

低时延直播

# 客户端 SDK 参考

文档版本 01  
发布日期 2024-07-05



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

---

# 目录

---

<b>1 SDK 介绍</b> .....	<b>1</b>
<b>2 版本规划</b> .....	<b>3</b>
<b>3 Web SDK</b> .....	<b>4</b>
3.1 浏览器适配.....	4
3.2 开发前准备.....	5
3.3 SDK 使用.....	6
3.4 基本使用逻辑.....	8
3.5 接口参考.....	11
3.5.1 主入口 (HWLLSPlayer) .....	11
3.5.2 客户端对象 (HWLLSClient) .....	15
3.5.3 客户端对象 (HWFlvClient) .....	26
3.5.4 客户端对象 (HWHlsClient) .....	33
3.5.5 客户端事件通知 (HWLLSClientEvent) .....	40
3.5.6 错误码 (ErrorCode) .....	44
3.5.7 公网地址.....	45
3.5.8 客户端错误码.....	45
3.6 常见问题.....	47
3.7 修订记录.....	48
<b>4 修订记录</b> .....	<b>49</b>

# 1 SDK 介绍

华为云低时延直播的软件开发工具包是对低时延直播提供的REST API进行的封装，以简化用户开发工作。用户直接调用低时延直播SDK提供的接口函数，即可实现使用低时延直播业务能力的目的。各类客户端SDK的下载和集成操作、接口参考等，请参见[表1-1](#)。

## 须知

目前不支持信令接入低时延直播业务，仅支持SDK接入。

表 1-1 客户端 SDK

客户端	SDK下载	版本号	开发者	主要功能	集成SDK	接口参考
Web	SDK: <a href="#">HWLLS_SDK_Web_2.6.9.tar.gz</a> 完整性校验: <a href="#">HWLLS_SDK_Web_2.6.9.tar.gz.sha256</a>	2.6.9	华为云	提供低时延直播业务能力	<a href="#">Web SDK集成</a>	<a href="#">Web SDK接口参考</a>

## 软件包完整性校验

用户可对下载的SDK包进行完整性校验，判断下载过程中是否存在篡改和丢包现象。

详细操作如下所示：

**步骤1** 在[表1-1](#)中下载SDK包及其完整性校验sha256包至本地。

**步骤2** 打开本地命令提示符框，输入如下命令，在本地生成已下载SDK包的SHA256值。

其中，“D:\HWLLS\_SDK\_Web\_2.6.0.tar.gz”为SDK包本地存放路径和SDK包名，请根据实际情况修改。

```
certutil -hashfile D:\HWLLS_SDK_Web_2.6.0.tar.gz SHA256
```

命令执行结果示例，如下所示：

```
SHA256 的 D:\HWLLS_SDK_Web_2.6.0.tar.gz 哈希:  
3ac83be852e8dcc9e90f236801fd4c494983073543e1ae66ee4d0c29043dccd1  
CertUtil: -hashfile 命令成功完成。
```

**步骤3** 比对查询出的SDK包SHA256值和下载后的SDK包SHA256值。

如果一致，说明下载过程中不存在篡改和丢包现象。

----结束

# 2 版本规划

低时延直播（Low Latency Live）客户端SDK的版本规划说明。

## 版本号说明

版本号格式为a.b.c，其中：

- a为主版本号：在版本架构重构的情况下更新。例如，在版本间接口存在兼容问题时会进行变更。
- b为次版本号：正常迭代版本，如有新功能特性、接口新增或优化等，则该字段递增。
- c补丁版本号：如有功能优化或缺陷修复时，则该字段递增。

版本号样例：2.0.1

## 版本周期

默认1-2个月发布一个版本，或根据客户的诉求进行发布变更。

## 版本约束

无，新老版本兼容。

# 3 Web SDK

## 3.1 浏览器适配

本章节介绍低时延直播Web SDK支持的浏览器类型、版本以及使用限制。

表 3-1 浏览器适配详情

操作系统类型	浏览器类型	浏览器版本
Windows	Chrome浏览器	67+
	QQ浏览器（极速内核）	10.4+
	360安全浏览器（极速模式）	12
	微信内嵌浏览器	-
	Firefox浏览器	90+
	Edge浏览器	80+
	Opera浏览器	54+
macOS	Chrome浏览器	67+
	微信内嵌浏览器	-
	Safari浏览器	13+
	Firefox浏览器	90+
	Opera浏览器	56+
Android	微信内嵌浏览器（TBS内核）	-
	微信内嵌浏览器（XWEB内核）	-
	移动版Chrome浏览器	83+
	移动版QQ浏览器	12+

操作系统类型	浏览器类型	浏览器版本
	华为系统浏览器	11.0.8+
iOS	微信内嵌浏览器	iOS 14.3+ 微信6.5+版本
	移动版Chrome浏览器	iOS 14.3+
	移动版Safari浏览器	13+

表 3-2 浏览器使用限制

浏览器类型	使用限制
Chrome浏览器	<ol style="list-style-type: none"> <li>在华为移动设备上，Chrome浏览器（包括华为浏览器）支持WebRTC的版本为91+。</li> <li>Android移动端WebView对WebRTC能力的支持参差不齐，受影响的因素很多，如设备厂家、浏览器内核、版本等，使用的兼容性较差，因此可用性不能保证，不建议使用这类浏览器。</li> </ol>
Safari浏览器	<ul style="list-style-type: none"> <li>Safari 13的用户可能听不到远端用户的声音。</li> <li>iOS Safari 14.2和macOS Safari 14.0.1上音频可能断断续续。</li> </ul>
Firefox浏览器	Apple M1芯片的Mac设备上Firefox不支持H.264编解码。
Opera浏览器	在华为移动设备上，Opera浏览器支持SDK的版本为64+。
其他浏览器	由于Android设备各厂家的浏览器内核、webview、版本等因素，移动端浏览器对WebRTC的支持度不一，除可以使用 <a href="#">表1 浏览器适配详情</a> 中列举的明确支持的浏览器外，还可以集成使用Native SDK（Android / iOS）。

## 3.2 开发前准备

### 前提条件

已下载SDK包。

### 环境要求

- 编译工具推荐安装Microsoft Visual Studio Code 1.43.2或以上版本。
- 如果客户端用Node.js开发，推荐安装14.19.1或以上版本。
- 支持的浏览器详情请参见[浏览器适配详情](#)。
- 如果客户端用TypeScript开发，TypeScript的版本不低于3.8.3。



## SDK 集成

**步骤1** 将[SDK下载](#)到本地，建议将SDK压缩包放置在自己项目的“sdk”目录下。

**步骤2** 在项目代码中引入“HWLLSPlayer”。

- 如果您通过<script>方式引入SDK，则通过访问HWLLSPlayer获取导出的模块：

```
<script src='./sdk/HWLLSPlayer.js'>
  console.log(HWLLSPlayer.getVersion())
</script>
```
- 如果您直接引用SDK静态JS文件，则通过以下方式访问：

```
import HWLLSPlayer from './sdk/HWLLSPlayer'
console.log(HWLLSPlayer.getVersion())
```
- 如果您通过npm模块化的方式引入，首先要安装HWLLSPlayer模块，在package.json的开发依赖里引入HWLLSPlayer，如：“HWLLSPlayer”: “./sdk/HWLLS\_SDK\_Web\_\*.tar.gz”。在终端执行安装命令（版本号按实际替换）：

```
npm install
```

，然后通过以下方式访问：

```
import HWLLSPlayer from 'HWLLSPlayer'
console.log(HWLLSPlayer.getVersion())
```

----结束

## 3.3 SDK 使用

**步骤1** 检测浏览器是否兼容SDK。具体接口详情请参见[checkSystemRequirements](#)。

```
async isBrowserSupport() {
  let check = false
  try {
    check = await HWLLSPlayer.checkSystemRequirements()
    console.warn('browser isSupport: ' + check)
  } catch (error) {
    console.error('check browser isSupport error: ${error.getCode()} - ${error.getMsg()}')
    if (error.getCode() && error.getCode() === 50000006) {
      let commonClient = HWLLSPlayer.createClient("flv") // "hls"
      await commonClient.startPlay(url,options)
    }
  }
  return check
}
```

**步骤2** 创建客户端。具体接口详情请参见[createClient](#)。

```
let client = HWLLSPlayer.createClient("webrtc")
```

**步骤3** 注册事件监听，如播放成功后，客户端会收到[video-start](#)和[audio-start](#)事件通知，如播放失败，客户端会收到[Error](#)事件通知，可根据错误进行业务降级处理。

```
client.on('video-start', () => {
  console.log('video-start')
})

client.on('audio-start', () => {
  console.log('audio-start')
})

client.on('Error', (error) => {
  if (error.errCode === 50000007 || error.errCode === 50000022) {
    let commonClient = HWLLSPlayer.createClient("flv") // "hls"
    await commonClient.startPlay(url,options)
  }
})
```

**步骤4** 开始播放。具体接口详情请参见[startPlay](#)。

```
async startPlay() {
  try{
    await client.startPlay(url,options)
    console.log('startPlay success')
  } catch(error){
    console.log('startPlay fail',error)
    if (error.getCode && error.getCode() === 50000021) {
      let commonClient = HWLLSPlayer.createClient("flv") // "hls"
      await commonClient.startPlay(url,options)
    }
  }
}
```

- url: 必选, string类型, 拉流地址。具体地址格式请参见[startPlay](#)详情。
- options: 可选, StartPlayOptions类型。

