage | String | 子页面标识（如创建白板时返回的boardID："2.58"），非必传。没有子页面标识的功能区，则为 "0.0" |

CRVideo_GetMeetingAttrs(keys, cookie)

- **描述：** 获取房间部份属性
- **返回值：** 无
- **回调：** 成功回调 [CRVideo_GetMeetingAttrsSuccess.callback](#)，失败回调 [CRVideo_GetMeetingAttrsFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| keys | Array | 是 | 需要查询的房间属性key，如：["key1", "key2"] |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetMeetingAllAttrs(cookie)

- **描述：** 获取房间所有属性
- **返回值：** 无
- **回调：** 成功回调 [CRVideo_GetMeetingAllAttrsSuccess.callback](#)，失败回调 [CRVideo_GetMeetingAllAttrsFail.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SetMeetingAttrs(attrs ,options ,cookie)

- **描述：** 重置房间属性集
- **返回值：** 无
- **回调：** 操作完成则回调 [CRVideo_SetMeetingAttrsRsIt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------------------------------------------------------|
| attrs | String | 是 | 房间属性集,json格式, 如: {"key1":"value1", "key2":"value2"} (key最大长度为64B, value最大长度为8KB) |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 , 可以传空串 |

CRVideo_AddOrUpdateMeetingAttrs(attrs ,options ,cookie)

- **描述**：增加或者更新房间属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_AddOrUpdateMeetingAttrsRslt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------------------------------------------------------|
| attrs | String | 是 | 房间属性集,json格式, 如: {"key1":"value1", "key2":"value2"} (key最大长度为64B, value最大长度为8KB) |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 , 可以传空串 |

CRVideo_DelMeetingAttrs(keys ,options ,cookie)

- **描述**：删除房间属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_DelMeetingAttrsRslt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------------------------------|
| keys | Array | 是 | 需要删除的房间属性key, 如: ["key1", "key2"] |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 , 可以传空串 |

CRVideo_ClearMeetingAttrs(options ,cookie)

- **描述**：清除房间所有属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_ClearMeetingAttrsRslt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------------------------------|
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 , 可以传空串 |

CRVideo_GetUserAttrs(uids ,keys ,cookie)

- **描述**：获取指定用户的指定属性
- **返回值**：无
- **回调**：成功回调 [CRVideo_GetUserAttrsSuccess.callback](#) ，失败回调 [CRVideo_GetUserAttrsFail.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------------------------|
| uids | Array | 是 | 目标用户id列表，一次最多包含50个用， 如：["uid1","uid2"] |
| keys | Array | 否 | 将要获取的用户属性key列表（空数组或者不传表获取全部），如： ["key1","key2"] |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SetUserAttrs(UID ,attrs ,options ,cookie)

- **描述**：重置用户属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_SetUserAttrsRsIt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|--------------------------------------------------------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| attrs | String | 是 | 用户属性集, json格式，如：{"key1":"value1", "key2":"value2"} (key最大长度为64B，value最大长度为8KB) |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AddOrUpdateUserAttrs(UID ,attrs ,options ,cookie)

- **描述**：增加或更新用户属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_AddOrUpdateUserAttrsRsIt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|--------------------------------------------------------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| attrs | String | 是 | 用户属性集, json格式，如：{"key1":"value1", "key2":"value2"} (key最大长度为64B，value最大长度为8KB) |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_DelUserAttrs(UID ,keys ,options ,cookie)

- **描述**：删除用户的特定属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_DelUserAttrsRsIt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| keys | Array | 是 | 需要删除的用户属性key列表，json格式，如：["key1","key2"] |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ClearUserAttrs(uids ,options ,cookie)

- **描述**：清除特定用户的所有属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_ClearUserAttrsRsIt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|----------------|----|----------------------------------------------------------|
| uids | Array String | 是 | 目标用户UID或UID数组 |
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ClearAllUserAttrs(options ,cookie)

- **描述**：清除所有用户的所有属性
- **返回值**：无
- **回调**：操作完成则回调 [CRVideo_ClearAllUserAttrsRsIt.callback](#)
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------------------------------|
| options | String | 否 | 操作选项,json格式参见 CRVideo_MeetingAttrOptions |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetMeetingAttrsSuccess.callback = function(attr, cookie){}

- **描述**：回调，成功获取到房间指定属性
- **回调参数**：

| 参数 | 类型 | 含义 |
|----|----|----|
|----|----|----|

| | | |
|--------|--------|-----------------------------------------------------|
| attrs | Object | 房间属性，结构体请参见 CRVideo_MeetingAttrObjs |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetMeetingAttrsFail.callback = function(sdkErr, cookie){}

- **描述：**回调，获取房间指定属性失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetMeetingAllAttrsSuccess.callback = function(attr, cookie){}

- **描述：**回调，成功获取到房间所有属性
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|-----------------------------------------------------|
| attrs | Object | 房间属性，结构体请参见 CRVideo_MeetingAttrObjs |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetMeetingAllAttrsFail.callback = function(sdkErr, cookie){}

- **描述：**回调，获取房间属性失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SetMeetingAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：**回调，设置用户属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AddOrUpdateMeetingAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，增加或更新房间属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_DelMeetingAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，删除房间属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ClearMeetingAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，清除房间属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetUserAttrsSuccess.callback = function(attrsMap, cookie){}

- **描述：** 回调，获取房间内用户属性成功
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|--------------------------------------------------------|
| attrsMap | Object | 用户属性，结构体请参见 CRVideo_UsrMeetingAttrObjs |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_GetUserAttrsFail.callback = function(sdkErr, cookie){}

- **描述：** 回调，获取房间内用户属性失败
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SetUserAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，设置用户属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_AddOrUpdateUserAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，增加或者更新用户属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_DelUserAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，删除特定用户属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ClearUserAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，清除特定用户属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_ClearAllUserAttrsRslt.callback = function(sdkErr, cookie){}

- **描述：** 回调，清除所有用户属性结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|----|----|----|
| | | |

| | | |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_NotifyMeetingAttrsChanged.callback = function(adds ,updates ,delKeys){}

- **描述：** 通知，房间属性发生改变
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------|--------|---------------------------------------------------------|
| adds | Object | 增加房间属性集，结构体请参见 CRVideo_MeetingAttrObjs |
| updates | Object | 变化的房间属性集，结构体请参见 CRVideo_MeetingAttrObjs |
| delKeys | Array | 被删除的房间属性的数组列表，如:["key1", "key2"] |

CRVideo_NotifyUserAttrsChanged.callback = function(UID ,adds ,updates ,delKeys){}

- **描述：** 通知，某个用户属性发生改变
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------|--------|---------------------------------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| adds | Object | 增加用户属性集，结构体请参见 CRVideo_MeetingAttrObjs |
| updates | Object | 变化的用户属性集，结构体请参见 CRVideo_MeetingAttrObjs |
| delKeys | Array | 被删除的用户属性列表，json格式，如:["key1", "key2"] |

CRVideo_GetAllMembers()

- **描述：** 获取房间内所有成员的信息
- **返回值：** 返回包含多个成员信息 [CRVideo_MemberInfo](#) 的列表数组：[memberInfo1, memberInfo2, ...]
- **回调：** 无

CRVideo_GetMemberInfo(UID)

- **描述：** 获取房间内指定成员的信息
- **返回值：** 返回房间内指定成员的信息 [CRVideo_MemberInfo](#)
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_GetMemberNickName(UID)

- **描述：** 获取房间内指定成员的昵称
返回该成员的昵称

- **返回值：** (String)

- **回调：** 无

- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_IsUserInMeeting(UID)

- **描述：** 判断某个用户是否在房间中
- **返回值：** 在房间中则返回 true，否则返回 false (Bool)

- **回调：** 无

- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_SetNickName(UID,newName).then(resolveCallback).catch(rejectCallback)

- **描述：** 设置房间内某个成员的昵称
- **返回值：** 返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback

- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| newName | String | 是 | 新昵称 |

- **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------------------------|
| errCode | Number | 是 | 错误码，详细见 错误码定义 |
| userID | String | 是 | 用户UID, 详细介绍见 关键词 |
| newname | String | 是 | 新昵称 |

- **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|-------------------------------|
| errCode | Number | 是 | 错误码，详细见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

CRVideo_NotifyNickNameChanged.callback = function(UID,oldName,newName){}

- **描述：** 通知，房间内某个成员昵称变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| oldName | String | 旧昵称 |
| newName | String | 新昵称 |

CRVideo_GetAudioMicNames()

- **描述：** 获取本地麦克风设备列表
- **返回值：** 返回麦克风设备列表Array，如：[MicInfoObj1, MicInfoObj2, ...]，详见：[CRVideo_MicInfoObj](#)
- **回调：** 无

CRVideo_GetAudioSpkNames()

- **描述：** 获取本地扬声器设备列表
返回扬声器设备列表Array，如：[SpeakerInfoObj1, SpeakerInfoObj2, ...]，详见：
- **返回值：** [CRVideo_SpeakerInfoObj](#)
- **回调：** 无

CRVideo_SetAudioCfg(AudioCfg)

- **描述：** 设置本地音频参数
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------|--------|----|----------------------------------------------------|
| AudioCfg | Object | 是 | 本地音频配置，详细介绍见类型定义： CRVideo_AudioCfg |

CRVideo_GetAudioCfg()

- **描述：** 获取本地音频参数
- **返回值：** 返回本地音频参数对象：[CRVideo_AudioCfg](#)
- **回调：** 无

CRVideo_GetMicEnergy(UID)

- **描述：** 获取指定成员的麦克风声音大小
- **返回值：** 返回该成员当前麦克风声音大小，0-10
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----|----|----|----|
|----|----|----|----|

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

CRVideo_OpenMic(UID)

- **描述：** 打开指定成员的麦克风
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

打开自己的麦克风时，先会进入到AOPENING状态，等服务器处理后会进入AOPEN状态，此时说话才能被采集到。

CRVideo_CloseMic(UID)

- **描述：** 关闭指定成员的麦克风
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

关麦操作是立即生效的，本地会立即停止采集

CRVideo_GetAudioStatus(UID)

- **描述：** 获取指定成员的麦克风状态
- **返回值：** 返回该成员当前的麦克风状态：[CRVideo_ASTATUS](#)
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_SetAllAudioClose()

- **描述：** 关闭房间内所有人麦克风
- **返回值：** 无
- **回调：** 无

CRVideo_PauseSpeaker()

- **描述：** 暂停播放房间内通话声音（静音）
- **返回值：** 无
- **回调：** 无

CRVideo_PlaySpeaker()

- **描述：**恢复播放房间内通话声音
- **返回值：**无
- **回调：**无

CRVideo_AudioStatusChanged.callback = function(UID,oldStatus,newStatus){}

- **描述：**通知，成员麦克风状态变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----------|--------|---------------------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| oldStatus | Number | 旧的麦克风状态，详见： CRVideo_ASTATUS |
| newStatus | Number | 新的麦克风状态，详见： CRVideo_ASTATUS |

CRVideo_MicEnergyUpdate.callback = function(UID,oldLevel,newLevel){}

- **描述：**通知，成员麦克风声音能量变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| oldLevel | Number | 原来的说话声音强度等级，0-10 |
| newLevel | Number | 现在的说话声音强度等级，0-10 |

CRVideo_CreatVideoObj(CreateUIParams)

- **描述：**创建成员视频UI显示对象
- **返回值：**返回成员视频UI显示对象 [CRVideo_VideoObj](#)
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------------|--------|----|-----------------------------------------------|
| CreateUIParams | Object | 否 | 详细介绍见： CRVideo_CreateUIParams |

