

ABLMediaServer 使用说明

目录

ABLMediaServer 使用说明	1
1. 功能说明	8
2. ABLMediaServer 主要功能	10
1. 网络协议媒体输入	10
2. 网络协议媒体输出	14
3. 配置文件说明	15
4. 对外接口说明	21
4.1. 开启摄像头录像	21
4.1.1. 基本信息	22
4.1.2. 请求参数	22
4.1.3. 返回参数	23
4.2. 关闭摄像头录像	24
4.2.1. 基本信息	24
4.2.2. 请求参数	24
4.2.3. 返回参数	26
4.3. 查询录像列表	26
4.3.1. 基本信息	26
4.3.2. 请求参数	27

4.3.3. 返回参数	28
4.4. 录像回放控制	31
4.4.1. 基本信息	32
4.4.2. 请求参数	32
4.4.3. 返回参数	34
4.5. 添加代理拉流	35
4.5.1. 基本信息	35
4.5.2. 请求参数	35
4.5.3. 返回参数	39
4.6. 删除代理拉流	40
4.6.1. 基本信息	40
4.6.2. 请求参数	41
4.6.3. 返回参数	42
4.7. 添加 rtsp、rtmp 推流	42
4.7.1. 基本信息	42
4.7.2. 请求参数	43
4.7.3. 返回参数	45
4.8. 删除代理推流	45
4.8.1. 基本信息	46
4.8.2. 请求参数	46
4.8.3. 返回参数	47

4.9. 添加 ffmpeg 代理拉流	47
4.9.1. 基本信息	48
4.9.2. 请求参数	48
4.9.3. 返回参数	50
4.10. 删除 ffmpeg 函数代理拉流	51
4.10.1. 基本信息	51
4.10.2. 请求参数	51
4.10.3. 返回参数	52
4.11. 创建 GB28181\Utt1078 接收端口	53
4.11.1. 基本信息	53
4.11.2. 请求参数	53
4.11.3. 返回参数	56
4.12. 暂停 GB28181\Utt1078 接收端口	57
4.12.1. 基本信息	57
4.12.2. 请求参数	57
4.12.3. 返回参数	58
4.13. 继续 GB28181\Utt1078 接收端口	59
4.13.1. 基本信息	59
4.13.2. 请求参数	59
4.13.3. 返回参数	60
4.14. 删除 GB28181 接收端口\Utt1078 接收端口	61

4.14.1. 基本信息	61
4.14.2. 请求参数	61
4.14.3. 返回参数	62
4.15. 创建 GB28181 发送端口\Jtt1078 发送端口	63
4.15.1. 基本信息	63
4.15.2. 请求参数	63
4.15.3. 返回参数	66
4.16. 删除 GB28181 发送端口\Jtt1078 发送端口	67
4.16.1. 基本信息	67
4.16.2. 请求参数	67
4.16.3. 返回参数	68
4.17. 获取流媒体服务器所有可用的媒体源	69
4.17.1. 基本信息	69
4.17.2. 请求参数	69
4.17.3. 返回参数	70
4.18. 删除 某一个媒体源	73
4.18.1. 基本信息	73
4.18.2. 请求参数	73
4.18.3. 返回参数	74
4.19. 获取流媒体服务器所有输出流列表	75
4.19.1. 基本信息	75

4.19.2. 请求参数	75
4.19.3. 返回参数	76
4.20. 删除 某一个服务器所有往外部输出码流列表	79
4.20.1. 基本信息	79
4.20.2. 请求参数	79
4.20.3. 返回参数	80
4.21. 删除任意一个或一组或者全部媒体输入列表	81
4.21.1. 请求参数	81
4.21.2. 返回参数	83
4.22. 获取系统配置参数	83
4.22.1. 基本信息	83
4.22.2. 请求参数	84
4.22.3. 返回参数	85
4.23. 图片抓拍	89
4.23.1. 基本信息	89
4.23.2. 请求参数	90
4.23.3. 返回参数	91
4.24. 图片列表查询	92
4.24.1. 基本信息	92
4.24.2. 请求参数	92
4.24.3. 返回参数	93

4.25. 获取当前转码的数量	94
4.25.1. 基本信息	94
4.25.2. 请求参数	95
4.25.3. 返回参数	95
4.26. 列举流媒体服务器所有占用端口	96
4.26.1. 基本信息	96
4.26.2. 请求参数	96
4.26.3. 返回参数	98
4.27. 控制代理拉流	99
4.27.1. 基本信息	99
4.27.2. 请求参数	99
4.27.3. 返回参数	101
4.28. 修改某一路的水印相关参数	102
4.28.1. 基本信息	102
4.28.2. 请求参数	102
4.28.3. 返回参数	104
4.29. 为调试视频需要增加了一个接口来打开代理拉流	104
4.29.1. 基本信息	105
4.29.2. 请求参数	105
4.29.3. 返回参数	107
4.30. 安全停止服务器	107

4.30.1. 基本信息	107
4.30.2. 请求参数	108
4.30.3. 返回参数	108
4.31. 重新启动服务器	109
4.31.1. 基本信息	109
4.31.2. 请求参数	109
4.31.3. 返回参数	110
4.32. 消息通知（主动推送）	111
4.32.1. 功能说明	111
4.32.2. 请求示例	112
4.33. 批量设置服务器的配置参数	119
4.33.1. 请求参数	120
5. 网络类型的对照表	121
1. 媒体输入网络类型对照表	121
2. 媒体输出网络类型对照表	121
6. 国标单端口接入	122
6.1. 服务器支持国标单端口接入特殊说明	122
7. 版本记录	123
7.1. 2022-01-25 发布	123
7.2. 2022-03-28 发布	123
7.3. 2022-04-11 发布	124
7.4. 2022-05-06 发布	124
7.5. 2022-06-30 发布	126
7.6. 2022-08-04 发布	128
7.7. 2022-10-28 发布	128

7.8. 2022-11-30 发布 129

7.9. 2023-03-29 发布 129

7.10. 2023-05-30 发布 129

7.11. 2023-07-28 发布 130

7.12. 2024-01-05 发布 130

7.13. 2024-05-06 发布 131

<div>文件状态：</div> <div><div><input type="checkbox"/> 草稿</div><div><input checked="" type="checkbox"/> 正式发布</div><div><input type="checkbox"/> 正在修改</div></div>	文档编码：	ABLMediaServer
	当前版本：	V1.0.0.0
	作 者：	罗家兄弟
	文 档：	逍遥子
	完成日期：	2024-05-26

1. 功能说明

ABLMediaServer 高性能流媒体服务器，windows 平台采用高性能的完成端口网络模型、Linux 下采用 epoll，加上线程池进行媒体数据的接收、转换、发送。服务器的性能强劲、运行稳定，坚如磐石。经过压力测试比较，转发性能、CPU 占有率、运行稳定性优势明显。

可以通过 ffmpeg 命令或者其他标准的 rtsp、rtmp 推流函数往 ABLMediaServer 推送 rtsp 流、rtmp 流，请求服务器代理拉流，请求服务器接收国标 GB28181。服务器通过转换，输

出标准的 rtsp 码流、rtmp 码流、http-flv、ws-flv 码流 (支持 H265 视频输出)、http-mp4(视频支持 H264、H265, 音频支持 AAC)、hls 码流输出 (视频支持 H264、H265, 音频支持 AAC)、GB28181 码流(国标 PS 流)输出。流媒体服务器支持录像、支持智能录像删除、录像查询、录像文件点播、录像文件极速下载。http-flv、ws-flv、http-mp4 协议点播时支持暂停继续、支持拖动播放。rtsp 点播录像文件时支持慢放 (1/16、1/8、1/4、1/2)、快放 (2、4、8、16)、正常速度、支持拖动播放。服务器支持 H265 转码为 H264, 转码输出视频支持指定分辨率、宽高、码率大小等等参数。Windows 平台支持英伟达显卡硬件加速转码。实测 linux 平台能最大并发转码 40 路 H265 (硬件环境 至强 E5 2650 V3), 实测 windows 平台能最大并发转码 35 路 H265 (i9 + 英伟达 RTX 2080)。

Linux 新增支持英伟达专业显卡,支持硬件解码、硬件编码, 支持多个显卡同时进行转码。两张特斯拉 T4 (16G) 显卡可以同时把 80 路 1080P 的 H265 视频转码为 1280 x 720 的 H264, 转码延时低于 10 毫秒, 如果需要更强的转码功能可以多插几张显卡或者集群。