StartPlayOptions定义如下: {

- elementId: 必选, 播放Dom标识ID。
- objectFit: 可选, string类型, 默认值为cover。支持的枚举值如下:
  - contain: 优先保证视频内容全部显示。视频尺寸等比缩放, 直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致, 在保持长宽比的前提下, 将视频进行缩放后填满视窗, 缩放后的视频四周会有一圈黑边。
  - cover: 优先保证视窗被填满。视频尺寸等比缩放, 直至整个视窗被视频填满。如果视频长宽与显示窗口不同, 则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
  - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
- muted: 可选, boolean类型, true表示静音, false表示不静音。默认值为false。
- sessionId: 可选, string类型, 一次完整会话的统一标识。
- showLoading: 可选, boolean类型, true表示开启loading的展示效果, 默认值为false。当该参数设置为true时, 起播loading效果同步开启, 播放过程中发生缓冲时loading的效果, 需根据[setParameter](#)接口中的LOADING\_CONFIG进行设置。
- autoPlay: 可选, boolean类型, true表示开启自动起播功能, false表示非自动起播, 需要人为触发播放, 默认为true。
- poster: 可选, 对象定义如下: {
  - url: 可选, string类型。设置播放封面图片完整地址, 图片格式限JPGPNG和静态GIF格式, 大小不超过1MB, 尺寸不超过1920 x 1080, 文件名不得含有中文字符。
  - mode: 可选, string类型。默认值为cover。支持的枚举值如下: {
    - fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
    - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。

- `startEnable`: 可选, `boolean`类型。启动播放时是否展示播放封面, `true`表示展示, `false`表示不展示播放封面, 默认值`false`。该参数只在设置非自动播放场景下生效。
- `pauseEnable`: 可选, `boolean`类型。触发暂停操作时, 是否在播放页面展示播放封面, `true`表示展示播放封面, `false`表示不展示, 默认值`false`。
- }
- `webrtcConfig`: 可选, `WebRTCConfig`类型。指定媒体类型进行拉流的配置参数, `WebRTCConfig`定义如下: {
  - `receiveVideo`: 可选, `boolean`类型。设置是否拉取视频进行播放, `true`表示拉取视频进行播放, `false`表示不拉取视频播放, 默认值`true`。该属性值和`receiveAudio`不能同时设置为`false`。
  - `receiveAudio`: 可选, `boolean`类型。设置是否拉取音频进行播放, `true`表示拉取音频进行播放, `false`表示不拉取音频播放, 默认值`true`。该属性值和`receiveVideo`不能同时设置为`false`。
- }
- `schedulePolicy`: 可选, `SchedulePolicy`类型。指定接入调度策略, `SchedulePolicy`定义如下: {
  - `DNS`: 可选, `string`类型。表示域名DNS解析接入, 默认值"`DNS`"。
  - `HTTPDNS`: 可选, `string`类型。表示使用HTTPDNS接入域名。
- }
- `domainPolicy`: 可选, `DomainPolicy`类型。指定接入域名的策略, 该设置仅在`schedulePolicy`参数为"`DNS`"模式时生效, `DomainPolicy`定义如下: {
  - `0`: 可选, `number`类型。表示使用用户自定义域名, 默认值`0`。
  - `1`: 可选, `number`类型。表示使用公共接入域名。
- }

**步骤5** 停止播放。具体接口详情请参见[stopPlay](#)。

```
stopPlay() {  
  try {  
    client.stopPlay()  
    console.log('stopPlay success')  
  } catch (error) {  
    console.log('stopPlay fail', error)  
  }  
}
```

**步骤6** 销毁客户端对象。具体接口详情请参见[destoryClient](#)。

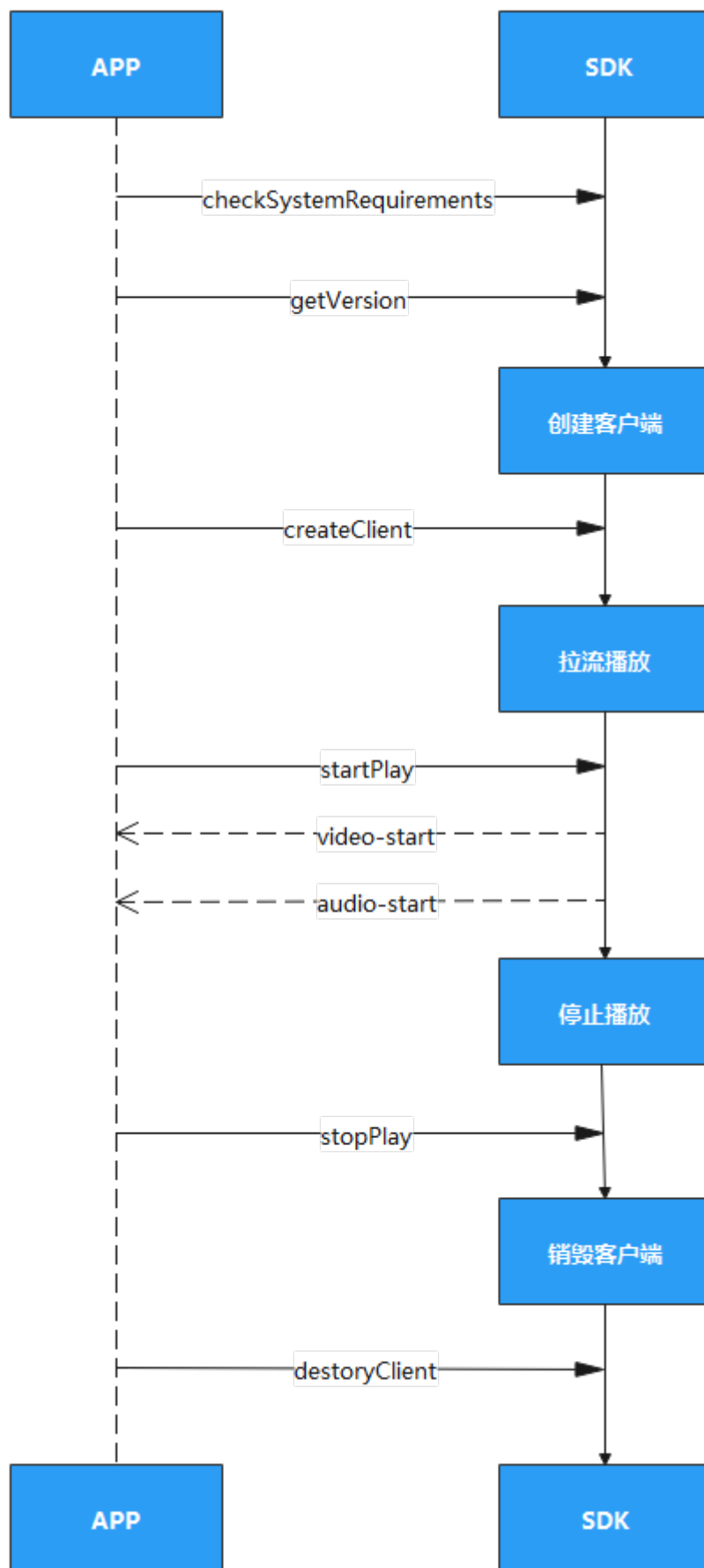
```
client.destoryClient()
```

----结束

## 3.4 基本使用逻辑

主业务流程包括:

- 预处理：获取浏览器的版本号、检查兼容性、创建客户端等。
- 拉流播放：拉流播放请求。
- 停止播放：停止播放请求。
- 后处理：销毁客户端等。



 说明

上图中，单击相应接口名称可快速跳转到相应接口位置查看其使用方法。

## 3.5 接口参考

### 3.5.1 主入口（HWLLSPlayer）

本章节介绍了低时延直播Web SDK的HWLLSPlayer接口详情。

表 3-3 主入口接口

接口	描述
<a href="#">checkSystemRequirements</a>	检测浏览器是否支持低时延直播Web SDK。
<a href="#">getVersion</a>	获取SDK版本号。
<a href="#">createClient</a>	创建一个直播拉流客户端对象，如果需要拉取多个直播流则需要创建多个客户端对象。
<a href="#">uploadLog</a>	上传日志。
<a href="#">saveLog</a>	保存日志
<a href="#">setParameter</a>	设置全局配置参数。
<a href="#">setLogLevel</a>	设置Console上打印的日志级别。
<a href="#">setReportConfig</a>	设置打点能力以及打点和日志上传的认证策略。

#### checkSystemRequirements

checkSystemRequirements(): Promise<boolean>

##### 【功能说明】

检测浏览器是否支持低时延直播Web SDK。

##### 【请求参数】

无

##### 【返回参数】

Promise<boolean>：返回一个Promise对象，true表示浏览器兼容低时延直播Web SDK。如果不兼容，则返回对应Error异常。

 注意

由于低延时直播需使用WebRTC能力，存在部分浏览器不支持播放WebRTC的情况，可根据特定错误码（HWLLS\_ERROR\_WEBRTC\_UNSUPPORTED），进行播放降级，请参考[SDK使用](#)。