示例代码：

```
// 创建对象
const user1VideoObj = CRVideo_CRVideo_CreatVideoObj();
// 设置对象属性
const userVideoDOM = user1VideoObj.handler(); // 获取组件DOM
$('#userVideo').append(userVideoDOM); // 将组件DOM放到页面上
const nicknameStyle = {
  left: "10px",
  top: "10px",
  backgroundColor: "rgba(0, 0, 0, .7)",
  padding: "0 10px",
  borderRadius: "15px",
  display: "block"
}
user1VideoObj.setNickNameStyle(nicknameStyle); // 设置昵称样式
user1VideoObj.setVisibleNickName(true); // 设置是否显示昵称
user1VideoObj.setVideo(UID); // 设置显示成员摄像头画面
```

必须创建成员视频UI对象才能在网页上显示成员的摄像头画面

CRVideo_GetVideoCfg()

- **描述**：获取当前视频参数
- **返回值**：返回当前视频配置参数对象，详见：[CRVideo_VideoCfg](#)
- **回调**：无

CRVideo_SetVideoCfg(VideoCfg)

- **描述**：设置视频参数
- **返回值**：无
- **回调**：无
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|----------|--------|----|-----------------------------------------|
| VideoCfg | Object | 是 | 详细介绍见： CRVideo_VideoCfg |

设置视频参数会根据浏览器支持的能力自适应，相关属性不一定能设置成功，超出浏览器能力时浏览器会选择适合的值。例如：设置fps:35，在chrome上不支持，最终结果是30fps，而在safari上最终结果是31fps

CRVideo_GetVideoStatus(UID)

- **描述**：获取指定成员的视频状态
- **返回值**：返回该用户的视频状态，详见：[CRVideo_VSTATUS](#)
- **回调**：无
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_GetDefaultVideo(UID)

- **描述**：获取指定成员的默认摄像头

- **返回值：** 返回该成员的默认摄像头ID（Number），如：1
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_SetDefaultVideo(UID,videoID)

- **描述：** 设置指定成员的默认摄像头
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| videoID | Number | 是 | 摄像头ID |

如果用户只有一个摄像头，可以不设置默认摄像头

如果用户有多个摄像头，但是只想用其中一个，需要把这个摄像头设置为默认摄像头（切换摄像头）

可以通过调用 [CRVideo_GetAllVideoInfo](#) 接口获取用户的摄像头列表

如果调用此接口设置了新的默认摄像头，房间内用户将会收到 [CRVideo_DefVideoChanged.callback](#) 接口通知

CRVideo_GetAllVideoInfo(UID)

- **描述：** 获取指定成员的所有摄像头信息
- **返回值：** 返回该用户摄像头信息列表Array，如：[VideoInfoObj1,VideoInfoObj2,...]，详见：[CRVideo_VideoInfoObj](#)
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_GetOpenedVideoIDs(UID)

- **描述：** 获取指定成员当前打开的摄像头ID列表
返回该用户当前打开的摄像头ID列表集合，如：[1, 2, 3,
- **返回值：** ...]
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_OpenVideo(UID)

- **描述：** 打开指定成员的摄像头

- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

打开指定成员的摄像头，以便本地和远端显示视频图像
调用此接口后，如果获取到对应的视频流，将会收到CRVideo_NotifyVideoStream.callback(#CRVideo_NotifyVideoStream) 接口通知

CRVideo_CloseVideo(UID)

- **描述：**关闭指定成员的摄像头
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_SetEnableMutiVideo(UID,bool)

- **描述：**设置是否启用多摄像头
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| bool | bool | 是 | 是否启用多摄像头 |

CRVideo_GetEnableMutiVideo(UID,bool)

- **描述：**获取当前是否启用多摄像头
- **返回值：**返回当前是否启用了多摄像头
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_AddStreamCam(videoStream,videoName)

- **描述：**添加WebRTC媒体流摄像头
- **返回值：**返回摄像头ID
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------------|-------------|----|-------------------------------------------|
| videoStream | MediaStream | 是 | WebRTC媒体流，详见： MediaStream |
| videoName | String | 是 | 自定义摄像头名称，如：“媒体流摄像头1” |

添加媒体流摄像头后，可以调用 [CRVideo_GetAllVideoInfo](#) 接口来获取该摄像头对应的videoID，打开此摄像头时需要传入videoID

CRVideo_RemoveStreamVideo(videoID)

- **描述：**移除WebRTC媒体流摄像头
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------------|
| videoID | Number | 是 | 需要移除的媒体流摄像头的ID |

CRVideo_AddCanvasVCam(canvas,videoName,options)

- **描述：**添加canvas虚拟摄像头
- **返回值：**返回摄像头ID
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|-------------------------|
| canvas | Object | 是 | canvas element，Dom对象 |
| videoName | String | 是 | 自定义摄像头名称，如：“canvas摄像头1” |
| options | Object | 否 | 保留参数 |

添加canvas摄像头后，可以调用 [CRVideo_GetAllVideoInfo](#) 接口来获取该摄像头对应的videoID，打开此摄像头时需要传入videoID

CRVideo_RemoveCanvasVCam(videoID)

- **描述：**移除canvas摄像头
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|-------------------|
| videoID | Number | 是 | 需要移除的canvas摄像头的ID |

CRVideo_VideoDevChanged.callback = function(UID){}

- **描述：**通知，某个成员的摄像头设备发生了变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|----|----|----|
|----|----|----|

| | | |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
|-----|--------|---------------------------------|

CRVideo_DefVideoChanged.callback = function(UID,videoID){}

- **描述：** 通知，某个成员的默认摄像头发生了变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| videoID | Number | 摄像头ID |

CRVideo_VideoStatusChanged.callback = function(UID,oldStatus,newStatus){}

- **描述：** 通知，某个成员的视频状态发生了变化
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----------|--------|--------------------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| oldStatus | Number | 旧的视频状态，详见： CRVideo_VSTATUS |
| newStatus | Number | 新的视频状态，详见： CRVideo_VSTATUS |

CRVideo_NotifyVideoStream.callback = function(UID,videoStream){}

- **描述：** 通知，获取到某个成员的WebRTC视频流
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------------|-------------|-----------------------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| videoStream | MediaStream | 该成员的WebRTC视频流，详见： MediaStream |

获取到成员的视频流后，可以给界面上的<video>标签设置srcObject属性显示此视频流的画面，一般不建议这样使用，可以直接通过创建该成员的视频UI组件 [CRVideo_CreatVideoObj](#) 来显示他的画面

CRVideo_CreatScreenShareObj(CreatParams)

- **描述：** 创建屏幕共享显示UI组件
- **返回值：** 返回屏幕共享UI显示对象 [CRVideo_ScreenObj](#)
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------------|--------|----|-----------------------------------------------|
| CreateUIParams | Object | 否 | 详细介绍见： CRVideo_CreateUIParams |

CRVideo_SetScreenShareCfg(CRVideo_ScreenCfgObj)

- **描述：**配置屏幕共享参数
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------------------|--------|----|--------------------------------------------------------|
| CRVideo_ScreenCfgObj | Object | 是 | 屏幕共享配置参数对象，详细介绍见： CRVideo_ScreenCfgObj |

CRVideo_StartScreenShare()

- **描述：**开启本地屏幕共享
- **返回值：**无
- **回调：**[CRVideo_StartScreenShareRslt.callback](#)

CRVideo_StopScreenShare()

- **描述：**结束本地屏幕共享
- **返回值：**无
- **回调：**[CRVideo_StopScreenShareRslt.callback](#)

CRVideo_StartScreenMark()

- **描述：**开启房间内屏幕共享标注
- **返回值：**无
- **回调：**[CRVideo_StartScreenMarkRslt.callback](#)

CRVideo_StopScreenMark()

- **描述：**停止房间内屏幕共享标注
- **返回值：**无
- **回调：**[CRVideo_StopScreenMarkRslt.callback](#)

CRVideo_StartScreenShareRslt.callback = function(sdkErr){}

- **描述：**回调，开启屏幕共享的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |

CRVideo_StopScreenShareRslt.callback = function(sdkErr){}

- **描述：**回调，停止屏幕共享的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|----|----|----|
|----|----|----|

| | | |
|--------|--------|------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
|--------|--------|------------------------------|

CRVideo_StartScreenMarkRslt.callback = function(sdkErr){}

- **描述：** 回调，开启房间内屏幕共享标注的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |

CRVideo_StopScreenMarkRslt.callback = function(sdkErr){}

- **描述：** 回调，停止房间内屏幕共享标注的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |

CRVideo_NotifyScreenShareStarted.callback = function(UID){}

- **描述：** 通知，他人开启了屏幕共享
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_NotifyScreenShareStopped.callback = function(UID){}

- **描述：** 通知，他人停止了屏幕共享
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_NotifyScreenMarkStarted.callback = function(UID){}

- **描述：** 通知，房间内开启了屏幕共享标注
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_NotifyScreenMarkStopped.callback = function(UID){}

- **描述：** 通知，房间内停止了屏幕共享标注
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_CreatMediaObj(CreatParams)

- **描述：** 创建影音共享显示UI组件
- **返回值：** 返回影音共享UI显示对象 [CRVideo_MediaObj](#)
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------------|--------|----|-----------------------------------------------|
| CreateUIParams | Object | 否 | 详细介绍见： CRVideo_CreateUIParams |

无论是共享自己的影音还是收看别人共享的影音，都需要创建此对象

CRVideo_StartPlayMedia(mediaObj,file,locPlay,bPauseWhenFinished)

- **描述：** 开始播放影音
- **返回值：** 返回播放状态，0未在播放，1播放中
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------------------|-------------------|----|--------------------------------|
| mediaObj | MediaContainerObj | 是 | 封装好的videoUI对象，即需要播放的视频容器 |
| file | File | 是 | 支持File类型文件也支持同源播放地址URL(String) |
| locPlay | Number | 否 | 是否仅仅本地播放（0：房间内播放，1：本地播放）缺省为0 |
| bPauseWhenFinished | Number | 否 | 是否播放完毕自动暂在再最后一帧（0：否，1：是）缺省为0 |

MediaContainerObj 通过调用 [CRVideo_CreatMediaObj](#) 接口创建

CRVideo_PausePlayMedia(bPause)

- **描述：** 暂停或恢复播放影音
- **返回值：** 无
- **回调：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|------|----|------------------|
| bPause | Bool | 是 | ture为暂停，false为播放 |

CRVideo_StopPlayMedia()

- **描述：** 停止播放影音及共享

- 返回值：无
- 回调：无

房间内其他人将会收到 [CRVideo_NotifyMediaStop](#) 通知

CRVideo_SetMediaplaypos(setPTS)

- 描述：设置播放进度
- 返回值：无
- 回调：无
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|--------------|
| setPTS | Number | 是 | 设置播放位置，单位：毫秒 |

CRVideo_SetMediaVolume(level)

- 描述：设置本地媒体音量
- 返回值：无
- 回调：无
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|------------|
| level | Number | 是 | 本地音量 0-100 |

CRVideo_Getmediainfo(UID)

- 描述：获取影音播放信息
- 返回值：返回影音信息，详见：[CRVideo_MediaInfoObj](#)
- 回调：无
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |

CRVideo_Getmediaimg(UID,format)

- 描述：获取影音图像数据
- 返回值：当前播放的一帧图片
- 回调：无
- 参数：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|---------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| format | String | 否 | 图片格式，缺省为jpg |