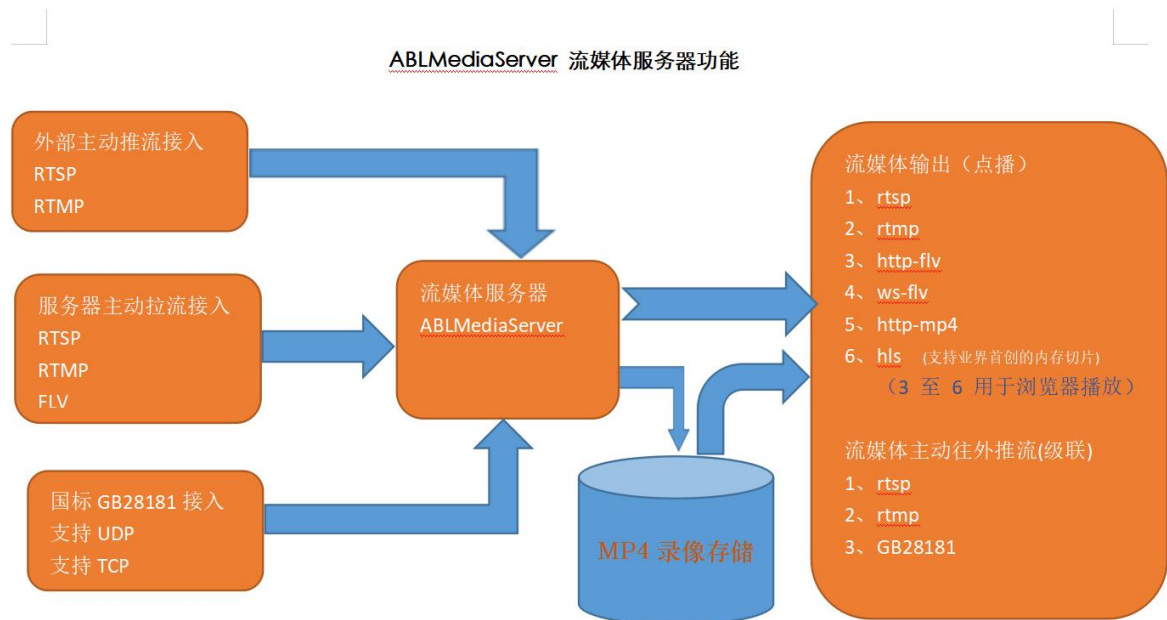
流媒体服务器支秒级(基本上 1 秒以内)图片抓拍, 支持高速并发抓拍最大并发每秒 66 张, 支持对抓拍的图片进行查找, 支持抓拍图片以 http 协议下载, 支持抓拍对象复用功能, 最快 31 毫秒抓拍出一张图片, 适合一些每隔 1~3 秒抓拍 1 次的高速抓拍场合。

非常感谢 ZLMediaKit 和 Media-Server 两个工程项目的作者, ABLMediaSever 参考复用了 ZLMediaKit 的 http 请求函数的参数表(请求的 URL 及请求的参数名及一些初始值)以及整个框架的一些设计思想。直接调用了 Media-Server 工程的 rtmp、flv、hls、fmp4 复用与解复用等等。

非常崇拜与感谢两位大佬为中国的开源流媒体努力与付出, 再次感谢。

【欢迎加入高性能流媒体服务 QQ 群 [873666268](#)】

2. ABLMediaServer 主要功能



1. 网络协议媒体输入

rtsp、rtmp 外部主动推流输入

- 1、rtsp 外部主动推流 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 2、rtmp 外部主动推流 (支持 视频: H264、H265 , 音频: AAC)
- 3、国标 GB28181 输入 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 4、交通运输部jtt1078 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)

rtsp、rtmp、http-flv 拉流输入:

- 1、rtsp 拉流(支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 2、rtmp 拉流(支持 视频: H264、H265 , 音频: AAC)
- 3、http-flv 拉流 (支持 视频: H264、H265 , 音频: AAC)
- 4、本地 mp4 拉流 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)

5、websocket 方式私有协议发送 PCM 数据进入流媒体服务器(可以在浏览器录音通过 websocket 以下面的私有协议把语音接入流媒体服务器方便国标对讲、或者其他应用比如语音广播)

1、数据包格式

包头 + 数据类型 + 长度 + 数据包

包头 4 个字节 0xAB 0xCD 0xEF 0xAB

数据类型 1 个字节 , [0x01 推流音频注册] [0x02 发送 PCM 音频数据包] [0x03 断开音频推流]

长度 2 个字节 , 网络字节顺序 (大端)

数据包 音频注册包、pcm 音频数据包、注销包

2、数据包类型

1)、音频注册包 json 格式

```
{"method":"register","app":"talk","stream":"xian_1","audioCodec":"pcm","channels":1,"sampleRate":16000,"targetAudioCodec":"g711a"}
```

样例：发送音频注册

websocket 只要连接上来，就首先发送 音频注册包

(0xAB 0xCD 0xEF 0xAB) + 0x01 + strlen(音 频 注 册 包) +

```
{"method":"register","app":"talk","stream":"xian_1","audioCodec":"pcm","channels":1,"sampleRate":16000,"targetAudioCodec":"g711a"}
```

2)、音频数据包 只支持 16 位采样精度的 pcm , 单通道、双通道、常见的采样频率都支持 (8000、16000、32000、48000)

样例: 现在只发送纯 pcm 音频数据

(0xAB 0xCD 0xEF 0xAB) + 0x02 + strlen(pcm 纯音频数据) + pcm 纯音频数据

3)、注销包

```
{"method":"destruction"}
```

样例: 发送注销包

(0xAB 0xCD 0xEF 0xAB) + 0x03 + strlen({"method":"destruction"}) + {"method":"destruction"}

支持用 ffmpeg 推纯音频流接入流媒体服务器, 支持的音频格式有(g711A、G711U、AAC、MP3)

1、以 rtsp 协议推本地 aac 文件接入流媒体

```
ffmpeg -re -stream_loop -1 -i F:\music\xxx.aac -c:a aac  
-rtsp_transport tcp -f rtsp rtsp://190.15.240.11:554/music/aac1
```

2、以 rtmp 协议推本地 aac 文件接入流媒体

```
ffmpeg -re -stream_loop -1 -i F:\music\xxx.aac -c:a aac -f flv
```

rtmp://190.15.240.11:1935/music/aac2

3、以 rtsp 协议推本地 mp3 文件接入流媒体

```
ffmpeg -re -stream_loop -1 -i F:\music\xxx.mp3 -c:a mp3  
-rtsp_transport tcp -f rtsp rtsp://190.15.240.11:554/music/aac3
```

4、以 rtmp 协议推本地 mp3 文件接入流媒体

```
ffmpeg -re -stream_loop -1 -i F:\music\xxx.mp3 -c:a mp3 -f flv  
rtmp://190.15.240.11:1935/music/aac4
```

5、以 rtsp 协议推本地 mp4 文件中的纯音频流接入服务器

1) 比如推一个视频格式为 h264, 音频为 aac 的 mp4 文件中的音频流进入服务器, 那么服务器接收到的音频流为 aac ,视频不做推送

```
ffmpeg -re -stream_loop -1 -i  
F:\video\H264_AAC_2021-02-10_1080P.mp4 -vn -acodec copy -rtsp_transport tcp -f rtsp  
rtsp://44.35.33.39:554/music/aac5
```

2) 比如推一个视频格式为 h264, 音频为 g711A 的 mp4 文件中的音频流进入服务器, 那么服务器接收到的音频流为 g711A ,视频不做推送

```
ffmpeg -re -stream_loop -1 -i F:\video\H264-G711A-92017.mp4  
-vn -acodec copy -rtsp_transport tcp -f rtsp rtsp://44.35.33.39:554/music/aac6
```

3) 比如推一个视频格式为 h264, 音频为 g711U 的 mp4 文件中的音频流进入服务器, 那么服务器接收到的音频流为 g711U ,视频不做推送

```
ffmpeg -re -stream_loop -1 -i F:\video\H264-G711U-56434.mp4  
-vn -acodec copy -rtsp_transport tcp -f rtsp rtsp://44.35.33.39:554/music/aac7
```