## getVersion

getVersion(): string

### 【功能说明】

获取当前SDK版本号。

### 【请求参数】

无

### 【返回参数】

string: SDK当前版本号。

## createClient

createClient(type: string): HWLLSClient | HWFlvClient | HWHlsClient

### 【功能说明】

创建一个直播拉流客户端对象，如果需要拉取多个直播流则需要创建多个客户端对象。

### 【请求参数】

type: string类型，可选。创建的拉流客户端类型。

- 低时延直播拉流客户端类型：webrtc。
- flv直播拉流客户端类型：flv。
- HLS直播拉流客户端类型：hls（预留，暂未开放）。

缺省值：webrtc。

### 【返回参数】

client: 拉流客户端对象。

## uploadLog

async uploadLog(): Promise<void>

### 【功能说明】

上传日志。

### 【请求参数】

无

### 【返回参数】

Promise<void>: 使用tryCatch获取error为数组，返回多个appid对应错误信息。

## saveLog

async saveLog(): Promise<Blob>

### 【功能说明】

为用户提供灵活保存日志的功能

【请求参数】

无

【返回参数】

Promise<Blob>: 经过zip压缩好的Promise<Blob>,用户可以直接保存为zip文件

## setParameter

setParameter(parameterKey: string, parameterValue: any): boolean

【功能说明】

设置全局配置参数。

【请求参数】

参数名称	参数值
LOADING_CONFIG	LoadingConfig类型, 定义如下: { netQualityLoading: 可选, boolean类型。true表示开启根据网络质量进行loading效果展示, 默认值为false, 关闭。 netQualityLoadingThreshold: 可选, number类型。展示loading效果的网络质量 ( <b>network-quality</b> ) 的阈值, 默认网络质量等级为5。 frameStuckLoading: 可选, boolean类型。true表示开启根据帧卡顿时长进行loading效果展示, 默认值为false frameStuckThreshold: 可选, number类型。展示loading效果帧卡顿时长的阈值, 单位为100ms。默认值为10, 表示帧卡顿时长为1000ms。 } <b>注意</b> 需要在 <b>起播</b> 之前进行设置。
DNS_QUERY_ENABLE	boolean类型, 可选, 默认为false, true表示开启DNS结果解析, false表示关闭DNS结果解析。
ACCESS_DOMAIN	string类型, 可选, 默认为空, 主要用于拉流环境配置, 沟通华为工程师填入。
GLSB_DOMAIN	string类型, 可选, 默认为空, 主要用于GSLB环境配置, 沟通华为工程师填入。

【返回参数】

boolean: 配置参数设置结果。true表示参数设置成功, false表示参数设置失败。

## setLogLevel

setLogLevel(level: string): boolean



### 【功能说明】

设置Console上打印的日志级别，如不设置日志级别，则console日志打印级别默认为error。

### 【请求参数】

level: string类型，必选，日志级别标识。

- none: 关闭全部级别的日志打印。
- error: 打印error级别日志。
- warn: 打印warn级别及更高级别日志。
- info: 打印info级别及更高级别日志。
- debug: 打印debug级别及更高级别日志。

### 【返回参数】

boolean: 设置日志级别结果。true表示日志级别设置成功，false表示日志级别设置失败。

## setReportConfig

```
setReportConfig(reportConfig:ReportConfig):boolean
```

### 【功能说明】

设置打点能力以及打点和日志上传的认证策略。

### 【请求参数】

reportConfig: ReportConfig类型，必选。ReportConfig定义如下所示：

- enable: 必选，布尔类型，true表示开启打点，false表示关闭打点。默认true。
- tokenConfig: 可选，对象定义如下所示：
  - enable: 布尔类型，true表示开启认证，false表示关闭认证。默认false。
  - tokenInfo: 数组类型，数组内部ReportTokenInfo类。ReportTokenInfo定义如下所示：
    - appid: string类型。传入appid。
    - expTimestamp: string类型。过期时间戳，系统当前UNIX时间戳加上鉴权过期时间（推荐7200秒，最长需要小于43200秒，即12个小时）。  
例如：当前UNIX时间戳为：1708531200，鉴权过期时间自定义为7200秒，那么过期的时间戳为：1708538400，即表示该校验字符串在2024-02-22 02:00:00过期。
    - token: string类型。**hmac\_sha256**生成的字符串。**hmac\_sha256(共享密钥, 过期的时间戳 + appID)**。共享密钥由用户控制获取。

### 【返回参数】

返回值布尔值，true表示设置成功，false表示设置失败。

开启了认证策略，实际请求状态会通过on函数注册Error回调获得。

## 3.5.2 客户端对象（HWLLSClient）

本章节介绍了低时延直播Web SDK的HWLLSClient接口详情。

表 3-4 主入口接口

接口	描述
<b>startPlay</b>	开始播放，客户端根据输入的URL到服务端拉取对应的主播流。
<b>switchPlay</b>	快速切换下一路流播放。
<b>stopPlay</b>	停止播放。
<b>replay</b>	重新播放。
<b>startPlayerCustomize</b>	自定义播放
<b>resume</b>	恢复播放。
<b>pause</b>	暂停播放。
<b>pauseVideo</b>	暂停视频。
<b>resumeVideo</b>	恢复视频。
<b>pauseAudio</b>	暂停音频。
<b>resumeAudio</b>	恢复音频。
<b>setPlayoutVolume</b>	设置播放音量。
<b>getPlayoutVolume</b>	获取音频音量。
<b>muteAudio</b>	静音
<b>streamStatistic</b>	设置是否开启流信息统计。
<b>enableStreamStateDetection</b>	开启/关闭音视频码流状态探测功能。
<b>on</b>	注册客户端对象事件回调接口。
<b>off</b>	取消注册客户端对象事件回调接口。
<b>destoryClient</b>	销毁客户端对象。
<b>fullScreenToggle</b>	开启关闭全屏

### startPlay

startPlay(url: string, options: StartPlayOptions): Promise<void>

#### 【功能说明】

开始播放，客户端根据输入的URL到服务端拉取对应的主播流。

#### 【请求参数】

- url: string类型, 必选。拉流URL, 格式如: webrtc://{domain}/{AppName}/{StreamName}。
  - webrtc://: 固定不变, 表示使用webrtc方式拉流。
  - domain: 拉流域名, 使用在华为云注册的拉流域名。
  - AppName: 应用名, 使用在华为云注册的应用名。
  - StreamName: 流名, 和推流的流名保持一致。
- options: 必选, StartPlayOptions类型。播放配置参数, StartPlayOptions定义如下: {
  - elementId: 必选, 播放DOM标识ID。
  - objectFit: 可选, string类型, 默认值为cover。支持的枚举值如下:
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放, 直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致, 在保持长宽比的前提下, 将视频进行缩放后填满视窗, 缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放, 直至整个视窗被视频填满。如果视频长宽与显示窗口不同, 则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
  - muted: 可选, boolean类型, true表示静音, false表示不静音。默认值为false。
  - sessionId: 可选, string类型, 一次完整会话的统一标识。
  - showLoading: 可选, boolean类型, true表示开启loading的展示效果, 默认为false。当该参数设置为true时, 起播loading效果同步开启, 播放过程中发生缓冲时loading的效果, 需根据[setParameter](#)接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选, boolean类型, true表示开启自动起播功能, false表示非自动起播, 需要人为触发播放, 默认为true。
  - poster: 可选, 对象定义如下: {
    - url: 可选, string类型。设置播放封面图片完整地址, 图片格式限JPGPNG和静态GIF格式, 大小不超过1MB, 尺寸不超过1920 x 1080, 文件名不得含有中文字符。
    - mode: 可选, string类型。默认值为cover。支持的枚举值如下: {
      - fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
      - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。
  - startEnable: 可选, boolean类型。启动播放时是否展示播放封面, true表示展示, false表示不展示播放封面, 默认值false。该参数只在设置非自动播放场景下生效。

- pauseEnable: 可选, boolean类型。触发暂停操作时, 是否在播放页面展示播放封面, true表示展示播放封面, false表示不展示, 默认值false。
- }
- webrtcConfig: 可选, WebRTCConfig类型。指定媒体类型进行拉流的配置参数, WebRTCConfig定义如下: {
  - receiveVideo: 可选, boolean类型。设置是否拉取视频进行播放, true表示拉取视频进行播放, false表示不拉取视频播放, 默认值true。该属性值和receiveAudio不能同时设置为false。
  - receiveAudio: 可选, boolean类型。设置是否拉取音频进行播放, true表示拉取音频进行播放, false表示不拉取音频播放, 默认值true。该属性值和receiveVideo不能同时设置为false。
- }
- schedulePolicy: 可选, SchedulePolicy类型。指定接入调度策略, SchedulePolicy定义如下: {
  - DNS: 可选, string类型。表示域名DNS解析接入, 默认值"DNS"。
  - HTTPDNS: 可选, string类型。表示使用HTTPDNS接入域名。
- }
- domainPolicy: 可选, DomainPolicy类型。指定接入域名的策略, 该设置仅在schedulePolicy参数为"DNS"模式时生效, DomainPolicy定义如下: {
  - 0: 可选, number类型。表示使用用户自定义域名, 默认值0。
  - 1: 可选, number类型。表示使用公共接入域名。
- }
- downgradeUrl: 可选, 对象定义如下: {
  - hlsUrl?: 可选, string类型。标识降级hls播放地址。
  - flvUrl?: 可选, string类型。标识降级flv播放地址。
- }