CRVideo_NotifyMediaOpened.callback = function(){}

- **描述：** 通知，影音文件已打开

CRVideo_NotifyMediaStart.callback = function(UID){}

- **描述：** 通知，影音开始播放
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_NotifyMediaStop.callback = function(){}

- **描述：** 通知，影音播放停止

CRVideo_NotifyMediaPause.callback = function(UID,status){}

- **描述：** 通知，影音播放暂停/取消暂停
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| status | Number | 是否暂停， 0：播放中 1：暂停中 |

CRVideo_NotifyPlayPosSetted.callback = function(setPTS){}

- **描述：** 通知，影音播放位置更新
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|--------------|
| setPTS | Number | 新的播放位置，单位：毫秒 |

CRVideo_Creatboard(title,width,height,pageCount).then(resolveCallback).catch(rejectCallback)

- **描述：** 创建白板
- **返回值：** 返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|-------------------------|
| title | String | 是 | 白板名称 |
| width | Number | 是 | 白板宽度 |
| height | Number | 是 | 白板高度 |
| pageCount | Number | 是 | 白板页数（一般空白板1页，文档白板为实际页数） |

- **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| boardID | String | 是 | 白板标识，创建白板时返回，如："2.58" |

• **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

CRVideo_Closeboard(boardID).then(resolveCallback).catch(rejectCallback)

- **描述：** 关闭白板
- **返回值：** 返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|-----------------------|
| boardID | String | 是 | 白板标识，创建白板时返回，如："2.58" |

• **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |

• **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

CRVideo_SetBoardCurPageNo(boardID,pageNo).then(resolveCallback).catch(rejectCallback)

- **描述：** 白板翻页
- **返回值：** 返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|-----------------------|
| boardID | String | 是 | 白板标识，创建白板时返回，如："2.58" |
| pageNo | Number | 是 | 白板页码 |

• **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |

- **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

CRVideo_CreateBoardUIObj(jsonParams)

- **描述：** 白板翻页
- **返回值：** [CRVideo_BoardUIObj](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------------|--------|----|----------|
| jsonParams | Object | 否 | 保留参数，非必传 |

CRVideo_NotifyInitBoards.callback = function(CRVideo_BoardObjs){}

- **描述：** 通知，初始化白板列表
- **回调参数：**

| 参数 | 类型 | 含义 |
|-------------------|-------|-------------------------------------------------------------------------|
| CRVideo_BoardObjs | Array | 白板对象的集合，详见： [CRVideo_BoardObj , CRVideo_BoardObj, ...] |

CRVideo_NotifyCreateBoard.callback = function(CRVideo_BoardObj,UID){}

- **描述：** 通知，创建了新的白板
- **回调参数：**

| 参数 | 类型 | 含义 |
|------------------|--------|-------------------------------------------|
| CRVideo_BoardObj | Object | 白板对象，详见： CRVideo_BoardObj |
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_NotifyCloseBoard.callback = function(boardID,UID){}

- **描述：** 通知，关闭某个白板
- **回调参数：**

| 参数 | 类型 | 含义 |
|---------|--------|-----------------------|
| boardID | String | 白板标识，创建白板时返回，如："2.58" |

| | | |
|-----|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
|-----|--------|---------------------------------|

CRVideo_NotifyBoardCurPageNo.callback = function(boardID,curPage,pagePos1,pagePos2,UID){}

- **描述：** 通知，白板翻页
- **回调参数：**

| 参数 | 类型 | 含义 |
|----------|--------|---------------------------------|
| boardID | String | 白板标识，创建白板时返回，如："2.58" |
| curPage | Number | 目标页码 |
| pagePos1 | Number | SDK保留参数 |
| pagePos2 | Number | SDK保留参数 |
| UID | String | 用户ID, 详细介绍见 关键词 |

CRVideo_StartSvrMixer(Cfgs,Contents,Outputs).then(resolveCallback).catch(rejectCallback)

- **描述：** 开始云端录制、云端直播推流
- **返回值：** 返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------|-------|----|----------------------------------------------------------------|
| Cfgs | Array | 是 | 服务器混图配置列表，如：[CRVideo_MutiMixerCfg , ...] |
| Contents | Array | 是 | 服务器混图内容列表，如：[CRVideo_MutiMixerContent , ...] |
| Outputs | Array | 是 | 服务器混图输出列表，如：[CRVideo_MutiMixerOutput , ...] |

- **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |

- **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

示例代码：

```
// 正确录制参数格式示例：  
// 以下参数同时包含1.音视频和2.纯音频两个混图器，若只需要录制音视频，则应去掉id为2的混图器；
```


// MixerCfgs (多个混图器对象的录制配置参数列表集合, id要和MixerContents及MixerOutputs中的id——对应) :

```
[[
  "id": 1, // 混图器1的录制参数配置
  "streamTypes": 3, // 混图器类型为音视频
  "cfg": {
    "width": 1280,
    "height": 720,
    "frameRate": 15,
    "bitRate": 2000000,
    "defaultQP": 15
  }
}, {
  "id": 2, // 混图器2的录制参数配置
  "streamTypes": 1, // 混图器类型为纯音频
  "aCfg": {
    "channelType": 1,
    "audioFormat": 1
  }
}]
```

// MixerContents (多个混图器对象的录制内容列表集合, id要和MixerCfgs及MixerOutputs中的id——对应) :

```
[[
  "id": 1, // 混图器1的录制内容
  "content": [
    {
      "type": 0, // 录一个摄像头的画面
      "left": 0,
      "top": 0,
      "width": 1280,
      "height": 720,
      "param": {
        "camid": "张三.1"
      },
      "keepAspectRatio": 1
    }, {
      "type": 4, // 录一个时间戳
      "left": 35,
      "top": 3,
      "width": 175,
      "height": 32,
      "keepAspectRatio": 1
    }, {
      "type": 7, // 录一个纯文本
      "left": 40,
      "top": 30,
      "param": {
        "text": "<span style='font-size:14px;color:#e21918'>张三</span>"
      }
    }
  ]
}, {
  "id": 2, // 混图2的录制内容
  "content": [
    {
      "type": 9, // 录张三的语音
      "param": "张三"
    }
  ]
}]
```

// MixerOutputs (多个混图器对象的录制输出列表集合, id要和MixerCfgs及MixerContents中的id——对应) :

```
[[
  "id": 1, // 混图器1的录制输出
  "output": [

```

```

"type": 0, // 输出一个录像视频文件
"filename": "/2020-07-21/张三的录像.mp4"
}, {
"type": 1, // 输出一个rtmp协议的直播流
"liveUrl": "rtmp://pub.cloudroom.com/live/xxxx"
}}
}, {
"id": 2, // 混图器2的录制输出
"output": [{
"type": 0, // 输出一个音频文件
"filename": "/2020-07-21/张三的录音.mp3"
}]
}]
}

```

启动云端功能有一定耗时，请关注混图器的状态变化事件 [CRVideo_SvrMixerStateChanged](#) 每个输出有私有的状态变化事件 [CRVideo_SvrMixerOutputInfo](#)

rCfgs、MutiMixerContents、MixerOutputs 三个参数均为数组，数组中的具体项请参考类型定义

当调用此接口后，会收到混图器状态变化通知，若状态一直为启动中，则可能参数不正确或者录制服务异常，请先检查传参格式是否正确（建议用json工具格式化后与文档对比）

CRVideo_UpdateSvrMixerContent(MixerContents).then(resolveCallback).catch(rejectCallback)

- **描述：**更新云端录制、云端直播推流内容
- **返回值：**返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------------|-------|----|-----------------------------------------------------------------|
| MixerContents | Array | 是 | 服务器混图内容列表，如： [CRVideo_MutiMixerContent , ...] |

- **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |

- **rejectCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

MixerContents 参数格式与 [CRVideo_StartSvrMixer](#) 时的 MixerContents 参数格式一致

CRVideo_StopSvrMixer().then(resolveCallback).catch(rejectCallback)

- **描述：**停止云端录制、云端直播推流
- **返回值：**返回一个Promise对象，成功返回resolveCallback，失败返回rejectCallback
- **resolveCallback 参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |

- rejectCallback 参数：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------------------------------|
| errCode | Number | 是 | 错误码，详见 错误码定义 |
| errDesc | String | 是 | 错误详情 |

停止有有一定耗时，请关注混图器的状态变化事件 [CRVideo_SvrMixerStateChanged](#) 每个输出有私有的状态变化事件 [CRVideo_SvrMixerOutputInfo](#)

CRVideo_GetSvrMixerState()

- 描述：** 获取云端录制、云端直播推流状态
- 返回值：** 返回当前云端录制、云端直播推流状态（Number），详见：[CRVideo_MIXER_STATE](#)
- 回调：** 无

CRVideo_SvrMixerStateChanged.callback = function(MIXER_STATE,sdkErr,UID){}

- 描述：** 通知，云端录制、云端直播推流状态变化
- 回调参数：**

| 参数 | 类型 | 含义 |
|-------------|--------|---------------------------------------------------------|
| MIXER_STATE | Number | 当前云端录制、云端直播推流状态，详见： CRVideo_MIXER_STATE |

CRVideo_SvrMixerOutputInfo.callback = function(MixerOutputInfo){}

- 描述：** 通知，云端录制文件、云端直播推流信息变化
- 回调参数：**

| 参数 | 类型 | 含义 |
|-----------------|--------|------------------------------------------------------------------|
| MixerOutputInfo | Object | 云端录制文件、云端直播推流输出信息，详见： CRVideo_MixerOutputInfoObj |

云端录制开启成功并不代表本次录制就一定是成功的，开启云端录制之后，需关注此通知。
服务器上一次完整的云端录制包括：启动录制、结束录制、生成录像文件、上传录像文件等过程，最终收到此通知，并且 `state` 为7时，则表示此次录制全部完成

VideoObj.id(ID)

- 描述：** 设置/查询组件的独立ID
- 返回值：** 不传参时返回改组件的独立ID，默认为空
- 参数：**

| 参数 | 类型 | 必传 | 含义 |
|----|--------|----|--------|
| ID | String | 否 | 组件独立ID |

非必须调用的方法。不调用此方法设置时，组件没有独立ID，所有调用 [CRVideo_CreatVideoObj](#) 方法创建的组件独立ID均为""

VideoObj.height(height)

- 描述：** 设置/查询组件的高度

- **返回值**：不传参时返回组件的高度，单位px
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|-----------|
| height | String | 否 | 组件高度，单位px |

示例代码：

```
VideoObj.height('720px');
// 或者
const domHeight = VideoObj.height();
```

非必须调用的方法。不调用此方法设置时，默认为父元素高度的100%

VideoObj.width(width)

- **描述**：设置/查询组件的宽度
- **返回值**：不传参时返回组件的宽度，单位px
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|-----------|
| width | String | 否 | 组件宽度，单位px |

示例代码：

```
VideoObj.width('1280px');
// 或者
const domWidth = VideoObj.width();
```

非必须调用的方法。不调用此方法设置时，默认为父元素宽度的100%

VideoObj.left(left)

- **描述**：设置/查询组件在父元素中的left位置
- **返回值**：不传参时返回组件在父元素中的left位置，单位px
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|------|--------|----|---------------------|
| left | String | 否 | 组件在父元素中的left位置，单位px |

示例代码：

```
VideoObj.left('100px');
// 或者
const domLeftPos = VideoObj.left();
```

非必须调用的方法。不调用此方法设置时，默认为0

VideoObj.top(top)

- **描述**：设置/查询组件在父元素中的top位置

- **返回值：** 不传参时返回组件在父元素中的top位置，单位px
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----|--------|----|--------------------|
| top | String | 否 | 组件在父元素中的top位置，单位px |