2. 网络协议媒体输出

被动拉流输出

- 1、rtsp (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 2、rtmp (支持 视频: H264、H265 , 音频: AAC)
- 3、GB28181 码流 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 4、http-flv (支持 视频: H264、H265 , 音频: AAC)
- 5、http-hls (支持 视频: H264、H265 , 音频: AAC)
- 6、http-mp4 (支持 视频: H264、H265 , 音频: AAC)
- 7、websocket-flv (支持 视频: H264、H265 , 音频: AAC)
- 8、webrtc (支持 视频: H264、H265 , 音频: G711A、G711U、
OPUS)
- 9、交通运输部 jtt1078 (支持 视频: H264、H265 , 音频: AAC、G711A、
G711U)

rtsp、rtmp、gb28181 主动推流输出:

- 1、rtsp 推流 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 2、rtmp 推流 (支持 视频: H264、H265 , 音频: AAC)
- 3、GB28181 推流 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)
- 4、交通运输部 jtt1078 (支持 视频: H264、H265 , 音频: AAC、G711A、G711U)

3. 配置文件说明

配置文件名称: ABLMediaServer.ini

名称	说明	是否必填
MediaServerID	流媒体服务器 ID, 用于集群识别流媒体服务器	是
secret	http 请求接口密钥 对外接口 一些比较敏感的 http api 在访问时需提供 secret, 否则提示无权限调用	是
localipAddress	配置本机的 IP 地址, 如果不配置, 程序会自动获取本机所有 IP 默认取一个地址, 如果本机有多个地址可能会不准确, 因为调用 getMediaList 获取可用的媒体源的 json 中, 会使用到本机的 IP 地址来拼接 rtsp、rtmp、http-flv、ws-flv、hls、http-mp4 的播放 url	否
maxTimeNoOneWatch	所有接入流, 当多长时间持续无人观看时会自动断开码流接入, 包括推流接入, 拉流接入, 国标接入, 时间单位 分钟 【注意: 当该路接入流启动了录像, 不管是否有人观看都不会关闭, 除非关闭录像功能】	是
recordPath	录像文件保存路径, 如果不配置录像文件保存在应用程序所在的路径下的 record 子路径, 如果配置路径则保存在配置的路径的 record 里面 #注意: 如果需要录像存储, 存储的硬盘千万不要分区, 整个硬盘作为一个区, 因为服务器没有执行两个以上的盘符操作。 #录像保存路径配置 windows 平台的路径配置 比如	否

	D:\video ,Linux 平台配置 /home/video #录像路径使用了默认路径,就一直使用默认路径,如果使用了配置路径就一直使用配置路径。确保使用的路径的硬盘空间为最大的。	
picturePath	#图片抓拍保存路径,如果不填就在应用程序的路径作为根目录,图片文件在./picture,可以跟录像文件设置为同一个根路径,因为图片路径还会创建子路径 ../picture/app/stream,录像文件子路径为 ../record/app/stream	否
captureReplayType	#抓拍图片回复类型 1 回复抓拍产生图片的链接地址 比如: http://192.168.1.218:7088/Media/Camera_00001/2022032008501201.jpg 2 返回抓拍图片的内容即返回抓拍的 jpg 完整图片内容, http 响应返回类型为 Content-Type: image/jpeg	是
deleteSnapPicture	#当 接入视频源注销时或者流媒体服务器安全退出时 删除该路视频源抓拍所产生的所有图片 设置为 0 不删除, 1 删除	
pictureMaxCount	#每一个媒体源允许最大保留的抓拍图片数量,如果超过就会覆盖最老的图片,覆盖方式和录像文件一致,用户根据自己需要估计每天、每周、每月能产生多少张抓拍图片,设置好最大抓拍数量,也就是设置了图片最大保存时长	是
pushEnable_mp4	#rtsp、rtmp 推流接入是否开启 mp4 录像 0 未开启, 1 开启。【注意:代理拉流接入、国标 gb28181 接入如果开启录像则需指定参数 enable_mp4 等于 1】	是
fileSecond	#mp4、fmp4、ts 录制切片时间,单位秒,最大不宜超过 600 秒(10 分钟)【如果采用 ts 切片时长不宜超过 180 秒】,因为时间越长产生的 mp4 文件就越大,在点播时打开录像文件的速度会变慢	是
fileKeepMaxTime	#录像文件最大保留时长,单位小时	是
httpDownloadSpeed	http 方式录像下载速度设置 设置的值越大,下载的速度就越快,用户根据需要调整,范围 1 ~ 10	是
fileRepeat	MP4 点播 (rtsp、rtmp、http-flv、ws-flv、hls)是否循环播放文件 0: 否 1: 是	是
hook_enable	#事件通知部分,当 设置 1 时,开启事件通知,设置 0 时关闭事件通知,通知服务器的地址,端口号一定要配置准确比如: IP: 192.168.1.158 端口 18080	是
noneReaderDuration	#某一路码流持续多少秒无人观看就会发送 无人观看通知消息,单位秒	是
iframeArriveNoticCount	#某一路 i 帧 到达通知次数总数,当超过这数量时不再通知,如果想一直通知可以把 30 修改为一个很大的数量,比如	

httpFlvPort	#http-flv 运行端口,http-flv 拉流时采用该端口 (tcp 类型)	是
webrtcPort	#webrtc 播放视频端口 (tcp 类型)	是
WsRecvPcmPort	#websocket 协议以私有格式接入 pcm 数据	是
hls_enable	#是否启用 hls 切片 , 即 9088 端口服务是否开启 【 1 开启 HLS 切片 , 0 关闭 HLS 切片 】 , 如果 hls_enable 等于 0 , 那么 hlsPort 、 hlsCutType 、 h265CutType 这三个参数失效	是
hlsPort	#hls 运行端口,播放 hls 媒体流时使用该端口 (tcp 类型)	是
hlsCutTime	#hls 切片时间长,单位: 秒 (1 ~ 10 秒)	否
maxTsFileCount	#当切片到硬盘时,最大保留 ts 文件个数	否
hlsCutType	#HLS 切片方式 【hlsCutType 等于 1 设置为切片文件到硬盘 , 如果服务器内存比较小,可以切片文件保存到硬盘】 【hlsCutType 等于 2 设置为切片文件保存到内存中 , 如果内存很大,比如 64G、128G , 推荐切片文件保存到内存】	是
wwwPath	#当切片到硬盘时,可以设置 ts 文件保存的路径,如果需要 hls 切片的视频路数较多,建议切片到内存硬盘上,这样 HLS 切片时最大程度不影响服务器的性能,当 wwwPath 不填写,切片路径就在 当前服务所在路径的 www 子路径里 #sudo mkdir /mnt/ramdisk 创建内存硬盘的路径 , wwwPath 的值可以填写 /mnt/ramdisk #sudo mount -t tmpfs tmpfs /mnt/ramdisk -o size=8G,defaults,noatime,mode=777 创建内存硬盘	
h265CutType	H265 视频切片方式 【1 切片为 TS 文件,2 切片为 mp4 文件 】	是
enable_audio	#是否开启音频,开启音频时,流媒体输出才有音频,否则过滤掉音频数据 enable_audio 等于 1 时开启,enable_audio 等于 0 时,关闭声音	是
IOContentNumber	iocontent 数量	是
ThreadCountOfIOContent	每个 iocontent 上创建线程数量	是
RecvThreadCount	网络数据接收线程数量,可以根据 CPU 核数量 x 2 来配置 , 也可以使用默认参数 64	是
SendThreadCount	网络数据发送线程数量,可以根据 CPU 核数量 x 2 来配置 , 也可以使用默认参数 64	是
RecordReplayThread	录像回放模块线程数量	是