#### 【返回参数】

Promise<void>: 返回一个Promise对象。

#### 注意

在低延时直播服务发生故障的场景下, 可根据特定错误码 (HWLLS\_BUSSINESS\_DOWNGRADE), 进行播放降级, 请参考[SDK使用](#)。

## switchPlay

switchPlay(url: string, options: StartPlayOptions): Promise<void>

#### 【功能说明】

起播成功后, 快速切换下一路流播放。

### 【请求参数】

- url: 必选, string类型。拉流URL, 格式如: webrtc://{domain}/{AppName}/{StreamName}。
  - webrtc://: 固定不变, 表示使用webrtc方式拉流。
  - domain: 拉流域名, 使用在华为云注册的拉流域名。
  - AppName: 应用名, 使用在华为云注册的应用名。
  - StreamName: 流名, 和推流的流名保持一致。
- options: 可选, StartPlayOptions类型。播放配置参数, 如果不携带该参数, 则复用首次起播携带的options数据。StartPlayOptions定义如下: {
  - elementId: 必选, 播放DOM标识ID。
  - objectFit: 可选, string类型, 默认值为cover。支持的枚举值如下:
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放, 直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致, 在保持长宽比的前提下, 将视频进行缩放后填满视窗, 缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放, 直至整个视窗被视频填满。如果视频长宽与显示窗口不同, 则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
  - muted: 可选, boolean类型, true表示静音, false表示不静音。默认值为false。
  - sessionId: 可选, string类型, 一次完整会话的统一标识。
  - showLoading: 可选, boolean类型, true表示开启loading的展示效果, 默认为false。当该参数设置为true时, 起播loading效果同步开启, 播放过程中发生缓冲时loading的效果, 需根据setParameter接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选, boolean类型, true表示开启自动起播功能, false表示非自动起播, 需要人为触发播放, 默认为true。
  - poster: 可选, 对象定义如下: {
    - url: 可选, string类型。设置播放封面图片完整地址, 图片格式限JPGPNG和静态GIF格式, 大小不超过1MB, 尺寸不超过1920 x 1080, 文件名不得含有中文字符。
    - mode: 可选, string类型。默认值为cover。支持的枚举值如下: {
      - fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
      - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。
  - startEnable: 可选, boolean类型。启动播放时是否展示播放封面, true表示展示, false表示不展示播放封面, 默认值false。该参数只在设置非自动播放场景下生效。

- pauseEnable: 可选, boolean类型。触发暂停操作时, 是否在播放页面展示播放封面, true表示展示播放封面, false表示不展示, 默认值false。
- }
- webrtcConfig: 可选, WebRTCConfig类型。指定媒体类型进行拉流的配置参数, WebRTCConfig定义如下: {
  - receiveVideo: 可选, boolean类型。设置是否拉取视频进行播放, true表示拉取视频进行播放, false表示不拉取视频播放, 默认值true。该属性值和receiveAudio不能同时设置为false。
  - receiveAudio: 可选, boolean类型。设置是否拉取音频进行播放, true表示拉取音频进行播放, false表示不拉取音频播放, 默认值true。该属性值和receiveVideo不能同时设置为false。
- }
- schedulePolicy: 可选, SchedulePolicy类型。指定接入调度策略, SchedulePolicy定义如下: {
  - DNS: 可选, string类型。表示域名DNS解析接入, 默认值"DNS"。
  - HTTPDNS: 可选, string类型。表示使用HTTPDNS接入域名。
- }
- domainPolicy: 可选, DomainPolicy类型。指定接入域名的策略, 该设置仅在schedulePolicy参数为"DNS"模式时生效, DomainPolicy定义如下: {
  - 0: 可选, number类型。表示使用用户自定义域名, 默认值0。
  - 1: 可选, number类型。表示使用公共接入域名。
- }

#### 【返回参数】

Promise<void>: 返回一个Promise对象。

#### 注意

在低延时直播服务发生故障的场景下, 可根据特定错误码 (HWLLS\_BUSSINESS\_DOWNGRADE), 进行播放降级, 请参考[SDK使用](#)。

## stopPlay

stopPlay(): boolean

#### 【功能说明】

停止播放。

#### 【请求参数】

无

#### 【返回参数】

boolean: 停止播放结果。true表示成功, false表示失败。

## replay

replay(): Promise<boolean>

### 【功能说明】

重新播放。

### 【请求参数】

无

### 【返回参数】

Promise<boolean>: 重新播放结果, true表示成功, false表示失败。

## startPlayerCustomize

startPlayerCustomize(url: string): Promise<boolean>

### 【功能说明】

自定义播放。

### 【请求参数】

- url: 必选, string类型。拉流URL, 自定义URL播放
- options: 可选, StartPlayOptions类型。播放配置参数, 如果不携带该参数, 则复用首次起播携带的options数据。StartPlayOptions定义如下: {
  - elementId: 必选, 播放DOM标识ID。
  - objectFit: 可选, string类型, 默认值为cover。支持的枚举值如下:
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放, 直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致, 在保持长宽比的前提下, 将视频进行缩放后填满视窗, 缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放, 直至整个视窗被视频填满。如果视频长宽与显示窗口不同, 则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
  - muted: 可选, boolean类型, true表示静音, false表示不静音。默认值为false。
  - sessionId: 可选, string类型, 一次完整会话的统一标识。
  - showLoading: 可选, boolean类型, true表示开启loading的展示效果, 默认为false。当该参数设置为true时, 起播loading效果同步开启, 播放过程中发生缓冲时loading的效果, 需根据setParameter接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选, boolean类型, true表示开启自动起播功能, false表示非自动起播, 需要人为触发播放, 默认为true。
  - poster: 可选, 对象定义如下: {
    - url: 可选, string类型。设置播放封面图片完整地址, 图片格式限JPGPNG和静态GIF格式, 大小不超过1MB, 尺寸不超过1920 x 1080, 文件名不得含有中文字符。

- mode: 可选, string类型。默认值为cover。支持的枚举值如下: {
    - fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不匹配, 那么视频将被拉伸以适应视窗。
    - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。}
  - startEnable: 可选, boolean类型。启动播放时是否展示播放封面, true表示展示, false表示不展示播放封面, 默认值false。该参数只在设置非自动播放场景下生效。
  - pauseEnable: 可选, boolean类型。触发暂停操作时, 是否在播放页面展示播放封面, true表示展示播放封面, false表示不展示, 默认值false。
- }
- webrtcConfig: 可选, WebRTCConfig类型。指定媒体类型进行拉流的配置参数, WebRTCConfig定义如下: {
    - receiveVideo: 可选, boolean类型。设置是否拉取视频进行播放, true表示拉取视频进行播放, false表示不拉取视频播放, 默认值true。该属性值和receiveAudio不能同时设置为false。
    - receiveAudio: 可选, boolean类型。设置是否拉取音频进行播放, true表示拉取音频进行播放, false表示不拉取音频播放, 默认值true。该属性值和receiveVideo不能同时设置为false。}
  - schedulePolicy: 可选, SchedulePolicy类型。指定接入调度策略, SchedulePolicy定义如下: {
    - DNS: 可选, string类型。表示域名DNS解析接入, 默认值"DNS"。
    - HTTPDNS: 可选, string类型。表示使用HTTPDNS接入域名。}
  - domainPolicy: 可选, DomainPolicy类型。指定接入域名的策略, 该设置仅在schedulePolicy参数为"DNS"模式时生效, DomainPolicy定义如下: {
    - 0: 可选, number类型。表示使用用户自定义域名, 默认值0。
    - 1: 可选, number类型。表示使用公共接入域名。}

#### 【返回参数】

Promise<void>: 返回一个Promise对象。

## resume

resume(): Promise<boolean>

#### 【功能说明】

恢复播放。

#### 【请求参数】



无

**【返回参数】**

Promise<boolean>: 恢复音视频播放结果, true表示成功, false表示失败。

## pause

pause(): boolean

**【功能说明】**

暂停音视频播放。

**【请求参数】**

无

**【返回参数】**

boolean: 暂停播放结果, true表示成功, false表示失败。

## pauseVideo

pauseVideo(): boolean

**【功能说明】**

暂停视频, 暂停后卡在当前画面。

**【请求参数】**

无

**【返回参数】**

boolean: 暂停视频播放结果, true表示成功, false表示失败。

## resumeVideo

resumeVideo(): Promise<boolean>

**【功能说明】**

恢复视频播放。

**【请求参数】**

无

**【返回参数】**

Promise<boolean>: 恢复视频播放结果, true表示成功, false表示失败。

## pauseAudio

pauseAudio(): boolean

**【功能说明】**

暂停音频。

**【请求参数】**

无

**【返回参数】**

boolean: 暂停音频播放结果, true表示成功, false表示失败。

## resumeAudio

resumeAudio(): Promise<boolean>

**【功能说明】**

恢复音频。

**【请求参数】**

无

**【返回参数】**

Promise<boolean>: 恢复音频播放结果, true表示成功, false表示失败。

## setPlayoutVolume

setPlayoutVolume(volume: number): boolean

**【功能说明】**

设置音频音量, iOS 不支持。

**【请求参数】**

volume: 必选, number类型, 取值范围为[0,100], 音频的音量值。

**【返回参数】**

boolean: 设置音频音量是否成功, true表示成功, false表示失败。

## getPlayoutVolume

getPlayoutVolume(): number

**【功能说明】**

获取音频音量。

**【请求参数】**

无

**【返回参数】**

number: 音量值, 取值范围为[0,100]。

## muteAudio

muteAudio(isMute: boolean): void

**【功能说明】**

静音。

**【请求参数】**

- isMute: 必选, boolean类型, 是否静音, true表示静音, false表示取消静音。

**【返回参数】**

无

## streamStatistic

streamStatistic(enable: boolean, interval: number): void

**【功能说明】**

设置是否开启流信息统计。

**【请求参数】**

- enable: 必选, boolean类型, 是否开启流信息统计, true表示开启统计。
- interval: 必选, number类型, 设置统计间隔, 单位为秒, 取值范围为[1, 60], 默认值为1。