示例代码：

```
VideoObj.top('100px');
// 或者
const domTopPos = VideoObj.top();
```

非必须调用的方法。不调用此方法设置时，默认为0

VideoObj.show()

- **描述：** 显示组件
- **返回值：** 无

组件默认是显示状态，在组件隐藏后调用此方法重新显示组件。SDK此时会重新订阅该组件所对应的视频流

VideoObj.hide()

- **描述：** 隐藏组件
- **返回值：** 无

组件默认是显示状态，调用此方法隐藏组件时，SDK会退订组件对应的视频流以节省带宽

VideoObj.handler()

- **描述：** 获取该组件的DOM对象
- **返回值：** 返回该组件的DOM对象

示例代码：

```
$('#userVideo').append(CRVideo_VideoObj.handler());
// 或者
$('#userScreen').append(CRVideo_ScreenShareObj.handler());
```

VideoObj.setVideo(UID,camID)

- **描述：** 设置组件上要显示的视频画面，订阅视频
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|-----------------------------------------------------------------|
| UID | String | 是 | 用户ID, 详细介绍见 关键词 |
| camID | Number | 否 | 要显示画面的摄像头ID，-1为默认摄像头。可以不传，缺省为-1（仅组件类型为CRVideo_VideoObj时才需要传此参数） |

示例代码：

```
VideoObj.setVideo('Victor');
```

组件类型为 CRVideo_VideoObj 时，可以传camID参数，为其它类型时不需要此参数

VideoObj.clear()

- **描述**：清除组件中的图像
- **返回值**：无

若需要再次显示图像，需重新调用 setVideo() 方法

VideoObj.isPicEmpty()

- **描述**：检查组件当前图像是否为空
- **返回值**：返回组件当前图像是否为空，0为空，1为非空

VideoObj.setNickNameStyle(style)

- **描述**：设置组件上昵称的显示样式
- **返回值**：无
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|-----------|
| style | Object | 是 | 昵称显示css样式 |

示例代码：

```
const style = {
  left: "10px",
  top: "10px",
  backgroundColor: "rgba(0, 0, 0, .7)",
  padding: "0 10px",
  borderRadius: "15px",
  display: "block"
}
VideoObj.setNickNameStyle(style);
```

VideoObj.setNickNameContent(content)

- **描述**：设置组件上昵称的显示内容
- **返回值**：无
- **参数**：

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|----------|
| content | String | 是 | 要显示的昵称内容 |

示例代码：

```
VideoObj.setNickNameContent("主持人的屏幕");
```

当组件类型为 CRVideo_VideoObj 时，昵称内容默认为该用户的昵称；

当组件类型为 CRVideo_ScreenShareObj 时，昵称内容默认为 xxx's screenShare；
当组件类型为 CRVideo_MeidaObj 时，昵称内容默认为 xxx's media；
若需要显示其他内容，可调用此接口修改

VideoObj.setVisibleNickName(isVisible)

- **描述：** 设置组件上是否显示昵称
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----------|------|----|-----------------------------|
| isVisible | Bool | 是 | 是否显示昵称，true：显示昵称，false：隐藏昵称 |

示例代码：

```
VideoObj.setVisibleNickName(true);
```

VideoObj.getVisibleNickName()

- **描述：** 获取组件当前是否显示昵称
- **返回值：** 返回组件当前是否显示昵称，true：显示昵称，false：隐藏昵称

VideoObj.dblClickFullScreen(bool)

- **描述：** 设置/查询组件是否支持双击全屏
- **返回值：** 不传参时返回组件是否支持双击全屏
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------|------|----|------------------------------|
| bool | Bool | 否 | 是否显示支持双击全屏，true：支持，false：不支持 |

VideoObj.savePic(fileName)

- **描述：** 保存组件当前显示的一帧画面为图片文件
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----------|--------|----|--------------------|
| fileName | String | 是 | 保存的图片文件名，如：123.png |

示例代码：

```
VideoObj.savePic('123.png');
```

调用此接口后，会自动触发浏览器的下载，图片将以下载的方式保存到本地，下载路由用户行为决定，无法指定支持常见的图片格式，如：jpg、jpeg、png、bmp、gif 等
组件上有视频画面时才可以调用此方法，否则SDK不会有任何响应

VideoObi.savePicToBase64(format)

- **描述：**保存组件当前显示的一帧画面为base64数据
- **返回值：**返回当前一帧画面的base64数据
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|-------------|
| format | String | 否 | 图片格式，缺省为png |

示例代码：

```
const picBase64 = VideoObj.savePicToBase64('png');
```

返回的base64数据带有mimeType头，比如："data:image/png;base64,iVBORw0KGgoAAAANSUhhEUgAA..."，如不需要可将"data:image/png;base64," 去掉，只保留后面的数据
支持常见的图片格式，如：jpg、jpeg、png、bmp、gif 等
组件上有视频画面时才可以调用此方法，否则SDK不会有任何响应

CRVideo_VideoObj.setVideo(uid, camID)

- **描述：**订阅成员摄像头画面
- **返回值：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|---------------------------------------|
| uid | String | 是 | 用户ID, 详细介绍见 关键词 |
| camID | Number | 否 | 成员的摄像头ID，非必传。不传或传-1则订阅该成员默认摄像头。传0则为退订 |

若要退订某个UI组件上的视频，则调用：CRVideo_VideoObj.setVideo("",0) 即可
每个组件可以反复订阅不同成员的不同摄像头ID，订阅新的摄像头之后，此前订阅的摄像头则会自动退订
组件setVideo之后，若不退订或不进行销毁，则无论该组件是否显示，都会持续传输其所订阅的摄像头视频流，并占用资源

CRVideo_VideoObj.getVideoID()

- **描述：**查询组件当前订阅的摄像头ID
- **返回值：**返回组件当前订阅的摄像头ID

CRVideo_VideoObj.destroy()

- **描述：**销毁组件，退订视频
- **返回值：**无

调用此方法彻底销毁UI组件，组件销毁后，SDK会退订相关视频流，以节省带宽

CRVideo_ScreenShareObj.setMark(CRVideo_MarkOptions)

- **描述：**设置屏幕共享标注样式
- **返回值：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----|----|----|----|
|----|----|----|----|

| | | | |
|---------------------|--------|---|-------------------------------------------------------|
| CRVideo_MarkOptions | Object | 是 | 屏幕共享标注样式参数对象, 详见： CRVideo_MarkOptions |
|---------------------|--------|---|-------------------------------------------------------|

CRVideo_BoardUIObj.handler()

- **描述：** 获取白板UI组件的Dom对象
- **返回值：** 返回白板UI组件的Dom对象（div）

CRVideo_BoardUIObj.setCurBoardPage(boardID,pageNo)

- **描述：** 设置该组件上要显示的白板页
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|-----------------------|
| boardID | String | 是 | 白板标识，创建白板时返回，如："2.58" |
| pageNo | Number | 是 | 白板页码 |

CRVideo_BoardUIObj.isScaledContents(bool)

- **描述：** 设置（查询）当前组件是否内容允许缩放
- **返回值：** 不传参时返回 true/false
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------|------|----|--------------|
| bool | Bool | 否 | 当前组件是否内容允许缩放 |

CRVideo_BoardUIObj.setTitleStyle(style)

- **描述：** 设置当前组件的标题样式
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|-----------------|
| style | Object | 是 | 当前组件的标题的css样式对象 |

CRVideo_BoardUIObj.setBoardStyle(style)

- **描述：** 设置当前组件的canvas样式
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------|--------|----|---------------------|
| style | Object | 是 | 当前组件的canvas的css样式对象 |

CRVideo_BoardUIObj.setPen(CRVideo_BoardPenOptions)

- **描述：** 设置当前组件的画笔样式
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-------------------------|--------|----|----------------------------------------------------------|
| CRVideo_BoardPenOptions | Object | 是 | 白板UI组件画笔配置对象，详见： CRVideo_BoardPenOptions |

CRVideo_BoardUIObj.undo()

- **描述：** 撤销一笔
- **返回值：** 无

CRVideo_BoardUIObj.redo()

- **描述：** 恢复一笔
- **返回值：** 无

CRVideo_BoardUIObj.clear()

- **描述：** 清空组件
- **返回值：** 无

CRVideo_BoardUIObj.addElement(element)

- **描述：** 添加自定义图元（暂不支持）
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|---------|--------|----|------|
| element | Object | 是 | 图元对象 |

CRVideo_BoardUIObj.removeElement(eIDs)

- **描述：** 移除指定图元（暂不支持）
- **返回值：** 无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|------|-------|----|--------|
| eIDs | Array | 是 | 图元ID集合 |

CRVideo_SendCmd(UID,data,cookie)

- **描述：** 发送点对点消息
- **返回值：** uuid，本次发送任务的ID
- **回调：** [CRVideo_SendCmdRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| UID | String | 是 | 发送目标用户ID |
| data | String | 是 | 要发送的数据，字符串 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

发送点对点消息不需要进入房间，但是需要登录呼叫系统，即 `init` 时 `isCallSer` 参数的值不能为false
可以通过此接口发送自定义的json格式消息来实现一些自定义功能

CRVideo_SendBuffer(UID,data,cookie)

- **描述：**发送大块数据
- **返回值：**uuid，本次发送任务的ID
- **回调：**[CRVideo_SendBufferRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| UID | String | 是 | 发送目标用户ID |
| data | String | 是 | 要发送的数据 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

发送进度通知 [CRVideo_SendProgress.callback](#)
发送结果回调 [CRVideo_SendBufferRslt.callback](#)

CRVideo_SendFile(UID,File,cookie)

- **描述：**发送文件数据
- **返回值：**uuid，本次发送任务的ID
- **回调：**[CRVideo_SendFileRslt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|--------|--------|----|----------------------------------|
| UID | String | 是 | 发送目标用户ID |
| File | File | 是 | 要发送的文件对象 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

发送进度通知 [CRVideo_SendProgress.callback](#)
发送结果回调 [CRVideo_SendFileRslt.callback](#)

CRVideo_CancelSend(taskID)

- **描述：**取消发送
- **返回值：**无
- **回调：**无
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|----|----|----|----|
|----|----|----|----|

| | | | |
|--------|--------|---|------------|
| taskID | Number | 是 | 要取消发送的任务ID |
|--------|--------|---|------------|

CRVideo_SendCmdRslt.callback = function(taskID,sdkErr,cookie){}

- **描述：** 回调，发送点对点消息的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| taskID | String | 发送任务id |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SendBufferRslt.callback = function(taskID,sdkErr,cookie){}

- **描述：** 回调，发送大块数据的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| taskID | String | 发送任务id |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SendFileRslt.callback = function(taskID,sdkErr,cookie){}

- **描述：** 回调，发送文件的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| taskID | String | 发送任务id |
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SendProgress.callback = function(taskID,sendedLenth,totalLenth,cookie){}

- **描述：** 通知，当前发送进度
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|--------|
| taskID | String | 发送任务id |
| | | |

| | | |
|-------------|--------|----------------------------------|
| sendedLenth | Number | 已发送的数据长度 |
| totalLenth | Number | 需要发送的总长度 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_NotifyCmdData.callback = function(UID,data){}

- **描述：** 通知，收到点对点消息
- **回调参数：**

| 参数 | 类型 | 含义 |
|------|--------|----------|
| UID | String | 消息来源用户ID |
| data | String | 收到的数据 |

CRVideo_NotifyBufferData.callback = function(UID,data){}

- **描述：** 通知，收到大块数据
- **回调参数：**

| 参数 | 类型 | 含义 |
|------|--------|----------|
| UID | String | 数据来源用户ID |
| data | String | 收到的数据 |