GB28181RtpTCPHeadType	GB28181 TCP 方式发送 rtp(负载 PS)码流时，包头长度选择（1： 4 个字节方式，2： 2 个字节方式）	是
rtpPacketOfRtsp	rtsp 拉流时的 rtp 打包方式，1 为 rtp 负载 ES 流， 2 为 rtp 负载 PS 流 新版（2022-03-24）已经删除 改成程序自动识别	是
ReConnectingCount	代理拉流、推流、国标等等网络连接断线后重连次数，40320 次能尝试重连 7 天 7 夜，用户根据需要修改重连次数	是
maxSameTimeSnap	#图片抓拍最大并发数量限制，意思说服务器最大允许多少路在同时进行抓拍，因为抓拍很消耗硬件资源，需要解码视频为 YUV，再把 YUV 编码为 JPEG 图片，用户调大 maxSameTimeSnap 这个参数，看看服务器能否支撑的住大量并发抓拍，服务器 CPU 性能不同，这个参数也不同	是
snapOutPictureWidth	#输出的抓拍图片宽，0 就是默认的宽，因为视频分辨率不知道，不同的摄像头各个不同，就使用 0 来代替默认原尺寸的宽。如果用户需要指定宽的大小可以填入 1920、1280、960、720、640、352	
snapOutPictureHeight	#输出的抓拍图片高，0 就是默认的高，因为视频分辨率不知道，不同的摄像头各个不同，就使用 0 来代替默认原尺寸的高。如果用户需要指定高的大小可以填入 1080、720、640、576、480、288	
snapObjectDestroy	#抓拍对象是否销毁，如果针对 200 路以内的媒体流进行长期进行高频率的抓拍，可以设置 0，就是每次抓拍完毕后不立即销毁对象，下一次再重新使用抓拍对象。比如针对 200 以内路视频每隔 2 秒进行抓拍图片，可以设置为 0，这样提供抓拍速度。	
snapObjectDuration	#抓拍对象最大保存时长(单位：秒)，当 snapObjectDestroy 等于 0 时，该参数有效，意思说当抓拍对象复用时，最大有效时间内复用抓拍对象才有效，这样做是防止内存泄漏，当超出最长复用时长，对象会自动销毁，再抓拍时会重新创建抓拍对象	
H265ConvertH264_enable	#H265 视频转码部分,H265ConvertH264_enable 等于 0 未开启转码，H265ConvertH264_enable 等于 1 开启转码，这个参数是全局范围，只要设置为 1，所有接入的 H265 视频都会转码为 H264，如果转码部分的 H265 视频需要把 H265ConvertH264_enable 参数 必须设置为 0 #即使 H265ConvertH264_enable 设置为 0，但是在代理拉流接入、国标接入 当指定宽 convertOutWidth、高 convertOutHeight 参数的值时 就会转码该路的 H265 视频，也会覆盖配置文件中默认的视频转码 宽、高，以代理拉流、国标接入的宽、高为准	
H266DecodeCpuGpuType	#H266 转码类型 0 CPU，1 英伟达的 GUP 【注意：要电脑确定安装有英伟达显卡并且安装好驱动情况下才能开启，否则也会切换为 cpu】，2 AMD 的 GPU（尚未支持）【注意：要电脑确定安装有 AMD 显卡并且安装好驱动情况下才能开	

	启，否则也会切换为 cpu 】	
convertOutWidth	#转码输出宽，在 H265 转码为 H264 时，指定输出的宽，比如 1280、800、720、352	
convertOutHeight	#转码输出高，在 H265 转码为 H264 时，指定输出的宽，比如 720、480、480、288、	
convertMaxObject	#支持最大的转码路数，由于防止服务器被转码功能过度使用，设置一个最大转码路数，超过这个路数则 265 的视频不再转码，由于电脑 CPU、GPU 不太相同，这个最大转码路数也会不同，实际商用时需要用户实测之后再修改这个参数值，可能会变大，也可能会变小	
convertOutBitrate	#输出转码视频的码流，单位 K/bps，可以填写 512、768、1024、2048、3072、4096，但是由于转码输出的分辨率不高，不需要填写太大的码率 建议 1024 左右即可	
on_record_ts	#录制 hls ts 切片完成事件通知（当 TS 文件切片至硬盘时才通知，即 hlsCutType 等于 1）注意：如果不需要通知该事件，就把 http://10.0.0.238:4088/index/hook/on_record_ts 删除掉	
H264DecodeEncode_enable	#H264 转码类型 H264DecodeEncode_enable 等于 1 开始转码功能，当输入为 H264，若关闭此功能，则缩放、水印功能会失效	
filterVideo_enable	#水印过滤器 FilterVideo_enable 等于 1 开启水印功能 为 0 时 FilterVideo_text 字段失效 【注意：水印需要重新编解码，水印功能需保证 H265ConvertH264_enable 为 1、H264DecodeEncode_enable 为 1 转码开启】	
filterVideo_text	#水印文字 为空视为不开启水印 长度不超过 64 例如：filterVideo_text=ABL 水印测试 123	
FilterFontSize	#水印字体大小 8、9、12、15、18、20、22、36、48、60 FilterFontSize=30	
FilterFontColor	#水印字体颜色 green, red, blue, yellow, black, white FilterFontColor=red	
FilterFontLeft	#水印字体输出起始 x 坐标 FilterFontLeft=5	
FilterFontTop	#水印字体输出起始 y 坐标 FilterFontTop=5	
FilterFontAlpha	#水印字体文字透明度 0.1、0.2、0.3 ~ 0.9 FilterFontAlpha=0.6	
on_stream_not_arrive	#在代理拉流接入、国标接入中，当申请成功后，但是这一路的码流没有达到时会通知一次，注意：如果不需要通知该事件，就把 http://127.0.0.1:18080/index/hook/on_stream_not_arrive 删除掉	
ps_tsRecvPort	国标单端口接收 ps_tsRecvPort=10000	

MaxDiconnectTimeoutSecond	网络断线超时检测最大时长，意思说最大时长没有网络数据达到就认为已经断线，进行销毁处理 MaxDiconnectTimeoutSecond=24	
ForceSendingIFrame	强制发送 I 帧，当有许多摄像头的 gop 设置为 50，或者需要点播视频时快速的出画面，可以开启强制发送 i 帧。 ForceSendingIFrame=0	
videoFileFormat	#录像文件采用的文件格式 1 为 fmp4 格式，2 为 mp4 格式，3 ts 格式【如果采用 ts 切片，在录像回放时支持 hls 回放，视频画面可以秒开】	
on_record_progress	#录像进度:当前录像文件已经录制了多少秒，总录像时长已经到多少秒：如果不需要通知该事件，就把 http://10.0.0.238:7088/index/hook/on_record_progress 删除掉	
enable_GetFileDuration	#查询录像列表是否获取录像文件真正时长，如果开启获取录像真正时长会造成录像查找比较慢，如果不开启就会返回录像切片时间 fileSecond 的值，【0 默认不开启】，【1 开启 获取真正录像时长】	
on_server_started	#服务器启动后会发送通知消息，告知服务器上线 http://127.0.0.1:18080/index/hook/on_server_started	
on_server_keepalive	#服务器正在运行会发送保活事件，告知服务器正在正常运行，相当于服务器心跳，20 秒发送一次心跳信息，告知服务器处于正常运行，如果不需要该通知，则把 http://127.0.0.1:18080/index/hook/on_server_keepalive 删除	
keepaliveDuration	#流媒体服务器发送心跳时长间隔，单位秒	
on_play	#播放某一路视频时，会发送事件通知，如果不需要 则删除 http://127.0.0.1:18080/index/hook/on_play	
on_publish	#当服务器接入某一路码流时，会发送事件通知，如果不需要则删除 http://127.0.0.1:18080/index/hook/on_publish	
on_delete_record_mp4		

4.对外接口说明

4.1. 开启摄像头录像

4.1.1. 基本信息

请求接口： /index/api/startRecord

请求方式： GET, POST

Content-Type: application/json

注：摄像头录像根据配置文件 fileKeepMaxTime 保留最长时长设置，分片存储，当录像文件达到一个周期时, 第一个生成的文件将被覆盖 (避免频繁创建删除造成磁盘碎片影响性能), 并更新文件名。

4.1.2. 请求参数

4.1.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/startRecord?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=_defaultVhost_&app=Media&stream=Camera_00001

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "vhost": "_defaultVhost_",  
  
  "app": "Media",  
  
  "stream": "Camera_00001"  
}
```

4.1.2.2.参数说明

参数名	示例值	参 数 类 型	是 否 必 填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求 秘钥
vhost	_defaultVhost_	String	是	虚拟主机 标识
app	Media	String	是	应用标识 (可以区分分组)
stream	Camera_00001	String	是	流媒体标 识

4.1.3. 返回参数

4.1.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "MediaSource: /Media/Camera_00001 start Record."  
  
}
```

4.1.3.2.参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number	是	返回标识 0 代表成功，其他都代表失败
memo		String	否	说明信息

4.2. 关闭摄像头录像

4.2.1. 基本信息

请求接口： /index/api/ stopRecord

请求方式： GET, POST

Content-Type: application/json

4.2.2. 请求参数

4.2.2.1.请求样例

➤ GET

http://127.0.0.1:7088/index/api/stopRecord?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=_defaultVhost_&app=Media&stream=Camera_00001