**【返回参数】**

无

## enableStreamStateDetection

enableStreamStateDetection(enable: boolean, interval: number, interruptRetry: StreamInterruptRetry): boolean

**【功能说明】**

开启/关闭音、视频码流状态探测功能, 开启后可探测推流侧是否处于断流的状态。

**【请求参数】**

- enable: 必选, boolean类型, true表示开启音视频码流状态探测, false表示关闭音视频码流状态探测。默认值为false。
- interval: 必选, number类型, 单位为秒, 取值范围为[1,60]。音视频无码流状态的判断时间。默认值为3, 推荐设置为3。
- interruptRetry: 可选, StreamInterruptRetry类型。断流重试播放配置参数, StreamInterruptRetry定义为: {  
enable: boolean类型, 开启断流后尝试自动恢复播放。默认值为false, 即不开启自动重试。  
retryInterval: number类型, 拉流播放的重试周期, 单位为秒。最小值10, 最大值建议不超过60, 默认值为30。  
retryTimes: number类型, 尝试重新恢复播放的最大重试次数。最小值1, 默认值为30。  
}

**【返回参数】**

boolean: 是否成功, true表示成功, false表示失败。

## on

on(event: string, handler: function, withTimeout?: boolean): void

**【功能说明】**

注册客户端对象事件回调接口。

**【请求参数】**

- event: 必选, string类型, 事件名称。具体请参见[HWLLSClientEvent](#)。
- handler: 必选, function类型, 事件处理方法。
- withTimeout: 选填, boolean类型, 是否超时报错

**【返回参数】**

无

## off

```
off(event: string, handler: function): void
```

**【功能说明】**

取消注册客户端对象事件回调接口。

**【请求参数】**

- event: 必选, string类型, 事件名称。具体请参见[HWLLSClientEvent](#)。
- handler: 必选, function类型, 事件处理方法。

**【返回参数】**

无

## destoryClient

```
destoryClient(): void
```

**【功能说明】**

销毁客户端对象。

**【请求参数】**

无

**【返回参数】**

无

## fullScreenToggle

```
fullScreenToggle(isExit: boolean): void
```

**【功能说明】**

开启关闭全屏。

**【请求参数】**

isExit: 必选, 布尔类型, 默认false。

**【返回参数】**

无

### 3.5.3 客户端对象 ( HWFlvClient )

本章节介绍了低时延直播Web SDK的HWFlvClient接口详情。

表 3-5 主入口接口

接口	描述
<a href="#">startPlay</a>	开始播放，客户端根据输入的URL到服务端拉取对应的主播流。
<a href="#">switchPlay</a>	快速切换下一路流播放。
<a href="#">stopPlay</a>	停止播放。
<a href="#">replay</a>	重新播放。
<a href="#">resume</a>	恢复播放。
<a href="#">pause</a>	暂停播放。
<a href="#">pauseVideo</a>	暂停视频。
<a href="#">resumeVideo</a>	恢复视频。
<a href="#">pauseAudio</a>	暂停音频。
<a href="#">resumeAudio</a>	恢复音频。
<a href="#">setPlayoutVolume</a>	设置播放音量。
<a href="#">getPlayoutVolume</a>	获取音频音量。
<a href="#">muteAudio</a>	静音。
<a href="#">streamStatistic</a>	设置是否开启流信息统计。
<a href="#">enableStreamStateDetection</a>	开启/关闭音视频码流状态探测功能。
<a href="#">destoryClient</a>	销毁客户端对象。
<a href="#">fullScreenToggle</a>	开启关闭全屏。

#### startPlay

```
startPlay(url: string, options: StartPlayOptions): Promise<void>
```

##### 【功能说明】

开始播放，客户端根据输入的URL到服务端拉取对应的主播流。

##### 【请求参数】

- url：必选，string类型。拉流URL，是以flv结尾的播放地址。
- options：可选，StartPlayOptions类型。播放配置参数，如果不携带该参数，则复用首次起播携带的options数据。StartPlayOptions定义如下：{

- elementId: 必选, 播放DOM标识ID。
  - objectFit: 可选, string类型, 默认值为cover。支持的枚举值如下:
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放, 直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致, 在保持长宽比的前提下, 将视频进行缩放后填满视窗, 缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放, 直至整个视窗被视频填满。如果视频长宽与显示窗口不同, 则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
  - muted: 可选, boolean类型, true表示静音, false表示不静音。默认值为false。
  - sessionId: 不需要传。
  - showLoading: 可选, boolean类型, true表示开启loading的展示效果, 默认为false。当该参数设置为true时, 起播loading效果同步开启, 播放过程中发生缓冲时loading的效果, 需根据setParameter接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选, boolean类型, true表示开启自动起播功能, false表示非自动起播, 需要人为触发播放, 默认为true。
  - poster: 可选, 对象定义如下: {
    - url: 可选, string类型。设置播放封面图片完整地址, 图片格式限JPGPNG和静态GIF格式, 大小不超过1MB, 尺寸不超过1920 x 1080, 文件名不得含有中文字符。
    - mode: 可选, string类型。默认值为cover。支持的枚举值如下: {
      - fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
      - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。}
  - startEnable: 可选, boolean类型。启动播放时是否展示播放封面, true表示展示, false表示不展示播放封面, 默认值false。该参数只在设置非自动播放场景下生效。
  - pauseEnable: 可选, boolean类型。触发暂停操作时, 是否在播放页面展示播放封面, true表示展示播放封面, false表示不展示, 默认值false。
- webrtcConfig: 不需要传。
- schedulePolicy: 不需要传。
- domainPolicy: 不需要传。
- downgradeUrl: 不需要传。

#### 【返回参数】

Promise<void>: 返回一个Promise对象。

## switchPlay

```
switchPlay(url: string, options: StartPlayOptions): Promise<void>
```

### 【功能说明】

起播成功后，快速切换下一路流播放。

### 【请求参数】

- url: 必选，string类型。拉流URL，是以flv结尾的播放地址。
- options: 可选，StartPlayOptions类型。播放配置参数，如果不携带该参数，则复用首次起播携带的options数据。StartPlayOptions定义如下：
  - elementId: 必选，播放DOM标识ID。
  - objectFit: 可选，string类型，默认值为cover。支持的枚举值如下：
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放，直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致，在保持长宽比的前提下，将视频进行缩放后填满视窗，缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放，直至整个视窗被视频填满。如果视频长宽与显示窗口不同，则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配，那么视频将被拉伸以适应视窗。
  - muted: 可选，boolean类型，true表示静音，false表示不静音。默认值为false。
  - sessionId: 不需要传。
  - showLoading: 可选，boolean类型，true表示开启loading的展示效果，默认为false。当该参数设置为true时，起播loading效果同步开启，播放过程中发生缓冲时loading的效果，需根据[setParameter](#)接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选，boolean类型，true表示开启自动起播功能，false表示非自动起播，需要人为触发播放，默认为true。
  - poster: 可选，对象定义如下：
    - url: 可选，string类型。设置播放封面图片完整地址，图片格式限JPGPNG和静态GIF格式，大小不超过1MB，尺寸不超过1920 x 1080，文件名不得含有中文字符。
    - mode: 可选，string类型。默认值为cover。支持的枚举值如下：
      - fill: 视频内容完全填充视窗，如果视频的宽高比与视窗不相匹配，那么视频将被拉伸以适应视窗。
      - crop: 播放封面原始尺寸大小展示，如果海报超出播放区域，则会对超出部分进行裁剪，否则在播放窗口居中展示。

- `startEnable`: 可选, `boolean`类型。启动播放时是否展示播放封面, `true`表示展示, `false`表示不展示播放封面, 默认值`false`。该参数只在设置非自动播放场景下生效。
- `pauseEnable`: 可选, `boolean`类型。触发暂停操作时, 是否在播放页面展示播放封面, `true`表示展示播放封面, `false`表示不展示, 默认值`false`。

}

- `webrtcConfig`: 不需要传。
- `schedulePolicy`: 不需要传。
- `domainPolicy`: 不需要传。
- `downgradeUrl`: 不需要传。