CRVideo_NotifyFileData.callback = function(UID,fileBuffer,fileName){}

- **描述：** 通知，收到文件数据
- **回调参数：**

| 参数 | 类型 | 含义 |
|------------|-------------|----------------------|
| UID | String | 文件来源用户ID |
| fileBuffer | ArrayBuffer | 收到的文件数据， ArrayBuffer |
| fileName | String | 收到的文件文件名 |

示例代码：

```
// 将ArrayBuffer转成url并借助a标签自动下载
const blob = new Blob([fileBuffer]);
const url = window.URL.createObjectURL(blob);
const a = document.createElement('a');
a.style.display = 'none';
a.href = url;
a.download = fileName;
document.body.appendChild(a);
a.click();
```

收到的文件数据为ArrayBuffer格式，需要业务层根据需要转成url下载

CRVideo_SendMeetingCustomMsg(stringMsg,cookie)

- **描述：** 发送房间内广播消息
- **返回值：** 无
- **回调：** [CRVideo_SendMeetingCustomMsgRsIt.callback](#)
- **参数：**

| 参数 | 类型 | 必传 | 含义 |
|-----------|--------|----|----------------------------------|
| stringMsg | String | 是 | 要发送的消息数据 |
| cookie | String | 否 | 详细介绍见 关键词 ，可以传空串 |

CRVideo_SendMeetingCustomMsgRsIt.callback = function(sdkErr,cookie){}

- **描述：** 回调，发送房间内广播消息的结果
- **回调参数：**

| 参数 | 类型 | 含义 |
|--------|--------|----------------------------------|
| sdkErr | Number | 错误码，详见 错误码定义 |
| cookie | String | 详细介绍见 关键词 ，可以传空串 |

CRVideo_NotifyMeetingCustomMsg.callback = function(UID,stringMsg){}

- **描述：** 通知，收到房间内广播消息
- **回调参数：**

| 参数 | 类型 | 含义 |
|-----------|--------|----------|
| UID | String | 消息来源用户ID |
| stringMsg | String | 收到的消息数据 |

常量定义

错误码

CRVideo_ERR_DEF

| 数值 | 含义 |
|-----|-----------------------|
| 0 | 没有错误 |
| 1 | 未知错误 |
| 3 | sdk内部错误 |
| 5 | 参数错误 |
| 6 | 无效数据 |
| 7 | 帐号密码不正确 |
| 8 | 服务异常 |
| 9 | 登录状态错误 |
| 10 | 登录用户被踢下线 |
| 11 | sdk未初始化 |
| 12 | 还没有登录 |
| 13 | base64转换失败 |
| 14 | 启用了第三方鉴权，但没有携带鉴权信息 |
| 15 | 没有启用第三方鉴权，但携带了鉴权信息 |
| 16 | 访问第三方鉴权服务异常 |
| 17 | 第三方鉴权不通过 |
| 18 | token已过期 |
| 19 | 鉴权信息错误 |
| 20 | appid不存在 |
| 21 | 鉴权失败 |
| 22 | 非token鉴权方式 |
| 202 | 服务器没有响应 |
| 204 | socket异常 |
| 400 | 队列ID错误 |
| 401 | 没有用户在排队 |
| 402 | 排队用户已取消 |
| 403 | 队列服务还未开启 |
| 404 | 已在其它队列排队(客户只能在一个队列排队) |
| 600 | 无效的呼叫ID |
| 601 | 已在呼叫中 |
| 602 | 对方忙 |
| 603 | 对方不在线 |

| | |
|------|------------------------|
| 604 | 对方无应答 |
| 605 | 用户不存在 |
| 606 | 对方拒接 |
| 800 | 房间不存在或已结束 |
| 801 | 房间密码不正确 |
| 802 | 房间终端数量已满（购买的license不够） |
| 803 | 分配房间资源失败 |
| 804 | 房间已加锁 |
| 805 | 余额不足 |
| 1000 | 发送失败 |
| 1001 | 有敏感词语 |
| 1100 | 发送信令数据过大 |
| 1101 | 发送数据过大 |
| 1102 | 目标用户不存在 |
| 8001 | 获取媒体设备权限未知错误 |
| 8002 | 用户拒绝授予媒体权限 |
| 8003 | 枚举媒体设备失败 |
| 8004 | 浏览器不支持相关API |
| 8005 | SDK不支持该浏览器 |
| 8006 | 浏览器版本过低 |

麦克风状态

CRVideo_ASTATUS

| 数值 | 含义 |
|----|----------------|
| 0 | 音频状态未知 |
| 1 | 没有麦克风设备 |
| 2 | 麦克风处于关闭状态（软开关） |
| 3 | 麦克风处于打开状态（软开关） |
| 4 | 向服务器发送打开消息中 |

视频状态

CRVideo_VSTATUS

| 数值 | 含义 |
|----|--------|
| 0 | 视频状态未知 |

| | |
|---|---------------|
| 1 | 没有视频设备 |
| 2 | 视频处于关闭状态（软开关） |
| 3 | 视频处于打开状态（软开关） |
| 4 | 向服务器发送打开消息中 |

主视频模式

CRVideo_MAIN_VIDEO_MODE

| 数值 | 含义 |
|----|---------|
| 1 | 语音激励模式 |
| 2 | 跟随演示者 |
| 3 | 跟随特定参会者 |

功能区类型

CRVideo_MAIN_PAGE_TYPE

| 数值 | 含义 |
|----|------|
| 4 | 白板 |
| 6 | 屏幕共享 |
| 7 | 影音共享 |
| 8 | 视频墙 |

混图器内容类型

CRVideo_MIXER_VCONTENT_TYPE

| 数值 | 含义 |
|----|--------|
| 0 | 摄像头画面 |
| 1 | 图片 |
| 2 | 本地屏幕画面 |
| 3 | 影音共享画面 |
| 4 | 时间戳 |
| 5 | 共享的屏幕 |
| 6 | 白板 |
| 7 | 文本 |
| 9 | 纯音频 |

混图器状态

CRVideo_MIXER_STATE

| 数值 | 含义 |
|----|------|
| 0 | 没有创建 |
| 1 | 正在开启 |
| 2 | 正在进行 |
| 3 | 录制出错 |
| 4 | 正在上传 |
| 5 | 上传完成 |
| 6 | 上传出错 |
| 7 | 全部完成 |

类型定义

CRVideo_SDKParams

SDK初始化参数对象

类型：Object

| 参数 | 类型 | 含义 |
|--------------|--------|-------------------------------------------|
| isSDKConsole | Bool | 是否开启浏览器控制台SDK日志显示，非必传，缺省为true，开启 |
| isUploadLog | Bool | 是否开启SDK日志上传，非必传，缺省为true，开启 |
| videoSendBW | Number | 单个视频画面上送最大码率(带宽)Kbps，非必传，缺省为2048，即256KB/s |
| screenSendBW | Number | 屏幕共享画面上送最大码率(带宽)Kbps，非必传，缺省不设置上限 |
| mediaSendBW | Number | 影音共享画面上送最大码率(带宽)Kbps，非必传，缺省不设置上限 |
| isHttp | Bool | 是否强制使用http协议（默认为https），非必传，缺省为false，不启用 |
| isCallSer | Bool | 是否启用队列呼叫服务，非必传，缺省为true，启用 |
| isWebRTC | Bool | 是否启用音视频服务，非必传，缺省为true，启用 |
| MSProtocol | Number | 流媒体打洞协议，1：UDP，2：TCP，非必传，缺省为1，UDP协议 |

CRVideo_UserStatus

用户在线信息
类型：Object

| 参数 | 类型 | 含义 |
|------------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| userStatus | Number | 用户的在线状态(0离线 1空闲 2忙碌 3房间中) |
| DNDType | Number | 用户的免打扰状态(0关闭免打扰 1开启免打扰) |

CRVideo_QueueInfo

队列信息
类型：Object

| 参数 | 类型 | 含义 |
|-------|--------|--------------|
| queID | Number | 队列ID |
| prio | Number | 优先级，值越小优先级越高 |
| name | String | 队列名称 |
| desc | String | 队列描述 |

CRVideo_QueueStatus

队列状态
类型：Object

| 参数 | 类型 | 含义 |
|-----------|--------|----------------|
| queID | Number | 队列ID |
| agent_num | Number | 正在服务此队列坐席数 |
| wait_num | Number | 正在此队列排队的客户数 |
| srv_num | Number | 此队列当前正在服务中的客户数 |

CRVideo_QueueUser

队列用户信息
类型：Object

| 参数 | 类型 | 含义 |
|-------|--------|------|
| queID | Number | 队列ID |
| | | |

| | | |
|-------------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| nickname | String | 用户昵称 |
| queuingTime | Number | 用户已排队时长（秒） |
| usrExtDat | String | 用户排队时携带的拓展数据 |

CRVideo_QueueingInfo

排队信息

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|-------------|
| queID | Number | 队列ID |
| position | Number | 我当前的位置 |
| queuingTime | Number | 我排队的时长（单位s） |

CRVideo_SessionInfo

会话信息

类型：Object

| 参数 | 类型 | 含义 |
|---------------|--------|----------------------|
| callID | String | 呼叫全局标识ID |
| peerID | String | 会话中对方的UID |
| peerName | String | 会话中对方的昵称 |
| bCallAccepted | Number | 呼叫是否已接受，0：暂未接受，1：已接受 |
| meetingID | Number | 当前会话中的房间ID |
| meetingPswd | String | 当前会话中房间的密码 |
| duration | Number | 当前会话持续的时间，单位：s |

CRVideo_CreateParams

创建房间的参数对象

类型：Object

| 参数 | 类型 | 含义 |
|----|----|----|
|----|----|----|

| | | |
|-------------|--------|------------------------------------------------------------|
| meetSubject | String | 房间名称 |
| needPswd | Number | 是否创建密码（ 0:房间无密码， 1:系统为房间随机创建一个密码 ） |
| isRegular | Number | 是否为固定房间（ 0:普通房间， 房间无人30分钟后，自动销毁; 1:固定房间， 房间一直存在直到用户主动销毁它 ） |

CRVideo_RoomInfoObj

房间信息对象

类型：Object

| 参数 | 类型 | 含义 |
|------------|--------|--------|
| ID | Number | 房间号 |
| pswd | String | 房间密码 |
| subject | String | 房间主题 |
| pubMeetUrl | String | 房间公共链接 |

CRVideo_MemberInfo

成员信息

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|--------------------------------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| nickname | String | 昵称 |
| audioStatus | Number | 音频状态，数值参考麦克风状态定义 CRVideo_ASTATUS |
| videoStatus | Number | 视频状态，数值参考摄像头（ 视频 ）状态定义 CRVideo_VSTATUS |

CRVideo_VideoInfoObj

用户摄像头信息

类型：Object

| 参数 | 类型 | 含义 |
|---------|--------|---------------------------------|
| UID | String | 用户ID, 详细介绍见 关键词 |
| videoID | Number | 摄像头ID |
| | | |

| | | |
|-----------|--------|-------|
| videoName | String | 摄像头名称 |
|-----------|--------|-------|

CRVideo_MicInfoObj

用户麦克风信息

类型：Object

| 参数 | 类型 | 含义 |
|---------|--------|-------|
| micID | Number | 麦克风ID |
| micName | String | 麦克风名称 |

CRVideo_SpeakerInfoObj

用户麦克风信息

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|-------|
| speakerID | Number | 扬声器ID |
| speakerName | String | 扬声器名称 |