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "vhost": "_defaultVhost_",  
  
  "app": "Media",  
  
  "stream": "Camera_00001"  
}
```

4.2.2.2. 参数说明

参数名	示例值	参 数 类 型	是 否 必 填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求 密钥
vhost	_defaultVhost_	String	是	虚拟主机 标识
app	Media	String	是	应用标识 (可以区分分组)
stream	Camera_00001	String	是	流媒体标 识

4.2.3. 返回参数

4.2.3.1.返回样例

```
{  
  
  "code": 0,  
  
  "memo": " MediaSource: /Media/Camera_00001 stop Record "  
}
```

4.2.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number	是	返回标识 0 代表成功，其他都代表失败
memo		String	否	说明信息

4.3. 查询录像列表

【注意： 1、开始时间必须小于 当前时间减去切片时长的时间 2、从 开始时间 至 结束时间 不能超过 3 天如果查询成功，立即会产生录像回放流 ， 超时 2 分钟如果没有观看，则会自动删除录像码流回放】

4.3.1. 基本信息

请求接口： /index/api/queryRecordList

请求方式： GET, POST

Content-Type: application/json

4.3.2. 请求参数

4.3.2.1.请求样例

➤ GET

http://127.0.0.1:7088/index/api/queryRecordList?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&app=Media&stream=Camera_00001&starttime=20220116154810&endtime=20220116155115

➤ POST

```
{  
  
  "secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "vhost": "_defaultVhost",  
  
  "app": "Media",  
  
  "stream": "Camera_00001",  
  
  "starttime": "20220116154810",  
  
  "endtime": "20220116155115"  
}
```

4.3.2.2. 参数说明

参数名	示例值	参 数 类	是 否 必	参数描述
-----	-----	-------	-------	------

		型	填	
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥
vhost	_defaultVhost_	String	否	虚拟主机标识
app	Media	String	是	应用标识 (可以区分分组)
stream	Camera_00001	String	是	流媒体标识
starttime	20220116154810	String	是	开始时间 (年月日时分秒)
endtime	20220116155115	String	是	结束时间 (年月日时分秒)

4.3.3. 返回参数

4.3.3.1.返回样例

```
{
    "code": 0,                # 查询录像成功，如果查询失败非 0
    "key": 30,                # 产生录像回放对象的 ID，如果不想观看录像码流，可以调用 delMediaStream 函数，并传入该 key 的值，即可删除查询录像产生的媒体流
```

```

        "app": "Media",
        "stream": "Camera_00001_20240420105925-20240420110323", # 查询录像
产生的流 ID 命名规则: stream_开始时间-结束时间
        "starttime": "20240420105925", #开始时间
        "endtime": "20240420110323", #结束时间
        "url":
        { # 该组 url 可以连续播放 20240420105925 ~ 20240420110323
            "rtsp":
"rtsp://192.168.1.238:554/Media/Camera_00001_20240420105925-20240420110323",
[该 url 可以拥有下载 , 调用 controlRecordPlay 函数设置 4 倍速]
            "rtmp":
"rtmp://192.168.1.238:1935/Media/Camera_00001_20240420105925-20240420110323",
            "http-flv":
"http://192.168.1.238:8088/Media/Camera_00001_20240420105925-20240420110323.flv",
            "ws-flv":
"ws://192.168.1.238:6088/Media/Camera_00001_20240420105925-20240420110323.flv",
            "http-mp4":
"http://192.168.1.238:5088/Media/Camera_00001_20240420105925-20240420110323.mp4" [该
url 可以拥有下载 , 调用 controlRecordPlay 函数设置 4 倍速 ]
        },

#文件列表
"recordFileList": [
    {
        "file": "20240420105925.mp4", #录像文件 1
        "duration": 120, #录像文件时长 单位秒
        "url": { #只是单独播放某个文件 下载某个文件
            "rtsp":
"rtsp://192.168.1.238:554/Media/Camera_00001_ReplayFMP4RecordFile__20240420105925",
            "rtmp":
"rtmp://192.168.1.238:1935/Media/Camera_00001_ReplayFMP4RecordFile__20240420105925",
            "http-flv":
"http://192.168.1.238:8088/Media/Camera_00001_ReplayFMP4RecordFile__20240420105925.flv",
            "ws-flv":
"ws://192.168.1.238:6088/Media/Camera_00001_ReplayFMP4RecordFile__20240420105925.flv",
            "http-mp4":
"http://192.168.1.238:5088/Media/Camera_00001_ReplayFMP4RecordFile__20240420105925.mp4",
            "download":
"http://192.168.1.238:5088/Media/Camera_00001_ReplayFMP4RecordFile__20240420105925.mp4?
download_speed=6"
        }
    },
},

```

```

        {
            "file": "20240420110124.mp4", #录像文件 2
            "duration": 120, #录像文件时长 单位秒
            "url": {
                "rtsp":
                "rtsp://192.168.1.238:554/Media/Camera_00001__ReplayFMP4RecordFile__20240420110124",
                "rtmp":
                "rtmp://192.168.1.238:1935/Media/Camera_00001__ReplayFMP4RecordFile__20240420110124",
                "http-flv":
                "http://192.168.1.238:8088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110124.flv",
                "ws-flv":
                "ws://192.168.1.238:6088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110124.flv",
                "http-mp4":
                "http://192.168.1.238:5088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110124.mp4",
                "download":
                "http://192.168.1.238:5088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110124.mp4?download_speed=6"
            }
        },
        {
            "file": "20240420110323.mp4", #录像文件 3
            "duration": 120, #录像文件时长 单位秒
            "url": {
                "rtsp":
                "rtsp://192.168.1.238:554/Media/Camera_00001__ReplayFMP4RecordFile__20240420110323",
                "rtmp":
                "rtmp://192.168.1.238:1935/Media/Camera_00001__ReplayFMP4RecordFile__20240420110323",
                "http-flv":
                "http://192.168.1.238:8088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110323.flv",
                "ws-flv":
                "ws://192.168.1.238:6088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110323.flv",
                "http-mp4":
                "http://192.168.1.238:5088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110323.mp4",
                "download":
                "http://192.168.1.238:5088/Media/Camera_00001__ReplayFMP4RecordFile__20240420110323.mp4?download_speed=6"
            }
        }
    ]
}

```

4.3.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number	是	返回标识 0 代表成功, 其他都代表失败
app	Media	String	是	应用标识(可以区分分组)
stream	Camera_00001	String	是	流媒体标识
starttime	20220116154810	String	是	开始时间(年月日时分秒)
endtime	20220116155115	String	是	结束时间(年月日时分秒)
recordFileList		String[]		文件列表
recordFileList.file	20220116154810.mp4	String		录像文件名称
recordFileList.url		Object		所有录像播放地址

4.4. 录像回放控制

往往用于作为国标下级时，把录像往上级平台推送的暂停、继续、倍速、拖动控制

4.4.1. 基本信息

请求接口：/index/api/controlRecordPlay

请求方式：GET, POST

Content-Type: application/json

4.4.2. 请求参数

4.4.2.1. 请求样例

➤ GET

`http://190.15.240.11:7088/index/api/controlRecordPlay?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=pause` 暂停

`http://190.15.240.11:7088/index/api/controlRecordPlay?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=resume` 继续

`http://190.15.240.11:7088/index/api/controlRecordPlay?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=scale&value=2` 2 倍速回放

`http://190.15.240.11:7088/index/api/controlRecordPlay?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=seek&value=360` 拖拽到 360 秒处继续回放

➤ POST

```
{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc","key":93,"command":"scale","value":2}
```

```
{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc","key":93,"command":"seek","value":360}
```


4.4.2.2. 参数说明

参数名	示例值	参 数 类型	是 否 必 填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥
key	93	String	否	调 用 queryRecordList 返回的 key 的 值
command	Media	String	是	pause 暂停回放 resume 不需要 继续回放 scale 2 2 倍速回放，还有 其他倍数 0.125、 0.25、0.5、1、2、 4、8、16 seek 360 单位秒
value	2	String	是	对应命令值

4.4.3. 返回参数

4.4.3.1. 返回样例