#### 【返回参数】

`Promise<void>`: 返回一个`Promise`对象。

## stopPlay

`stopPlay(): boolean`

#### 【功能说明】

停止播放。

#### 【请求参数】

无

#### 【返回参数】

`boolean`: 停止播放的结果。`true`表示成功, `false`表示失败。

## replay

`replay(): Promise<boolean>`

#### 【功能说明】

重新播放。

#### 【请求参数】

无

#### 【返回参数】

`Promise<boolean>`: 重新播放的结果。`true`表示成功, `false`表示失败。

## resume

`resume(): Promise<boolean>`

#### 【功能说明】

恢复播放。

#### 【请求参数】

无



**【返回参数】**

Promise<boolean>: 恢复音视频播放的结果。true表示成功，false表示失败。

## pause

pause(): boolean

**【功能说明】**

暂停音视频播放。

**【请求参数】**

无

**【返回参数】**

boolean: 暂停播放的结果。true表示成功，false表示失败。

## pauseVideo

pauseVideo(): boolean

**【功能说明】**

该接口不支持。

**【请求参数】**

无

**【返回参数】**

boolean: 只返回false。

## resumeVideo

resumeVideo(): Promise<boolean>

**【功能说明】**

该接口不支持。

**【请求参数】**

无

**【返回参数】**

boolean: 只返回false。

## pauseAudio

pauseAudio(): boolean

**【功能说明】**

暂停音频。

**【请求参数】**

无

#### 【返回参数】

boolean：暂停音频播放的结果。true表示成功，false表示失败。

## resumeAudio

resumeAudio(): Promise<boolean>

#### 【功能说明】

恢复音频。

#### 【请求参数】

无

#### 【返回参数】

Promise<boolean>：恢复音频播放的结果。true表示成功，false表示失败。

## setPlayoutVolume

setPlayoutVolume(volume: number): boolean

#### 【功能说明】

设置音频音量，会开启声音。

#### 【请求参数】

volume：必选，number类型，音频的音量值。取值范围为[0,100]。

#### 【返回参数】

boolean：设置音频音量是否成功。true表示成功，false表示失败。

## getPlayoutVolume

getPlayoutVolume(): number

#### 【功能说明】

获取音频音量。

#### 【请求参数】

无

#### 【返回参数】

number：音量值，取值范围为[0,100]。

## muteAudio

muteAudio(isMute: boolean): void

#### 【功能说明】

静音。

#### 【请求参数】

isMute：必选，boolean类型，是否静音。true表示静音，false表示取消静音。

#### 【返回参数】

无

### streamStatistic

```
streamStatistic(enable: boolean, interval: number): void
```

#### 【功能说明】

设置是否开启流信息统计。

#### 【请求参数】

- enable: 必选, boolean类型, 是否开启流信息统计, true表示开启统计。
- interval: 必选, number类型, 设置统计间隔, 单位为秒, 取值范围为[1, 60], 默认值为1。

#### 【返回参数】

无

### enableStreamStateDetection

```
enableStreamStateDetection(enable: boolean, interval: number, interruptRetry:StreamInterruptRetry):  
boolean
```

#### 【功能说明】

开启/关闭音、视频码流状态探测功能, 开启后可探测推流侧是否处于断流的状态。

#### 【请求参数】

- enable: 必选, boolean类型, true表示开启音视频码流状态探测, false表示关闭音视频码流状态探测。默认值为false。
- interval: 必选, number类型, 单位为秒, 取值范围为[1,60]。音视频无码流状态的判断时间。默认值为3, 推荐设置为3。
- interruptRetry: 可选, StreamInterruptRetry类型。断流重试播放配置参数, StreamInterruptRetry定义为: {  
enable: boolean类型, 开启断流后尝试自动恢复播放。默认值为false, 即不开启自动重试。  
retryInterval: number类型, 拉流播放的重试周期, 单位为秒。最小值10, 最大值建议不超过60, 默认值为30。  
retryTimes: number类型, 尝试重新恢复播放的最大重试次数。最小值1, 默认值为30。  
}

#### 【返回参数】

boolean: 是否成功, true表示成功, false表示失败。

### destoryClient

```
destoryClient(): void
```

#### 【功能说明】

销毁客户端对象。

【请求参数】

无

【返回参数】

无

## fullScreenToggle

fullScreenToggle(isExit: boolean): void

【功能说明】

开启关闭全屏。

【请求参数】

isExit: 必选，布尔类型，默认false。

【返回参数】

无

## 3.5.4 客户端对象（HWHlsClient）

本章节介绍了低时延直播Web SDK的HWHlsClient接口详情。

表 3-6 主入口接口

接口	描述
<a href="#">startPlay</a>	开始播放，客户端根据输入的URL到服务端拉取对应的主播流。
<a href="#">switchPlay</a>	快速切换下一路流播放。
<a href="#">stopPlay</a>	停止播放。
<a href="#">replay</a>	重新播放。
<a href="#">resume</a>	恢复播放。
<a href="#">pause</a>	暂停播放。
<a href="#">pauseVideo</a>	暂停视频。
<a href="#">resumeVideo</a>	恢复视频。
<a href="#">pauseAudio</a>	暂停音频。
<a href="#">resumeAudio</a>	恢复音频。
<a href="#">setPlayoutVolume</a>	设置播放音量。
<a href="#">getPlayoutVolume</a>	获取音频音量。
<a href="#">muteAudio</a>	静音。
<a href="#">streamStatistic</a>	设置是否开启流信息统计。

接口	描述
<a href="#">enableStreamStateDetection</a>	开启/关闭音视频码流状态探测功能。
<a href="#">destroyClient</a>	销毁客户端对象。
<a href="#">fullScreenToggle</a>	开启关闭全屏。

## startPlay

startPlay(url: string, options: StartPlayOptions): Promise<void>

### 【功能说明】

开始播放，客户端根据输入的URL到服务端拉取对应的主播流。

### 【请求参数】

- url: 必选，string类型。拉流URL，是以m3u8结尾的播放地址
- options: 可选，StartPlayOptions类型。播放配置参数，如果不携带该参数，则复用首次起播携带的options数据。StartPlayOptions定义如下：{
  - elementId: 必选，播放DOM标识ID。
  - objectFit: 可选，string类型，默认值为cover。支持的枚举值如下：
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放，直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致，在保持长宽比的前提下，将视频进行缩放后填满视窗，缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放，直至整个视窗被视频填满。如果视频长宽与显示窗口不同，则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配，那么视频将被拉伸以适应视窗。
  - muted: 可选，boolean类型，true表示静音，false表示不静音。默认值为false。
  - sessionId: 不需要传。
  - showLoading: 可选，boolean类型，true表示开启loading的展示效果，默认为false。当该参数设置为true时，起播loading效果同步开启，播放过程中发生缓冲时loading的效果，需根据[setParameter](#)接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选，boolean类型，true表示开启自动起播功能，false表示非自动起播，需要人为触发播放，默认为true。
  - poster: 可选，对象定义如下：{
    - url: 可选，string类型。设置播放封面图片完整地址，图片格式限JPGPNG和静态GIF格式，大小不超过1MB，尺寸不超过1920 x 1080，文件名不得含有中文字符。
    - mode: 可选，string类型。默认值为cover。支持的枚举值如下：{

- fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不匹配, 那么视频将被拉伸以适应视窗。
  - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。
- ```
}  
  
▪ startEnable: 可选, boolean类型。启动播放时是否展示播放封面, true表示展示, false表示不展示播放封面, 默认值false。该参数只在设置非自动播放场景下生效。  
  
▪ pauseEnable: 可选, boolean类型。触发暂停操作时, 是否在播放页面展示播放封面, true表示展示播放封面, false表示不展示, 默认值false。  
  
}  
  
- webrtcConfig: 不需要传。  
- schedulePolicy: 不需要传。  
- domainPolicy: 不需要传。  
- downgradeUrl: 不需要传。
```

#### 【返回参数】

Promise<void>: 返回一个Promise对象。

## switchPlay

switchPlay(url: string, options: StartPlayOptions): Promise<void>

#### 【功能说明】

起播成功后, 快速切换下一路流播放。

#### 【请求参数】

- url: 必选, string类型。拉流URL, 是以m3u8结尾的播放地址
- options: 可选, StartPlayOptions类型。播放配置参数, 如果不携带该参数, 则复用首次起播携带的options数据。StartPlayOptions定义如下: {
  - elementId: 必选, 播放DOM标识ID。
  - objectFit: 可选, string类型, 默认值为cover。支持的枚举值如下:
    - contain: 优先保证视频内容全部显示。视频尺寸等比缩放, 直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致, 在保持长宽比的前提下, 将视频进行缩放后填满视窗, 缩放后的视频四周会有一圈黑边。
    - cover: 优先保证视窗被填满。视频尺寸等比缩放, 直至整个视窗被视频填满。如果视频长宽与显示窗口不同, 则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill: 视频内容完全填充视窗。如果视频的宽高比与视窗不匹配, 那么视频将被拉伸以适应视窗。
  - muted: 可选, boolean类型, true表示静音, false表示不静音。默认值为false。

- sessionId: 不需要传。
  - showLoading: 可选, boolean类型, true表示开启loading的展示效果, 默认为false。当该参数设置为true时, 起播loading效果同步开启, 播放过程中发生缓冲时loading的效果, 需根据setParameter接口中的LOADING\_CONFIG进行设置。
  - autoPlay: 可选, boolean类型, true表示开启自动起播功能, false表示非自动起播, 需要人为触发播放, 默认为true。
  - poster: 可选, 对象定义如下: {
    - url: 可选, string类型。设置播放封面图片完整地址, 图片格式限JPGPNG和静态GIF格式, 大小不超过1MB, 尺寸不超过1920 x 1080, 文件名不得含有中文字符。
    - mode: 可选, string类型。默认值为cover。支持的枚举值如下: {
      - fill: 视频内容完全填充视窗, 如果视频的宽高比与视窗不相匹配, 那么视频将被拉伸以适应视窗。
      - crop: 播放封面原始尺寸大小展示, 如果海报超出播放区域, 则会对超出部分进行裁剪, 否则在播放窗口居中展示。}
  - startEnable: 可选, boolean类型。启动播放时是否展示播放封面, true表示展示, false表示不展示播放封面, 默认值false。该参数只在设置非自动播放场景下生效。
  - pauseEnable: 可选, boolean类型。触发暂停操作时, 是否在播放页面展示播放封面, true表示展示播放封面, false表示不展示, 默认值false。
- webrtcConfig: 不需要传。
- schedulePolicy: 不需要传。
- domainPolicy: 不需要传。
- downgradeUrl: 不需要传。