CRVideo_AudioCfg

系统音频参数

类型：Object

| 参数 | 类型 | 含义 |
|-----------|--------|-------|
| micID | Number | 麦克风ID |
| speakerID | Number | 扬声器ID |

CRVideo_VideoCfg

系统视频参数对象

类型：Object

| 参数 | 类型 | 含义 |
|------|--------|--------------------------------------|
| size | Number | 视频尺寸（ 1：流畅 2：标清 3：高清 4：超清 ） |
| fps | Number | 视频帧率（ 5~30 ） |

| | | |
|-------|--------|--------------------------------------------------------------|
| ratio | Number | 视频画面比例 (1 : 16/9 2 : 4/3 3 : 1/1 4 : 3/4 5 : 9/16) |
|-------|--------|--------------------------------------------------------------|

SDK默认视频尺寸为 2 : 标清，默认帧率为 15fps
PC端视频画面默认比例为 1 : 16/9，移动端视频画面默认比例为 3 : 1/1

CRVideo_ScreenCfgObj

屏幕共享配置参数对象

类型：Object

| 参数 | 类型 | 含义 |
|-----------|--------|------------------------------------------------------------------------|
| catchRect | Object | 共享区域 例如：{ left:100, top:100, width:400, height:400 }，不设置区域时此参数不传或传null |
| maxPFS | Number | 最大帧率 |
| maxKbps | Number | 最大码率Kbps |

CRVideo_CreateUIParams

创建成员视频UI显示组件的参数

类型：Object

| 参数 | 类型 | 含义 |
|--------|--------|--------------------------------------------------------------------------------------|
| poster | String | video标签的poster属性，当视频显示控件没有视频画面时的默认占位图片，非必传 |
| style | Object | video标签的style属性对象，标签css样式，非必传。例如：style: {objectFit:"cover",backgroundColor:"#000"} 等 |

此对象里面的各参数即video标签的官方属性，可以参考 <https://developer.mozilla.org/en-US/docs/Web/API/HTMLVideoElement>

CRVideo_MediaInfoObj

影音信息

类型：Object

| 参数 | 类型 | 含义 |
|-----------|--------|------------------------|
| userID | String | 用户ID |
| state | Number | 播放状态 0:播放中 1:暂停中 2:未播放 |
| mediaName | String | 影音文件名 |

CRVideo_MutiMixerCfg

服务器混图配置对象

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|-----------------------------------------------------------------------------|
| id | Number | 混图器ID，每个混图器的配置、内容、和输出要根据ID对应 |
| streamTypes | Number | 录像类型，1：单音频录制（对应aCfg参数） 2：单视频录制（暂不支持） 3：音视频录制（对应cfg参数） |
| cfg | Object | 音视频混图器配置对象，每个混图器只有一个配置对象（cfg或aCfg），详见： CRVideo_MixerCfgObj |
| acfg | Object | 单音频混图器配置对象，每个混图器只有一个配置对象（cfg或aCfg），详见： CRVideo_MixeraCfgObj |

示例：

```
[[
  "id": 1,
  "streamTypes": 3,
  "cfg": {
    "width": 1280,
    "height": 720,
    "frameRate": 15,
    "bitRate": 2000000,
    "defaultQP": 15
  }
], {
  "id": 2,
  "streamTypes": 1,
  "aCfg": {
    "channelType": 1,
    "audioFormat": 1
  }
}]
```

CRVideo_MutiMixerContent

服务器混图内容对象

类型：Object

| 参数 | 类型 | 含义 |
|---------|--------|-------------------------------------------------------|
| id | Number | 混图器ID，每个混图器的配置、内容、和输出要根据ID对应 |
| content | Object | 混图器内容配置对象，详见： CRVideo_MixerContentObj |

示例：


```
[[
  "id": 1,
  "content": [{
    "type": 0,
    "left": 0,
    "top": 0,
    "width": 1280,
    "height": 720,
    "param": {
      "camid": "张三.1"
    },
    "keepAspectRatio": 1
  }, {
    "type": 4,
    "left": 35,
    "top": 3,
    "width": 175,
    "height": 32,
    "keepAspectRatio": 1
  }, {
    "type": 7,
    "left": 40,
    "top": 30,
    "param": {
      "text": "<span style='font-size:14px;color:#e21918'>张三</span>"
    }
  }
]]
], {
  "id": 2,
  "content": [{
    "type": 9,
    "param": "张三"
  }]
}]
}]
```

CRVideo_MutiMixerOutput

服务器混图输出对象

类型：Object

| 参数 | 类型 | 含义 |
|--------|--------|------------------------------------------------------|
| id | Number | 混图器ID，每个混图器的配置、内容、和输出要根据ID对应 |
| output | Object | 混图器输出配置对象，详见： CRVideo_MixerOutputObj |

示例：

```
{
  "id": 1,
  "output": {
    "type": 0,
    "filename": "/2020-07-21/张三的录像.mp4"
  }, {
    "type": 1,
    "liveUrl": "rtmp://pub.cloudroom.com/live/xxx?xxx=xxx"
  }
}, {
  "id": 2,
  "output": {
    "type": 0,
    "filename": "/2020-07-21/张三的录音.mp3"
  }
}
```

CRVideo_MixerCfgObj

音视频混图器规格配置对象

类型：Object

| 参数 | 类型 | 含义 |
|-----------|--------|----------------------------------------------------------------------|
| width | Number | 图像宽度（要求8的倍数） |
| height | Number | 图像高度（要求8的倍数） |
| frameRate | Number | 图像帧率，取值范围:1-30(值越大，画面越流畅，cpu消耗更高。录像推荐15帧，直播推荐25帧) |
| bitRate | Number | 录制视频文件的最高码率(如1M:1000000)，当图像变化小时，实际码率会低于此值 |
| defaultQP | Number | 录制视频文件的目标质量，缺省值：26。取值范围：0~51，0表示完全无损,51表示质量非常差，推荐高质量取值18，中质量26，低质量34 |

示例：

```
// 音视频
{"width":1280,"height":720,"frameRate":15,"bitRate":2000000,"defaultQP":15}
```

CRVideo_MixeraCfgObj

单音频混图器规格配置对象

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|----------------------|
| channelType | Number | 音频文件通道类型，0：单声道 1：双声道 |

| | | |
|-------------|--------|--------------------------|
| audioFormat | Number | 音频文件格式，0：AAC 1：MP3 2：PCM |
|-------------|--------|--------------------------|

示例：

```
// 单音频
{"channelType":1,"audioFormat":1}
```

CRVideo_MixerContentObj

混图器内容配置对象

类型：Object

| 参数 | 类型 | 含义 |
|-----------------|--------|--------------------------------------------------------------|
| type | Number | 类容类型，详见混图器内容类型定义 CRVideo_MIXER_VCONTENT_TYPE |
| left | Number | 该内容在混图画面中水平方向相对于最左边的位置 |
| top | Number | 该内容在混图画面中垂直方向相对于最顶部的位置 |
| width | Number | 该内容在混图画面中的宽度 |
| height | Number | 该内容在混图画面中的高度 |
| keepAspectRatio | Number | 内容是否保持原始比例，0：不保持 1：保持 |
| param | Object | 参数配置，json格式，如：{"camid":"user_xxx.1"}，详见下方介绍 |

示例：

```
// 摄像头画面
{"type":0,"left":0,"top":0,"width":1280,"height":720,"param":{"camid":"user_xxx.1"},"keepAspectRatio":1}
// 影音
{"type":3,"left":0,"top":0,"width":1280,"height":720}
// 时间戳
{"type":4,"left":35,"top":3,"width":175,"height":32,"keepAspectRatio":1}
// 共享中的屏幕
{"type":5,"left":0,"top":0,"width":1280,"height":720}
// 文本
{"type":7,"left":40,"top":30,"param":{"text":"<span style='font-size:14px;color:#e21918'>H5_5598</span>"}}
// 纯音频
{"type":9,"param":"user_H5_5598"}
```

- type描述：
- 当 type=0 时，表示混图的是摄像头，param必须包含camid；
 - 当 type=1 时，表示混图的是指定的图片，param必须包含resourceid（仅用于本地混图，暂不支持）；
 - 当 type=2 时，表示混图的是本地屏幕，param可以增加附加参数screenid/pid/area（暂不支持）；
 - 当 type=3 时，表示混图的是影音共享，不用附加任何参数；
 - 当 type=4 时，表示混图的是时戳，不用附加任何参数；
 - 当 type=5 时，表示混图的是共享中的屏幕，不用附加任何参数；
 - 当 type=6 时，表示混图的是白板，不用附加任何参数。（仅用于云端混图，本地混图应该用MIXVTP_PIC，暂不支持）；

当 type=7 时，表示混图的是文本(支持简单html)，param必须包含text；
注：当 type=7 时，width和height可为0，此时元素大小由文本信息自动确定；
param支持的参数如下：
1.camid：用户id.摄像头id, 如："testuser.1"；
2.text：录制文本内容，支持一些简单的html标签，如：
客户姓名：
张三

录制日期：
2018-03-01
3.录制纯音频文件时，param为某个用户的UID，表示录制该用户的声音；

CRVideo_MixerOutputObj

混图器输出配置对象

类型：Object

| 参数 | 类型 | 含义 |
|----------|--------|-----------------------------------------------------------------------------------------------|
| type | Number | 输出类型，0：输出文件，如MP4、MP3文件 1：直播推流 |
| filename | String | 录像路径文件名（服务器录像名格式如：/2018-11-21/xxx.mp4），支持的文件格式为mp4/ts/flv/avi，其中flv和ts两种格式在程序异常结束时产生的录制文件仍可用。 |
| liveUrl | String | 直播推流地址，支持rtmp/rtsp协议 |

示例：

```
// 文件
{"type":0,"filename":"/2020-07-21/2020-07-21_14-21-53_Webrtc_53315471.mp4"}
{"type":0,"filename":"/2020-07-21/2020-07-21_14-21-53_Webrtc_53315471.mp3"}
// 推流
{"type":1,"liveUrl":"rtmp://pub.cloudroom.com/live/xxxx?xxx=xxx"}
```

当 type为 0 时，需要传参 filename
当 type为 1 时，需要传参 liveUrl

CRVideo_MixerOutputInfoObj

云端录制文件、云端直播推流信息

类型：Object

| 参数 | 类型 | 含义 |
|----------|--------|---------------------------------------------------------|
| id | Number | 混图器ID |
| state | Number | 当前云端录制、云端直播推流状态，详见： CRVideo_MIXER_STATE |
| duration | Number | 录像文件时长，单位：毫秒 |
| fileSize | Number | 录像文件大小 |
| fileName | String | 录像文件名 |
| | | |

| | | |
|---------|--------|-------------------------------|
| errCode | Number | 错误码，详见： 错误码定义 |
|---------|--------|-------------------------------|

示例：

```
// 文件
{"type":0,"filename":"/2020-07-21/2020-07-21_14-21-53_Webrtc_53315471.mp4"}
{"type":0,"filename":"/2020-07-21/2020-07-21_14-21-53_Webrtc_53315471.mp3"}
// 推流
{"type":1,"liveUrl":"rtmp://pub.cloudroom.com/live/xxxx?xxx=xxx"}
```

- 当 type为 0 时，需要传参 filename
- 当 type为 1 时，需要传参 liveUrl

CRVideo_MeetingAttrObjs

会议属性集

类型：Object

| 参数 | 类型 | 含义 |
|------------------|--------|-------------------------------|
| value | String | 属性值 |
| lastModifyUserID | String | 最后修改者 |
| lastModifyTs | Number | 最后的修改时间，1970-1-1 0:00:00以来的秒数 |