```
{
  "code": 0,           # 0 为操作成功, 其他值为操作失败
  "memo": "success",  # success 为成功 ,如果失败是其他值
}
```

4.4.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number	是	返回标识 0 代表成功, 其他都代表失败
app	Media	String	是	应用标识(可以区分分组)
stream	Camera_00001	String	是	流媒体标识
starttime	20220116154810	String	是	开始时间(年月日时分秒)
endtime	20220116155115	String	是	结束时间(年月日时分秒)
recordFileList		String[]		文件列表

recordFileList.file	20220116154810.mp4	String		录像文件名称
recordFileList.url		Object		所有录像播放 地址

4.5. 添加代理拉流

支持 rtsp、rtmp、flv

4.5.1. 基本信息

请求接口：/index/api/addStreamProxy

请求方式：GET, POST

Content-Type: application/json

4.5.2. 请求参数

4.5.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/addStreamProxy?secret=035c73f7-bb6b-4889-a715-d9e
b2d1925cc&vhost=_defaultVhost_&app=Media&stream=Camera_00001&url=rtsp://ad
min:abldyj2020@192.168.1.120:554&enable_mp4=0

➤ POST

```
{  
  
  "secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "vhost": "_defaultVhost_",  
  
  "app": "Media",  
  
  "stream": "Camera_00001",  
  
  "url": "rtsp://admin:ablcyj2020@192.168.1.120:554",  
  
  "enable_mp4": 0  
}
```

4.5.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889 -a715 -d9eb2d1925cc	String	是	Http 请求密钥
vhost	_defaultVhost_	String	是	虚拟主机标识
app	Media	String	是	应用标识(可以区分分组)
stream	Camera_00001	String	是	流媒体标识
url	rtsp://admin:ablcyj2 020@192.168.1.120:55	String	是	代理拉流 url 比如

	4			rtsp://admin:abldyjh 2020@192.168.1.120: 554 或者 rtmp://190.15.240.36: 1935/Media/Camera _00001 或者 http://190.15.240.36: 8088/Media/Camera _00001.flv 或 d:\\video\\xxxx.mp4 (windows 平 台)/home/video/xxxx .mp4 (Linux 平台)
enable_mp4	0	Number	否	是否录像 (1 录像, 0 不录像)
enable_hls	0	Number	否	1 进行 hls 切 片, 0 不切片
convertOutWidth	960	String	否	转码宽 转码输出宽如果指 定宽、 [1920x1080, 1280x 720, 960x640, 800x 480]就说明 H265 进 行转码为 H264
convertOutHeight	640	String	否	转码高
H264DecodeEncode	1	String	否	H264 是否解码 H264 分辨率高再编 码降分辨率, 有时

_enable				<p>候需要 H264 视频进行先解码再重新编码降低分辨率，可以设置 H264DecodeEncode_enable 为 1，降下来的分辨率为 convertOutWidth、convertOutHeight</p>
disableVideo	是否过滤掉视频	String	否	<p>【可选参数】默认 0 不过滤视频，国家电网协议中需要摄像头和摄像头直接需要对接，摄像头推上来的媒体流中过滤掉视频 (disableVideo=1) 再发给对方才能完成对讲</p>
disableAudio	是否过滤掉音频	String	否	<p>0 不过滤，1 过滤音频 【可选参数】默认 0 不过滤音频，有时候不需要音频、或者由于敏感原因禁止收听音频，可以过滤掉音频流接入流媒体服务器，只</p>

				需 要 设 置 disableAudio = 1 即 可过滤音频
isRtspRecordURL	代理拉流的 url 是否是 rtsp 录像回放的 url 默 认	String	0	代理拉流的 url 是否是 rtsp 录像回放的 url 默 认 0 , 1 是【可选参数】， 如果是 rtsp 录像回放的 url，可以进行控制代理 拉流，比如 暂停、继续、 控制倍速播放，拖动播 放等等 ， 参考 函数 /index/api/controlSt reamProxy
optionsHeartbeat	options 心跳	String		options 命令作为心跳 包 【可选参数】 代理 rtsp 拉流成后，是 否开启发送 0 不开启， 1 开启 ， 默认不开启， 如果从 ZLMediaKit 拉 取 rtsp 流，建议设置为 1

4.5.3. 返回参数

4.5.3.1.返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success ",  
  
  "key": 93  
}
```

4.5.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number	是	返回标识 0 代表成功，其他都代表失败
memo		String	否	说明信息
key	93	Number		成功时返回大于 0 的值，为代理拉流的 Key，删除代理拉流时需要用的

4.6. 删除代理拉流

4.6.1. 基本信息

请求接口：/index/api/delStreamProxy

请求方式：GET, POST

Content-Type: application/json

4.6.2. 请求参数

4.6.2.1.请求样例

➤ GET

http://127.0.0.1:7088/index/api/delStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93

➤ POST

```
{  
  
  "secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "key":93  
}
```

4.6.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d19 25cc	String	是	Http 请求秘钥
key	93	Numb er	是	调 用 添 加 代 理 (/index/api/addStreamPr

				oxy)返回的 key 值
--	--	--	--	---------------

4.6.3. 返回参数

4.6.3.1.返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success"  
}
```

4.6.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.7. 添加 rtsp、rtmp 推流

(注意：不是国标 GB28181 推流) 向其他流媒体推送

4.7.1. 基本信息

请求接口：/index/api/addPushProxy

请求方式：GET，POST

Content-Type: application/json

4.7.2. 请求参数

4.7.2.1.请求样例

➤ GET

http://127.0.0.1:7088/index/api/addPushProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=_defaultVhost_&app=Media&stream=Camera_00001&url=rtsp://190.15.240.36:554/Media/Camera_00001

➤ POST

```
{
  "secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc",
  "vhost":"_defaultVhost_",
  "app":"Media",
  "stream":"Camera_00001",
  "url":"rtsp://190.15.240.36:554/Media/Camera_00001"
}
```

4.7.2.2. 参数说明

参数名	示例值	参 数	是	参数描述
-----	-----	-----	---	------

		类型	否 必 填	
secret	035c73f7-bb6b-4889-a715-d9e b2d1925cc	String	是	Http 请求密钥
vhost	_defaultVhost_	String	否	虚拟主机标识
app	Media	String	是	应用标识(可以区分分组)
stream	Camera_00001	String	是	流媒体标识
url	rtsp://190.15.240.36:554/Media/ Camera_00001			
disableVideo	是否过滤掉视频 0	String	否	0 不过滤 ， 1 过滤视频 【可选参数】默认 0 不过滤视频，国家电网协议中需要摄像头和摄像头直接需要对接，摄像头推上来的媒体流中过滤掉视频（disableVideo=1）再发给对方才能完成对讲
disableAudio	是否过滤掉音频 0	String	否	0 不过滤 ， 1 过滤音频 【可选参数】默认 0 不过滤音频，有时候不需要音频、或者由于敏感原因禁止收听音频，可以过滤掉音频流接入流媒体服务器，只需要设置 disableAudio = 1 即可过滤音频

4.7.3. 返回参数

4.7.3.1.返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success",  
  
  "key": 93  
}
```

4.7.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息
key	93	Number		成功时返回大于 0 的值，为代理推流的 Key，删除代理推流时需要用的

4.8. 删除代理推流

4.8.1. 基本信息

请求接口： /index/api/delPushProxy

请求方式： GET, POST

Content-Type: application/json

4.8.2. 请求参数

4.8.2.1.请求样例

➤ GET

http://127.0.0.1:7088/index/api/delPushProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93

➤ POST

```
{  
  
  "secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "key":93  
}
```

4.8.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必	参数描述

			填	
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求秘钥
key	93	Number	是	调用代理推流 (/index/api/addPushProxy)返回的 key

4.8.3. 返回参数

4.8.3.1.返回样例

```
{
  "code": 0,
  "memo": "success"
}
```

4.8.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都表示失败
memo	success	String		说明信息

4.9. 添加 ffmpeg 代理拉流

调用 ffmpeg 函数实现 rtsp、rtmp、http-flv、http-mp4、http-hls 拉流，建议如果拉取 http-flv、http-mp4、http-hls 码流调用该函数，如果源流是 rtsp、rtmp、本地 MP4 文件 建议使用 /index/api/addStreamProxy，当然如果源流是 rtsp、rtmp，也可以尝试使用 addFFmpegProxy