#### 【返回参数】

Promise<void>: 返回一个Promise对象。

## stopPlay

stopPlay(): boolean

#### 【功能说明】

停止播放。

#### 【请求参数】

无

#### 【返回参数】

boolean: 停止播放结果。true表示成功, false表示失败。

## replay

replay(): Promise<boolean>

### 【功能说明】

重新播放。

### 【请求参数】

无

### 【返回参数】

Promise<boolean>: 重新播放的结果。true表示成功, false表示失败。

## resume

resume(): Promise<boolean>

### 【功能说明】

恢复播放。

### 【请求参数】

无

### 【返回参数】

Promise<boolean>: 恢复音视频播放的结果。true表示成功, false表示失败。

## pause

pause(): boolean

### 【功能说明】

暂停音视频播放。

### 【请求参数】

无

### 【返回参数】

boolean: 暂停播放的结果。true表示成功, false表示失败。

## pauseVideo

pauseVideo(): boolean

### 【功能说明】

该接口不支持。

### 【请求参数】

无

### 【返回参数】

boolean: 只返回false。



## resumeVideo

resumeVideo(): Promise<boolean>

### 【功能说明】

该接口不支持。

### 【请求参数】

无

### 【返回参数】

boolean: 只返回false。

## pauseAudio

pauseAudio(): boolean

### 【功能说明】

暂停音频。

### 【请求参数】

无

### 【返回参数】

boolean: 暂停音频播放结果。true表示成功，false表示失败。

## resumeAudio

resumeAudio(): Promise<boolean>

### 【功能说明】

恢复音频。

### 【请求参数】

无

### 【返回参数】

Promise<boolean>: 恢复音频播放的结果。true表示成功，false表示失败。

## setPlayoutVolume

setPlayoutVolume(volume: number): boolean

### 【功能说明】

设置音频音量，会开启声音。

### 【请求参数】

volume: 必选，number类型，音频的音量值。取值范围为[0,100]。

### 【返回参数】

boolean: 设置音频音量是否成功。true表示成功，false表示失败。

## getPlayoutVolume

getPlayoutVolume(): number

### 【功能说明】

获取音频音量。

### 【请求参数】

无

### 【返回参数】

number: 音量值, 取值范围为[0,100]。

## muteAudio

muteAudio(isMute: boolean): void

### 【功能说明】

静音。

### 【请求参数】

isMute: 必选, boolean类型, 是否静音。true表示静音, false表示取消静音。

### 【返回参数】

无

## streamStatistic

streamStatistic(enable: boolean, interval: number): void

### 【功能说明】

设置是否开启流信息统计。

### 【请求参数】

- enable: 必选, boolean类型, 是否开启流信息统计, true表示开启统计。
- interval: 必选, number类型, 设置统计间隔, 单位为秒, 取值范围为[1, 60], 默认值为1。

### 【返回参数】

无

## enableStreamStateDetection

enableStreamStateDetection(enable: boolean, interval: number, interruptRetry:StreamInterruptRetry): boolean

### 【功能说明】

开启/关闭音、视频码流状态探测功能, 开启后可探测推流侧是否处于断流的状态。

### 【请求参数】

- enable: 必选, boolean类型, true表示开启音视频码流状态探测, false表示关闭音视频码流状态探测。默认值为false。

- interval: 必选, number类型, 单位为秒, 取值范围为[1,60]。音视频无码流状态的判断时间。默认值为3, 推荐设置为3。
- interruptRetry: 可选, StreamInterruptRetry类型。断流重试播放配置参数, StreamInterruptRetry定义为: {  
enable: boolean类型, 开启断流后尝试自动恢复播放。默认值为false, 即不开启自动重试。  
retryInterval: number类型, 拉流播放的重试周期, 单位为秒。最小值10, 最大值建议不超过60, 默认值为30。  
retryTimes: number类型, 尝试重新恢复播放的最大重试次数。最小值1, 默认值为30。  
}

**【返回参数】**

boolean: 是否成功, true表示成功, false表示失败。

## destoryClient

destoryClient(): void

**【功能说明】**

销毁客户端对象。

**【请求参数】**

无

**【返回参数】**

无

## fullScreenToggle

fullScreenToggle(isExit: boolean): void

**【功能说明】**

该接口不支持。

**【请求参数】**

isExit: 必选, 布尔类型, 默认为false。

**【返回参数】**

无

## 3.5.5 客户端事件通知 ( HWLLSClientEvent )

本章节介绍了低时延直播Web SDK的HWLLSClientEvent接口详情。

表 3-7 HWLLSClientEvent 接口

| 接口                              | 描述      |
|---------------------------------|---------|
| <a href="#">media-statistic</a> | 媒体统计事件。 |

| 接口                                        | 描述         |
|-------------------------------------------|------------|
| <a href="#">network-quality</a>           | 网络质量报告事件。  |
| <a href="#">video-broken</a>              | 视频断流事件。    |
| <a href="#">audio-broken</a>              | 音频断流事件。    |
| <a href="#">video-recovery</a>            | 视频断流恢复事件。  |
| <a href="#">audio-recovery</a>            | 音频断流恢复事件。  |
| <a href="#">audio-start</a>               | 音频起播事件。    |
| <a href="#">video-start</a>               | 视频起播事件。    |
| <a href="#">video-stuck</a>               | 视频是否停顿事件   |
| <a href="#">fullscreen-status-changed</a> | 播放视图是否全屏事件 |
| <a href="#">player-changed</a>            | 播放降级事件     |
| <a href="#">Error</a>                     | 客户端出现错误事件。 |

 **注意**

事件注册监听应在业务结束时取消注册，否则注册监听事件累积会有内存泄漏风险。

## media-statistic

### 【事件说明】

媒体统计事件。此事件配合[streamStatistic](#)方法使用。

### 【回调参数】

StatisticInfo: StatisticInfo类型，媒体统计信息。

StatisticInfo定义为：{

- video: {
  - mediaType: MediaType媒体类型。
  - frameRate: number类型，视频帧率。
  - width: number类型，视频宽度。
  - height: number类型，视频高度。
  - jitter: number类型，抖动值。
  - bitRate: number类型，码率，单位：kbps。
  - bytesReceived: number类型，已接收字节数。
  - packetsReceived: number类型，已接收包数。
  - packetsLost: number类型，丢包数。

- audio: {  
    mediaType: MediaType媒体类型。  
    jitter: number类型，抖动值。  
    bitRate: number类型，码率，单位：kbps。  
    bytesReceived: number类型，已接收字节数。  
    packetsReceived: number类型，已接收包数。  
    packetsLost: number类型，丢包数。  
  }

## network-quality

### 【事件说明】

网络质量报告事件。

### 【回调参数】

NetworkQualityTypes: NetworkQualityTypes类型，网络质量详情。

NetworkQualityTypes枚举值参考如下：

- NETWORK\_QUALITY\_UNKNOWN = 0，网络质量未知。
- NETWORK\_QUALITY\_GREAT = 1，网络质量极好。
- NETWORK\_QUALITY\_GOOD = 2，用户主观感觉和极好差不多，但码率可能略低于极好。
- NETWORK\_QUALITY\_DEFECTS = 3，网络质量一般，用户主观感受有瑕疵但不影响观看。
- NETWORK\_QUALITY\_WEAK = 4，网络质量差，勉强能观看但不流畅。
- NETWORK\_QUALITY\_BAD = 5，网络质量很差，严重影响用户观看体验。
- NETWORK\_QUALITY\_DISCONNECT = 6，网络质量非常差甚至链接断开，无法观看。

## video-broken

### 【事件说明】

视频断流事件。

### 【回调参数】

无

## audio-broken

### 【事件说明】

音频断流事件。

### 【回调参数】

无

## video-recovery

### 【事件说明】

视频断流（非EOF）恢复事件。

### 【回调参数】

无

## audio-recovery

### 【事件说明】

音频断流（非EOF）恢复事件。

### 【回调参数】

无

## audio-start

### 【事件说明】

音频起播事件。

### 【回调参数】

无

## video-start

### 【事件说明】

视频起播事件。

### 【回调参数】

无

## video-stuck

### 【事件说明】

视频停顿事件。

### 【回调参数】

布尔值，True标识停顿，False标识非停顿。

## fullscreen-status-changed

### 【事件说明】

播放视图是否全屏事件

### 【回调参数】

- isFullScreen: 是否全屏展示。

- isPause: 是否停止播放。

## player-changed

### 【事件说明】

播放降级事件。

### 【回调参数】

string类型，降级信息。

- webrtc: 标识LLL播放模式。
- hls: 标识hls播放模式。
- flv: 标识flv播放模式。

## Error

### 【事件说明】

客户端错误事件，当出现不可恢复的错误后，Client会上报该事件通知。

### 【回调参数】

errorInfo: 必选，ErrorInfo类型，错误信息。

ErrorInfo定义为: {

errorCode: 必选，string类型，错误码。

errorMsg: 必选，string类型，错误描述。

}

### 注意

在网络防火墙限制（UDP端口限制）或者低时延直播多次重试播放失败的场景下，可根据特定错误码（HWLLS\_MEDIA\_NETWORK\_ERROR、HWLLS\_PLAY\_WEBRTC\_RETRY\_FAILED），进行播放降级，请参考[SDK使用](#)。