示例：

```
{
  "userdef Key1": {
    "value": "11111",
    "lastModifyUserID": "111",
    "lastModifyTs": 11111
  },
  "userdef Key2": {
    "value": "22222",
    "lastModifyUserID": "222",
    "lastModifyTs": 22222
  }
}
```

CRVideo_UsrMeetingAttrObjs

用户属性集

类型：Object

| 参数 | 类型 | 含义 |
|-------|--------|-----|
| value | String | 属性值 |

| | | |
|------------------|--------|-------------------------------|
| lastModifyUserID | String | 最后修改者 |
| lastModifyTs | Number | 最后的修改时间，1970-1-1 0:00:00以来的秒数 |

示例：

```
{
  "userid1": {
    "userdef Key1": {
      "value": "11111",
      "lastModifyUserID": "111",
      "lastModifyTs": 11111
    },
    "userdef Key2": {
      "value": "22222",
      "lastModifyUserID": "222",
      "lastModifyTs": 22222
    }
  },
  "userid2": {
    "userdef Key1": {
      "value": "11111",
      "lastModifyUserID": "111",
      "lastModifyTs": 11111
    },
    "userdef Key2": {
      "value": "22222",
      "lastModifyUserID": "222",
      "lastModifyTs": 22222
    }
  }
}
```

CRVideo_MeetingAttrOptions

操作房间属性选项

类型：jsonString

| 参数 | 类型 | 含义 |
|-----------|--------|------------------------|
| notifyAll | Number | 1：通知房间所有人员，0（默认值）:不通知。 |

示例：

```
{'notifyAll':1}
```

CRVideo_MarkOptions

屏幕共享标注参数对象

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|-----------------------------------------------------------------------------|
| allowMark | Bool | 是否允许标注。非必传，缺省为 false，不允许标注 |
| type | Number | 画笔类型。非必传，缺省为 1，铅笔 |
| strokeStyle | String | 画笔颜色。非必传，缺省为 '#000000'，黑色 |
| lineWidth | Number | 画笔线宽px。为必传，缺省为 2，2px |
| cursor | String | 画笔鼠标样式，如：'url("./image/pc/meeting_pen1_.ico"), auto'。非必传，缺省为 'default'，默认样式 |

示例：

```
{
  "allowMark": true,
  "type": 1,
  "strokeStyle": "#ff0000",
  "lineWidth": 3,
  "cursor": 'url("./image/pc/meeting_pen1_.ico"), auto'
}
```

CRVideo_BoardObj

白板对象

类型：Object

| 参数 | 类型 | 含义 |
|-----------|--------|-------|
| boardID | String | 白板标识 |
| title | String | 白板名 |
| width | Number | 白板宽度 |
| height | Number | 白板高度 |
| pageCount | Number | 白板页数 |
| curPage | Number | 白板当前页 |

示例：

```
{
  boardID: "2.59",
  title: "默认白板",
  width: 800,
  height: 600,
  pageCount: 3,
  curPage: 2,
}
```

CRVideo_BoardPenOptions

白板UI组件画笔配置对象

类型：Object

| 参数 | 类型 | 含义 |
|-------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type | Number | 画笔类型，缺省为4(铅笔)。其它类型——5(荧光笔，暂不支持)，6(一条直线，暂不支持)，7(单箭头线，暂不支持)，8(双箭头线，暂不支持)，9(文本，暂不支持)，10(实心矩形，暂不支持)，11(空心矩形，暂不支持)，12(圆角实心矩形，暂不支持)，13(圆角空心矩形，暂不支持)，14(实心椭圆，暂不支持)，15(空心椭圆，暂不支持)， |
| cursor | String | 鼠标样式（css），缺省为:"default"，其它样式可自定义为css样式，如图片：'cursor: url("./image/pc/meeting_pen1_.ico"), auto' |
| lineType | Number | 线型，缺省为1(实线)。1(实线)，3(虚线) |
| lineWidth | Number | 线宽 |
| strokeStyle | String | 线条颜色，hex值 |
| fillColor | String | 填充颜色，hex值 |
| fullOpacity | Number | 填充透明度，小数 |

示例：

```
{
  type: 4, // 画笔类型
  cursor: 'default', // 鼠标样式
  lineType: 1, // 线型 1实线 3虚线
  lineWidth: 2, // 线宽
  strokeStyle: '#0000FF', // 线条颜色
  fillColor: '#FC011A', // 填充颜色
  fullOpacity: 0.5, // 填充透明度
}
```


版本历史及Demo下载

SDK下载

- [点此下载](#)  最新版H5SDK 及 Demo
- SDK在线地址：<https://sdk.cloudroom.com/web/download/SDK/H5SDK/latest/RTCSDKV2.min.js> 

版本变更记录

2021-08-18 V2.5.1.6

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.5.1.6/RTCSDKV2.min.js> 

1. SDK：修复某些情况下进入房间报错的问题；
2. SDK：修复未建立连接时上报日志报错的问题；
3. SDK：修复录制过程中进入房间无法更新录制内容的问题；
4. SDK：修复某些场景下掉线重登导致成员列表异常，视频订阅异常的问题；
5. SDK：优化某些场景下掉线重登身份会重复的问题；
6. SDK：新增页面事件监听，强制离开页面时自动退出房间和登录；
7. SDK：解决某些情况下初始化失败响应不正确的问题；
8. SDK：启用新的入会接口；
9. Demo：PC会议Demo修复录制中退出房间，进入新的房间录制状态显示不正确的问题；
10. Demo：PC会议Demo修复某些情况下掉线重登不显示画面的问题；
11. 本次更新涉及接口变动（相对于2.5.0.x），详见开发文档
 - [CRVideo_StartSvrMixer](#)（改为promise）
 - [CRVideo_UpdateSvrMixerContent](#)（改为promise）
 - [CRVideo_StopSvrMixer](#)（改为promise）
 - [CRVideo_OpenMic](#)（改为promise）
 - [CRVideo_EnterMeeting](#)（废弃）
 - [CRVideo_EnterMeeting2](#)（废弃）
 - [CRVideo_EnterMeeting3](#)（新增）

2021-08-03 V2.5.0.2

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.5.0.2/RTCSDKV2.min.js> 

1. SDK：新增屏幕区域共享功能；
2. SDK：新增白板功能；
3. SDK：新增功能区相关接口；
4. SDK：新增添加canvas摄像头功能；
5. SDK：新增房间内日志按房间号分开同步上传；
6. SDK：修复全体静音时没有其他成员音频状态变化通知的问题；
7. SDK：修复某些情况下透明通道无法发送文件的问题；
8. SDK：修复点击浏览器工具条结束屏幕共享时内存泄漏的问题；
9. Demo：PC会议Demo新增共享区域设置；
10. Demo：PC会议Demo新增白板功能；
11. Demo：PC会议Demo新增支持视频墙分屏数切换；
12. Demo：PC会议Demo新增支持视频墙翻页；
13. Demo：PC会议Demo新增踢人按钮；
14. Demo：PC会议Demo新增修改昵称按钮；

15. Demo：PC会议Demo新增全体静音按钮；
16. Demo：PC会议Demo新增分享会议按钮；
17. Demo：PC会议Demo新增支持链接入会；
18. Demo：PC会议Demo添加虚拟摄像头（图片）按钮；
19. Demo：PC会议Demo新增界面同步功能；
20. 本次更新涉及接口变动（相对于2.4.0.x），详见开发文档
 - [CRVideo_Kickout](#)
 - [CRVideo_SetNickName](#)（改为promise）
 - [CRVideo_SetScreenShareCfg](#)
 - [CRVideo_SwitchToPage](#)
 - [CRVideo_GetCurrentMainPage](#)
 - [CRVideo_GetCurrentSubPage](#)
 - [CRVideo_Creatboard](#)
 - [CRVideo_CreateBoardUIObj](#)
 - [CRVideo_Closeboard](#)
 - [CRVideo_SetBoardCurPageNo](#)
 - [CRVideo_AddCanvasVCam](#)
 - [CRVideo_RemoveCanvasVCam](#)
 - [CRVideo_BoardUIObj.handler](#)
 - [CRVideo_BoardUIObj.setCurBoardPage](#)
 - [CRVideo_BoardUIObj.isScaledContents](#)
 - [CRVideo_BoardUIObj.setTitleStyle](#)
 - [CRVideo_BoardUIObj.setBoardStyle](#)
 - [CRVideo_BoardUIObj.setPen](#)
 - [CRVideo_BoardUIObj.undo](#)
 - [CRVideo_BoardUIObj.redo](#)
 - [CRVideo_BoardUIObj.clear](#)
 - *[CRVideo_BoardUIObj.addElement](#)
 - *[CRVideo_BoardUIObj.removeElement](#)
 - [CRVideo_NotifySwitchToPage.callback](#)
 - [CRVideo_NotifyInitBoards.callback](#)
 - [CRVideo_NotifyBoardCurPageNo.callback](#)
 - [CRVideo_NotifyCreateBoard.callback](#)
 - [CRVideo_NotifyCloseBoard.callback](#)
 - 带*号表示暂不可用

2021-06-18 V2.4.0.1

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.4.0.1/RTCSDKV2.min.js> 

1. SDK:新增屏幕共享标注功能；
2. SDK:新增业务层主动退订指定视频的方法；
3. Demo:PC端会议Demo新增支持共享标注；
4. Demo:移动端会议Demo新增支持显示房间内影音共享；
5. Demo:修复PC端会议Demo屏幕共享时退出房间再次进入仍然显示共享窗口的问题；
6. Demo:修复移动端会议Demo多人进出房间时视频墙异常的问题；
7. Demo:修复移动端会议Demo某些情况下无法正常观看屏幕共享的问题；
8. 本次更新涉及接口变动（相对于2.3.1.x），详见开发文档
 - [CRVideo_StartScreenMark](#)
 - [CRVideo_StopScreenMark](#)

- [CRVideo_ScreenShareObj.setMark](#)
- [CRVideo_StartScreenMarkRslt.callback](#)
- [CRVideo_StopScreenMarkRslt.callback](#)
- [CRVideo_NotifyScreenMarkStarted.callback](#)
- [CRVideo_NotifyScreenMarkStopped.callback](#)
- [CRVideo_VideoObj.setVideo](#)

2021-05-17 V2.3.1.0

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.3.1.0/RTCSDKV2.min.js> [↗](#)

1. SDK:新增修改房间成员昵称接口；
2. SDK:新增房间成员昵称改变的通知接口；
3. SDK:解决某些情况下屏幕共享无法正确停止的问题；
4. 本次更新涉及接口变动（相对于2.3.0.x），详见开发文档
 - [CRVideo_SetNickName](#)
 - [CRVideo_NotifyNickNameChanged.callback](#)

2021-04-13 V2.3.0.3

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.3.0.3/RTCSDKV2.min.js> [↗](#)

1. SDK:新增文件传输功能;
2. SDK:新增取消发送数据接口;
3. SDK:新增根据分辨率自动限制最大编码码率;
4. SDK:修复发送大块数据功能的一些错误;
5. SDK:优化影音共享，修复某些浏览器共享时对端黑屏的问题;
6. Demo:PC端会议Demo新增透明通道模块;
7. 本次更新涉及接口变动（相对于2.2.x.x），详见开发文档
 - [CRVideo_SendFile](#)
 - [CRVideo_CancelSend](#)
 - [CRVideo_SendFileRslt.callback](#)
 - [CRVideo_NotifyFileData.callback](#)

2021-03-29 V2.2.0.6

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.2.0.6/RTCSDKV2.min.js> [↗](#)