4.9.1. 基本信息

请求接口：/index/api/addFFmpegProxy

请求方式：GET, POST

Content-Type: application/json

4.9.2. 请求参数

4.9.2.1. 请求样例

➤ GET

http://190.15.240.11:7088/index/api/addFFmpegProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=_defaultVhost_&app=Media&stream=Camera_00001&url=http://190.15.240.36:8088/Media/Camera_00001.flv&enable_mp4=0

➤ POST

```
{  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  "vhost": "_defaultVhost_",  
  "app": "Media",  
  "stream": "Camera_00001",  
  "url": "http://190.15.240.36:8088/Media/Camera_00001.flv",  
}
```



```
"enable_mp4": 0
}
```

4.9.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715 -d9eb2d1925cc	String	是	Http 请求密钥
vhost	_defaultVhost_	String	是	虚拟主机标识
app	Media	String	是	应用标识(可以区分分组)
stream	Camera_00001	String	是	流媒体标识
url	rtsp://admin:abldyj2020@192.168.1.120:554	String	是	代理拉流 url 比如 rtsp://admin:abldyj2020@192.168.1.120:554 或者 rtmp://190.15.240.36:1935/Media/Camera_00001 或者 http://190.15.240.36:8088/Media/Camera

				_00001.flv 或 d:\\video\\xxxx.mp4 (windows 平 台)/home/video/xxxx .mp4 (Linux 平台)
enable_ mp4	0	Number	否	是否录像 (1 录像, 0 不录像)

4.9.3. 返回参数

4.9.3.1.返回样例

```
{  
  "code": 0,  
  "memo": "success",  
  "key": 93  
}
```

4.9.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表 成功, 其他都代 表失败
memo	success	String		说明信息

Key	93	NUmber		# 成功时返回 大于 0 的值, 为 代 理 拉 流 的 Key ,删除代理 拉流时需要用 的
-----	----	--------	--	--

4.10. 删除 ffmpeg 函数代理拉流

4.10.1.基本信息

请求接口: /index/api/delFFmpegProxy

请求方式: GET, POST

Content-Type: application/json

4.10.2. 请求参数

4.10.2.1. 请求样例

➤ GET

http://190.15.240.11:7088/index/api/delFFmpegProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93

➤ POST

```
{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc","key":93}
```

4.10.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥
key	93	Number	是	调用代理推流添加返回的 key

4.10.3. 返回参数

4.10.3.1. 返回样例

```
{
  "code": 0,
  "memo": "success"
}
```

4.10.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败

memo	success	String		说明信息
------	---------	--------	--	------

4.11. 创建 GB28181\Jtt1078 接收端口

4.11.1. 基本信息

请求接口：/index/api/openRtpServer

请求方式：GET, POST

Content-Type: application/json

4.11.2. 请求参数

4.11.2.1. 请求样例

➤ GET

http://190.15.240.11:7088/index/api/openRtpServer?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=_defaultVhost_&app=gb28181&stream_id=44030012343220234234&payload=96&port=0&enable_tcp=0&enable_mp4=0

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "vhost": "_defaultVhost_",  
  
  "app": "gb28181",
```

```

"stream_id": "44030012343220234234",

"payload": 96,

"port": 0,

"enable_tcp": 0,

"enable_mp4": 0

}

```

4.11.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求秘钥
vhost	_defaultVhost_	String	否	虚拟主机标识
app	gb28181	String	是	应用名
stream_id	44030012343220234234	String	是	媒体流名
payload	96	Number	是	PS 负载值(比如 96, 98 , 108 , 一定要从国标 SDP 里面获取)
port	0	Number	是	端口号(0 , 由服务器自动分配, 别的值 比如 26324 为指定端口)
enable_tcp	1	Number	是	0 为 udp , 1 为 tcp 被动方式 , 2 为 tcp 主动连接方式
enable_mp4	0	Number	否	是否录像(1 录像, 0 不录像)
dst_url		String	否	目标 IP 地址 【可选参数 , 当 enable_tcp 为 2 时必须填写】
dst_port		Number		目标端口 【可选参数 , 当 enable_tcp 为 2 时必须填写】
enable_hls	0	Number	否	1 进行 hls 切片 , 0 不切片
convertOutWidth	960	String	否	转码宽 转码输出宽如果指定宽、 [1920x1080, 1280x720, 960x640, 800x480]就说明 H265 进行转码为 H264
convertOutHeight	640	String	否	转码高
H264DecodeEncode_enable	1	String	否	H264 是否解码 H264 分辨率高再编码降分辨率, 有时候需要 H264 视频进行先解码再重新编码降低分辨率, 可以设

				置 H264DecodeEncode_enable 为 1， 降下来的分辨率为 convertOutWidth、convertOutHeight
disableVideo	0	String	否	【可选参数】默认 0 不过滤视频， 国家电网协议中需要摄像头和摄像头 直接需要对接，摄像头推上来的媒体 流中过滤掉视频（disableVideo=1） 再发给对方才能完成对讲
disableAudio	0	String	否	是否过滤掉音频 0 不过滤，1 过滤音频 【可 选参数】默认 0 不过滤音频，有时 候不需要音频、或者由于敏感原因禁 止收听音频，可以过滤掉音频流接入 流媒体服务器，只需要设置 disableAudio = 1 即可过滤音频
RtpPayloadDataType	1	String	否	rtp 打包数据格式 【可选参数，默认 1】[1 PS 国 标 gb28181]、[2 ES 视频支持 H246\H265，音频只支持 G711A、 G711U]、[3 XHB（一家公司的打包 格式）只支持视频，音频不能加入 打包] [4、Jt1078（2016 版本） 码流接入]
jtt1078_version		String	否	指定 1078 版本号 2013、2016、2019 【可 选参数】2013 指定为 2013 版本、 2016 指定为 2016 版本、2019 指 定为 2019 版本
send_app		String	否	应用名 用于把本服务器的码流 （send_app/send_stream_id）回复给 发送端
send_stream_id		String	否	媒体流名 用于把本服务器的码流 （send_app/send_stream_id）回复给 发送端
send_disableVideo		String	否	是否过滤掉视频 默认 0 不过滤视频，国家电网协议中 需要摄像头和摄像头直接需要对接， 摄像头推上来的媒体流中过滤掉视频 （disableVideo=1）再发给对方才能 完成对讲
send_disableAudio		String	否	是否过滤掉音频 是否过滤掉音频 0 不过滤，1 过

				滤音频 【可选参数】 默认 0 不过滤音频，有时候不需要音频、或者由于敏感原因禁止收听音频，可以过滤掉音频流接入流媒体服务器，只需要设置 disableAudio = 1 即可过滤音频
detectSendAppStream	1	String		是否检测发送流 ID 0 不检测 ， 1 检测 【可选参数】 默认 1 在需要回传码流时，事先检测 send_app、send_stream_id 是否存在。0 不检测，先打开接收成功后，稍后再把 send_app、send_stream_id 接入流媒体，openRtpServer 函数自动回复给下级流媒体。

4.11.3.返回参数

4.11.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "port": 8356,  
  
  "memo": "success",  
  
  "key": 93  
}
```

4.11.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代

				表失败
port	8356	Number		端口号
memo	success	String		说明信息
key	93	Number		成功时返回大于 0 的值 , GB28181接收实例 key ,关闭时需要

4.12. 暂停 GB28181\Utt1078 接收端口

说明：只是不检测接收端口是否有码流进入

4.12.1.基本信息

请求接口： /index/api/pauseRtpServer

请求方式： GET, POST

Content-Type: application/json

4.12.2. 请求参数

4.12.2.1. 请求样例

➤ GET

http://44.35.33.249:7088/index/api/pauseRtpServer?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=588

➤ POST

`{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc","key":588}`

4.12.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d192 5cc	String		Http 请求秘钥
key	93	Numbe r		调 用 /index/api/openRtpServ er 返回的 key 的值

4.12.3. 返回参数

4.12.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success"  
}
```

4.12.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.13. 继续 GB28181\Jtt1078 接收端口

4.13.1.基本信息

请求接口： /index/api/resumeRtpServer

请求方式： GET， POST

Content-Type: application/json

4.13.2. 请求参数

4.13.2.1. 请求样例

➤ GET

http://44.35.33.249:7088/index/api/resumeRtpServer?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=588

➤ POST

`{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc","key":588}`

4.13.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d192 5cc	String		Http 请求秘钥
key	93	Numbe r		调 用 /index/api/openRtpServ er 返回的 key 的值

4.13.3. 返回参数

4.13.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success"  
}
```