## 3.5.6 错误码（ErrorCode）

### getCode

getCode(): number

### 【功能说明】

获取错误码。

### 【请求参数】

无

### 【返回参数】

number类型，错误码值。

## getMsg

getMsg(): string

### 【功能说明】

获取错误描述。

### 【请求参数】

无

### 【返回参数】

string类型，错误码描述。

## 3.5.7 公网地址

表 3-8 公网地址列表

| 公网地址                                         | 信息           |
|----------------------------------------------|--------------|
| log-collection-new.hwcloudlive.com           | 国内日志和打点环境地址。 |
| log-collection-ap-southeast-3.rocket-cdn.com | 海外日志和打点环境地址。 |
| global-lll.huaweicloud.com                   | 默认拉流主环境地址。   |
| global-lll.huaweicloud.cn                    | 默认拉流备用环境地址。  |
| hcdnl-pull302-global-gslb.livehwc3.cn        | 默认GSLB环境地址。  |

## 3.5.8 客户端错误码

本章节介绍了低时延直播Web SDK的客户端错误码的详细信息。

表 3-9 错误码说明

| 类成员                             | 错误码    | 描述       | 错误原因或建议处理方式                   |
|---------------------------------|--------|----------|-------------------------------|
| HWLLS_OK                        | 0      | 成功。      | -                             |
| HWLLS_ERROR_INVALID_URL         | 500000 | URL不合法。  | 检查URL是否正确。                    |
| HWLLS_ERROR_INVALID_PARAMETER   | 500001 | 参数不合法。   | 参数传递错误，请检查接口入参是否符合接口的参数合法性要求。 |
| HWLLS_ERROR_SERVER_CONNECT_FAIL | 500002 | 无法连接服务端。 | 检查网络状态是否正常或联系华为工程师。           |



| 类成员                                  | 错误码     | 描述                  | 错误原因或建议处理方式                                                |
|--------------------------------------|---------|---------------------|------------------------------------------------------------|
| HWLLS_ERROR_SERVER_NO_RESPONSE       | 5000003 | 服务器无响应。             | 联系华为工程师。                                                   |
| HWLLS_ERROR_AUTH_FAIL                | 5000004 | 鉴权失败。               | 请检查服务侧Referer防盗链以及key防盗链的配置是否正确。                           |
| HWLLS_ERROR_STREAM_NOT_EXIST         | 5000005 | 请求的流不存在。            | 使用存在的流。                                                    |
| HWLLS_ERROR_WEBRTC_UNSUPPORTED       | 5000006 | 浏览器不支持。             | 1、参见 <a href="#">浏览器适配</a> ，使用可支持的浏览器。<br>2、建议降级FLV或HLS直播。 |
| HWLLS_MEDIA_NETWORK_ERROR            | 5000007 | 媒体网络连接异常。           | 检查网络状态、防火墙配置是否正确，或者建议降级FLV或HLS直播。                          |
| HWLLS_ERROR_STREAM_INVALID_PARAMETER | 5000008 | 流信息错误。              | 检查请求的流的URL是否正确。                                            |
| HWLLS_ERROR_BANDWIDTH_OVER_MAXIMUM   | 5000009 | 带宽超上限               | 联系华为工程师。                                                   |
| HWLLS_INTERNAL_ERROR                 | 5000020 | 其他内部错误。             | 联系华为工程师。                                                   |
| HWLLS_BUSINESS_DOWNGRADE             | 5000021 | 业务需要降级。             | 建议降级FLV或HLS直播。                                             |
| HWLLS_PLAY_WEBRTC_RETRY_FAILED       | 5000022 | 低时延直播播放中断后，多次尝试重播失败 | 建议降级FLV或HLS直播。                                             |
| HWLLS_PLAY_FLV_RETRY_FAILED          | 5000023 | flv直播播放中断后，多次尝试重播失败 | 联系华为工程师。                                                   |
| HWLLS_PLAY_HLS_RETRY_FAILED          | 5000024 | hls直播播放中断后，多次尝试重播失败 | 联系华为工程师。                                                   |
| HWLLS_ERROR_LIVE_UNSUPPORTED         | 5000030 | 播放内容格式，浏览器不支持       | 联系华为工程师。                                                   |
| HWLLS_ERROR_UNEXPECTED_EOF           | 5000031 | 播放内容，异常的网络EOF       | 建议重新尝试。                                                    |
| HWLLS_ERROR_MEDIA_ERROR              | 5000032 | 播放媒体内容异常            | 建议重新尝试或联系华为工程师。                                            |

| 类成员                            | 错误码     | 描述                 | 错误原因或建议处理方式                                                                                                                                                                                |
|--------------------------------|---------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HWLLS_ERROR_REPORT_TOKEN_ERROR | 5000033 | 打点和日志上传Token异常     | 联系华为工程师。                                                                                                                                                                                   |
| HWLLS_PLAY_NOT_ALLOWED         | 5100000 | 播放权限受限，需要手动触发播放。   | 由于浏览器自动播放安全策略的限制，浏览器直接拉起App并启动播放会返回该错误，在应用层需要根据该错误码，引导用户通过手动触发页面UI控件，调用 <code>replay</code> 接口恢复播放。<br><b>须知</b><br>Safari浏览器还有以下处理方法：<br>打开Safari浏览器偏好设置 > 网站 > 自动播放 > 选择相应网站，设置允许全部自动播放。 |
| HWLLS_PLAY_TIMEOUT             | 5100001 | 播放超时，10s内没有拉到有效帧数据 | 需确认推流情况或联系华为工程师。                                                                                                                                                                           |

## 3.6 常见问题

- 如果业务上App只能使用http协议，是否能够集成使用华为低时延直播Web SDK？**

部分浏览器（chrome）可以集成使用，但不推荐。由于浏览器兼容性识别是根据浏览器暴露的WebRTC对象判断的，在非https协议下，对象可能不存在。
- Firefox浏览器中无法使用华为低时延直播Web SDK？**

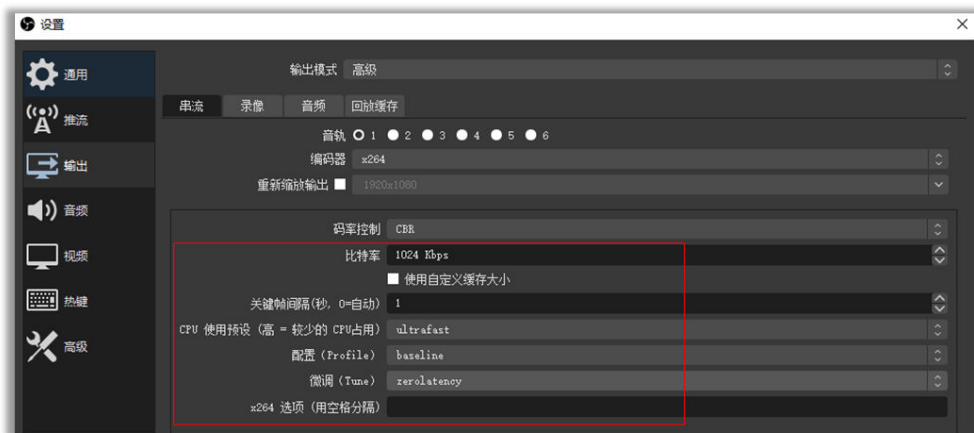
Firefox浏览器使用之前需要安装H264的编解码插件。浏览器中输入`about:addons`，跳转到插件安装页面，查看H264插件是否安装完成，如未安装请在该页面更新安装。
- 集成华为低时延直播Web SDK后，无法正常使用，可能原因？**

  - 需要检查用户自定义的域名配置是否完成，如：推、拉流域名，权威机构签发的https证书等。
  - 推流端设置及推流是否正常。
  - 播放地址是否填写正确，如：appName、streamName等。
  - 网络连接是否正常、网络防火墙配置是否有限制，如：UDP端口（8000-8063）是否放通。
- 华为低时延直播Web SDK，支持哪些类型浏览器？**

浏览器支持详情请参见[浏览器适配](#)。
- 推流端推流成功后，华为低时延直播Web SDK拉流播放失败？**

需要确认推流端的推流编码参数，是否为H264+无B帧。目前华为低时延直播Web SDK仅支持H264+无B帧的流，所以如果原始流为H265或者带B帧，则需要提前在租户Console上配置对应转码模板，开启转码服务，但这样会引入额外的转码延

迟，并且会产生转码费用。建议推流端尽量推H264+不包含B帧的流，可以通过调整推流端软件（如OBS）的视频编码参数去除B帧。如果使用OBS推流，可以通过设置，关闭B帧。如下图所示：



- **华为低时延直播Web SDK，播放报错：NotAllowedError:xxx?**  
由于浏览器自动播放安全策略的限制，浏览器直接拉起App并启动播放会返回该错误，在应用层需要根据该错误码，引导用户通过手动触发页面UI控件，并调用 [replay](#) 接口恢复播放。
- **开启认证策略，该如何获取token信息？**  
认证策略不开启不影响功能正常使用，也不影响打点和日志上传的能力。  
认证策略开启可以保证打点数据和日志上传数据的安全性。  
如果当前需要认证策略能力，请[提交工单](#)，联系技术支持获取appid和token。

## 3.7 修订记录

表 3-10 修订记录

| 修改时间       | 修改说明                                         |
|------------|----------------------------------------------|
| 2024-06-27 | 新增 <a href="#">setReportConfig</a> 接口，更新SDK。 |
| 2024-03-19 | 1、新增SDK包下载路径及其完整性校验方法。<br>2、新增FLV、HLS等相关资料。  |
| 2023-10-30 | 第一次正式发布。                                     |

# 4 修订记录

| 发布日期       | 修订记录             |
|------------|------------------|
| 2024-06-27 | SDK软件版本更新至2.6.9。 |
| 2024-06-12 | SDK软件版本更新至2.6.3。 |
| 2023-10-30 | 第一次正式发布。         |