1. 新增对ios14.3+上uni-app的支持；
2. 修复mac Safari上的一些错误；
3. 修复某些情况下无法切换摄像头的问题；
4. 修复某些情况下无法播放远端影音共享的问题；
5. 优化服务器选择逻辑，选择更优的会议服务器；
6. 其它优化；

2021-03-16 V2.2.0.2

<https://sdk.cloudroom.com/web/download/SDK/H5SDK/2.2.0.2/RTCSDKV2.min.js> [↗](#)

1. 新增支持订阅多个摄像头；
2. 新增接口查询用户当前打开的摄像头ID集合；
3. 新增接口设置/查询是否启用多摄像头；

4. 优化媒体流摄像头功能；
5. 优化视频、屏幕共享、影音共享相关功能；
6. 重构PC端会议Demo，优化PC端呼叫(双录)Demo；
7. 本次更新设计接口变动（相对于2.1.0.x）
 - [CRVideo_GetOpenedVideoIDs](#)
 - [CRVideo_SetEnableMutiVideo](#)
 - [CRVideo_GetEnableMutiVideo](#)
 - CRVideo_OpenStreamVideo（废弃）
 - CRVideo_CloseStreamVideo（废弃）

2021-02-23 V2.1.0.7

1. 新增支持iOS14.3上微信内置浏览器及QQ浏览器等；
2. 解决某些浏览器没有setsinkid接口报错的问题；
3. 修改rtc默认协议为UDP/TCP自动选择；

2021-01-28 V2.1.0.5

1. 解决SDK默认音频设备和系统默认音频设备不一致的问题；

2020-12-05 V2.1.0.4

1. 新增会议属性相关接口；
2. 新增用户属性相关接口；
3. 新增会议服务器Resm探测；
4. 新增播放/暂停房间内声音接口；
5. 修复被踢出房间后自动入会的问题；
6. 修复某些情况下进入房间失败的问题；
7. 解决某些移动设备浏览器不带移动端标识导致的一些问题；
8. 本次更新涉及接口变动（相对于2.0.2.x）
 - [CRVideo_PlaySpeaker](#)
 - [CRVideo_PauseSpeaker](#)
 - 所有房间属性、用户属性相关接口（详见文档）

2020-10-21 V2.0.2.1

1. 优化屏幕共享清晰度、流畅度；
2. 新增参数设置屏幕共享帧率；
3. 新增支持MacOS上Safari13屏幕共享；
4. 本次更新设计接口变动（相对于2.0.1.x）
 - [CRVideo_SetSDKParams](#)

2020-09-25 V2.0.1.2

1. 新增设置SDK参数接口；
2. 视频会议Demo新增邀请功能；
3. 修改视频会议Demo为自动登录；
4. 双录/呼叫Demo新增一些操作确认或提醒；

5. 解决多项目环境录像只存在默认项目的问题；
6. 解决vivo Chrome打不开摄像头的问题；
7. 解决iPhone 6Plus不能切换前后摄像头的问题；
8. 解决华为Mate30pro无法打开摄像头的问题；
9. 解决各Demo离线访问不显示设备列表的问题；
10. 解决Chrome离线访问时无法开启摄像头的问题；
11. 新增初始化时判断浏览器是否支持，对应错误码8005、8006；
12. 本次更新涉及接口变动（相对于2.0.0.x）：
 - [CRVideo_SetSDKParams](#)

2020-08-18 V2.0.0.0

1. 重构队列管理模块；
2. 重构呼叫模块；
3. 新增邀请功能；
4. 本次更新涉及接口变动：（相对于V1.8.3）
 - [CRVideo_GetSDKVersion](#)
 - [CRVideo_WriteLog](#)
 - [CRVideo_CreateMeetingFail.callback](#)
 - [CRVideo_GetAudioMicNames](#)
 - [CRVideo_GetAudioSpkNames](#)
 - [CRVideo_GetMicEnergy](#)
 - [CRVideo_SetAllAudioClose](#)
 - [CRVideo_GetAllVideoInfo](#)
 - [CRVideo_SetVideoCfg](#)
 - [CRVideo_GetVideoCfg](#)
 - [CRVideo_GetMediaImg（新增）](#)
 - [CRVideo_Getmediaimg（废弃）](#)
 - [CRVideo_SendCmd](#)
 - [CRVideo_SendCmdRslt](#)
 - [CRVideo_SendIMMsg（废弃）](#)
 - [CRVideo_SendIMMsgRslt（废弃）](#)
 - [CRVideo_NotifyIMMsg（废弃）](#)
 - [CRVideo_NotifyIMMsg（废弃）](#)
 - [CRVideo_ScreenShareObj.setVideo](#)
 - [CRVideo_MediaObj.setVideo](#)
 - 所有队列管理相关接口（详见文档）
 - 所有呼叫相关接口（详见文档）
 - 所有邀请相关接口（详见文档）

注意：此版本为大版本更新，在使用V1.x版本的项目中升级V2.x的SDK务必检查所有改动的接口，对照文档检查传参是否一致

2020-07-10 V1.8.3.2

1. 新增支持云端录制纯音频文件；
2. 解决移动端反初始化后再次登录入会音频建流异常的问题；
3. 新增后台是否支持多项目判断；
4. 重构录制模块；
5. 修改录制逻辑，SDK内部不再自动更新录制参数，需由业务层调用更新；

6. 本地更新涉及接口变动：（相对于V1.8.2）

- [CRVideo_StartSvrMixer](#)
- [CRVideo_UpdateSvrMixerContent](#)
- [CRVideo_SvrMixerStateChanged.callback](#)
- [CRVideo_SvrMixerCfgChanged.callback](#)
- [CRVideo_SvrMixerOutputInfo.callback](#)

注意：升级此版本后，使用H5SDK支持多项目需升级web后台！

2020-06-30 V1.8.2.1

1. 新增房间内广播消息接口，废弃原IM接口；
2. 各Demo聊天消息接口替换为房间内广播接口，与其他平台互通；
3. 解决某些情况下快速开关麦导致的各种异常；
4. 本次更新涉及接口变动：（相对于V1.8.1）
 - [CRVideo_SendMeetingCustomMsg](#)
 - [CRVideo_NotifyMeetingCustomMsg.callback](#)

2020-06-22 V1.8.1.3

1. 解决某些情况下读取sessionStorage报错的问题；
2. 解决ws协议会议服务器连接失败的问题；
3. 解决http协议时某些服务器地址端口不正确的问题；
4. 解决云端录制参数不正确导致最后一位成员没有画面的问题；
5. 兼容safari13对webrtc的修改（服务端）；
6. 新增init接口支持传参设置屏幕共享和影音共享上送最大码率；
7. 优化分辨率设置逻辑，不再强制指定分辨率，交由浏览器处理；
8. 本次更新没有接口变动

2020-05-11 V1.8.1.0

1. 新增支持添加媒体流作为摄像头相关接口；
2. 优化入会速度；
3. 优化部分错误提示；
4. 优化视频分辨率上报逻辑；
5. 修改各Demo登录界面账号密码字样为AppID；
6. 修复某些情况下全体静音接口出错的问题；
7. 修复某些情况下会议未启动导致进入会议失败的问题；
8. 本次更新涉及接口变动：（相对于V1.8.0）
 - [CRVideo_AddStreamCam](#)
 - [CRVideo_OpenStreamVideo](#)
 - [CRVideo_CloseStreamVideo](#)

2020-01-06 V1.8.0.3

1. CRVideo_Init接口改为promise类型，初始化时获取媒体设备权限；
2. 新增：支持第三方鉴权及相关错误回调；
3. 新增：init接口支持传入参数控制流媒体协议；
4. 新增：init接口支持传入参数控制是否连接callserver服务；

5. 新增：登录接口新增支持cookie参数；
6. 新增：入会接口新增支持cookie参数；
7. 新增：网络掉线通知接口，由业务层处理重登逻辑；
8. 新增：支持免插件共享屏幕；
9. 优化：入会后媒体设备开关逻辑；
10. 优化：网络波动导致的各种异常；
11. 优化：会议中开关设备操作过快导致的各种异常；
12. 本次更新涉及接口变动：（相对于V1.7.0）
 - [CRVideo_Init](#)

2019-11-29 V1.7.0.1

1. 新增：挂断接口支持发送自定义参数；
2. 新增：支持Token鉴权；
3. 新增：支持多项目帐号；
4. 新增：支持强制http模式；
5. 新增：发送透明通道消息新增结果回调接口；
6. 新增：登录鉴权失败接口回调增加详细错误码；
7. 新增：meeting掉线通知接口；
8. 新增：callserver掉线通知接口；
9. 解决：某些情况下2分钟呼叫被自动挂断的问题；
10. 解决：某些情况下业务层传参队列id类型不匹配导致的异常问题；
11. 修改：创建房间接口，只需要传房间名一个参数，且必须先登录鉴权才能创建房间；
12. 本次更新涉及接口变动：（相对于V1.6.7）
 - [CRVideo_Init](#)
 - [CRVideo_Login](#)
 - [CRVideo_LoginByToken](#)
 - [CRVideo_UpdateToken](#)
 - [CRVideo_NotifyTokenWillExpire.callback](#)
 - [CRVideo_CreateMeeting](#)
 - [CRVideo_SendCmdRslt.callback](#)
 - [CRVideo_LineOff.callback](#)
 - [CRVideo_MeetingDropped.callback](#)
 - [CRVideo_HungupCall](#)
 - [CRVideo_NotifyCallHungup.callback](#)

2019-11-22 V1.6.7.0

1. 新增Safari浏览器支持VP8编码互通；
2. 新增呼叫接口支持添加自定义数据；
3. 新增支持多台access服务器负载，自动选择网络最优服务器；
4. 新增支持业务层控制视频上传最大带宽；
5. 新增支持不显示对方画面时不建流，减少带宽占用；
6. 新增支持初始化接口传入参数控制日志是否上传、是否开启音视频等；
7. 新增支持对方打开多个摄像头时，订阅默认摄像头；
8. 新增语音能量通知接口；
9. 新增主视频模式设置，支持语音激励模式；
10. 新增主视频变化通知接口；

11. 新增callserver连接异常重连处理；
12. 新增云端录制参数中userID不正确的兼容处理；
13. 新增设置视频参数容错；
14. 新增销毁会议接口及通知；
15. 新增远程双录pc端demo；
16. 新增影音共享功能；
17. 新增呼叫会议外第三方入会；
18. 优化屏幕共享，调整码率，分辨率，带宽等；
19. 优化视频订阅建流逻辑；
20. 优化日志上报；
21. 优化入会耗时；
22. 优化savePictoBase64接口，新增参数支持修改输出图片大小；
23. 修复某些情况下关闭摄像头报错的问题；
24. 修复设置主视频接口日志报错的问题；
25. 修复某些情况下SDK账号不能创建固定房间的问题；
26. 修复某些情况下没能创建音频播放器导致会议内没有声音的问题；
27. 修复某些情况下队列ID类型不匹配导致坐席无法取消队列服务的问题；
28. 修复某些情况下chrome76+上送画面模糊的问题；
29. 修复某些设备上360浏览器poster属性导致视频画面不显示的问题；
30. 修复chrome76+上某些情况下上送带宽不足导致画面模糊的问题；
31. 修复IM聊天发送JSON数据对端解析失败的问题；
32. 修复云端录制某些情况下摄像头id不正确导致偶尔没有画面的问题；
33. 修复某些情况下队列状态不正确的问题；
34. SDK默认创建固定会议；
35. 合并基础白板功能到此版本SDK；