4.13.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.14. 删除 GB28181 接收端口\Jtt1078 接收端口

4.14.1. 基本信息

请求接口： /index/api/closeRtpServer

请求方式： GET, POST

Content-Type: application/json

4.14.2. 请求参数

4.14.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/closeRtpServer?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93

➤ POST

```
{  
  
  "secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "key":93  
  
}
```

4.14.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d192 5cc	String		Http 请求密钥
key	93	Numbe r		调 用 /index/api/openRtpServ er 返回的 key 的值

4.14.3. 返回参数

4.14.3.1. 返回样例

```
{  
  
  "code":0,
```

```
"memo": "success"
}
```

4.14.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.15. 创建 GB28181 发送端口\Jtt1078 发送端口

4.15.1. 基本信息

请求接口：/index/api/startSendRtp

请求方式：GET, POST

Content-Type: application/json

4.15.2. 请求参数

4.15.2.1. 请求样例

➤ GET

<http://190.15.240.11:7088/index/api/startSendRtp?secret=035c73f7-bb6b-4889-a715-d9>

eb2d1925cc&vhost=_defaultVhost_&app=gb28181&stream=44030012343220234234&payload=96&ssrc=5224&src_port=26324&dst_url=190.15.240.11&dst_port=9824&is_udp=1

➤ POST

```
{
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",
  "vhost": "_defaultVhost_",
  "app": "gb28181",
  "stream": "44030012343220234234",
  "payload": 96,
  "ssrc": "ssrc",
  "src_port": 0,
  "dst_url": "127.0.0.1",
  "dst_port": 5869,
  "is_udp": 1
}
```

4.15.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥
vhost	_defaultVhost	String	否	虚拟主机标识

	—			
app	gb28181	String	是	应用名
stream_id	4403001234322 0234234	String	是	媒体流名
payload	96	Number	是	PS 负载值(比如 96, 98 ,108 ,一定要从国标 SDP 里面获取)
ssrc	ssrc	String	是	ssrc
src_port	0	Number	是	发送端绑定的端口号(指定服务器在发送国标流时绑定的端口号, 如果为 0 , 由服务器自动分配, 别的值 比如 26324 为指定端口)
dst_url	127. 0. 0. 1	String	是	目标 IP 地址
dst_port	5869	Number	是	目标端口
is_udp	1	Number	是	0 TCP 主动方式, 1 udp 方式 , 2 为 TCP 被动方式
disableVideo	0	String	否	【可选参数】 默认 0 不过滤视频, 国家电网协议中需要摄像头和摄像头直接需要对接, 摄像头推上来的媒体流中过滤掉视频 (disableVideo=1) 再发给对方才能完成对讲
disableAudio	0	String	否	是否过滤掉音频 0 不过滤 , 1 过滤音频 【可选参数】 默认 0 不过滤音频, 有时候不需要音频、或者由于敏感原因禁止收听音频, 可以过滤掉音频流接入流媒体服务器, 只需要设置 disableAudio = 1 即可过滤音频
RtpPayloadDataType	1	String	否	【可选参数 , 默认 1 】 [1 PS 国标 gb28181]、[2 ES 视频支持 H246\H265, 音频只支持 G711A、G711U]、[3 XHB (一家公司的打包格式) 只支持视频, 音频不能加入打包] [4 、Jt1078 (2016 版本) 码流发送]
jtt1078_version		String	否	【可选参数】 2013 指定为 2013 版本 、2016 指定为 2016 版本、 2019 指定为 2019 版本
recv_app		String	否	应用名 接入进来 通过在发送的连接中接收进来的码流
recv_stream		String	否	媒体流名 接入进来 通过在发送的连接中接收进来的码流
recv_disableVideo	0	String	否	是否过滤掉视频 默认 0 不过滤视频, 国家电网协议中需要摄像头和摄像头直接需要对接, 摄像头推上来的媒体流中过滤掉视频 (disableVideo=1) 再发给对方才能完成对讲
recv_disableAudio	0	String	否	默认 0 不过滤音频, 有时候不需要音频、或者由于敏感原因禁止收听音频, 可以过滤掉音频流接入流媒体服务器, 只需要设置 disableAudio = 1

				即可过滤音频
--	--	--	--	--------

4.15.3. 返回参数

4.15.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "port": 8356,  
  
  "memo": "success",  
  
  "key": 93  
}
```

4.15.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
port	8356	Number		端口号
memo	success	String		说明信息
key	93	Number		成功时返回大于 0 的值，GB28181发送码

				流实例 key ,关闭时需要
--	--	--	--	----------------

4.16. 删除 GB28181 发送端口\Jtt1078 发送端口

4.16.1. 基本信息

请求接口： /index/api/stopSendRtp

请求方式： GET, POST

Content-Type: application/json

4.16.2. 请求参数

4.16.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/stopSendRtp?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93

➤ POST

```
{  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  "key": 93  
}
```

4.16.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d1925 cc	String		Http 请求密钥
key	93	Numbe r		调 用 /index/api/startSendRt p 返回的 key 的值

4.16.3. 返回参数

4.16.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success"  
}
```

4.16.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表

				成功，其他都代表失败
memo	success	String		说明信息

4.17. 获取流媒体服务器所有可用的媒体源

4.17.1. 基本信息

请求接口： /index/api/getMediaList

请求方式： GET, POST

Content-Type: application/json

4.17.2. 请求参数

4.17.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/getMediaList?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&app=Media&stream=Camera_00001

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",
```

```
"app": "Media",

"stream": "Camera_00001"

}
```

4.17.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求 秘钥
app	Media	String	否	媒体流标识
stream	Camera_00001	String	否	媒体流名

4.17.3. 返回参数

4.17.3.1. 返回样例

```
{
    "code": 0,
    "memo": "success",
    "mediaList": [
        {
            "key": 34,
            "app": "Media",
            "stream": "Camera_00001",
            "sim": "123456789123",
            "status": false,
            "enable_hls": false,
            "transcodingStatus": false,
        }
    ]
}
```

【1078 码流接入这个字段的值不会为空，代表设备的 sim 卡编码】

【false 尚未录像，true 正在录像】

【false 尚未启用，true 启用 hls 回放】

【false 尚未转码，true 正在转

码 **】**

```
        "sourceURL": "rtsp://10.0.0.239:554/Media/Camera_00001",
        "sourceType": 23,
        "readerCount": 0,
        "videoCodec": "H264",
        "width": 1920,
        "height": 1080,
        "networkType": 24,
        "audioCodec": "AAC",
        "audioChannels": 1,
        "audioSampleRate": 16000,
        "url": {
            "rtsp": "rtsp://10.0.0.239:554/Media/Camera_00001",
            "rtmp": "rtmp://10.0.0.239:1935/Media/Camera_00001",
            "http-flv":
"http://10.0.0.239:8088/Media/Camera_00001.flv",
            "ws-flv":
"ws://10.0.0.239:6088/Media/Camera_00001.flv",
            "http-mp4":
"http://10.0.0.239:5088/Media/Camera_00001.mp4",
            "http-hls":
"http://10.0.0.239:9088/Media/Camera_00001.m3u8",

            "webrtc": "http://10.0.0.239:8892/webrtc-streamer.html?video=/Media/Camera_00001"
        }
    }
}
]
```

4.17.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败

memo	success	String		说明信息
mediaList		String[]		可用的媒体源列表
mediaList.key	34	Number		添加时的key
mediaList. app	Media	String		应用名称
mediaList. stream	Camera_00001	String		媒体流名
mediaList. sim	123456789123	String		【1078 码流接入这个字段的值不会为空，代表设备的sim卡编码】
mediaList. status	true	bool		false 尚未录像，true 正在录像
mediaList. sourceURL	rtsp://127.0.0.1:554/Media/Camera_00001	String		添加时的流地址
mediaList.sourceType	23	NUmber		
mediaList. readerCount	0			
mediaList .videoCodec				
mediaList.width				
mediaList. height				
mediaList. networkType				

mediaList. audioCodec				
mediaList. audioChannels				
mediaList. audioSampleRate				
mediaList. url				播放地址列表

4.18. 删除 某一个媒体源

4.18.1.基本信息

请求接口： /index/api/delMediaStream

请求方式： GET, POST

Content-Type: application/json

4.18.2. 请求参数

4.18.2.1. 请求样例

➤ GET

http://190.15.240.11:7088/index/api/delMediaStream?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93

➤ POST

{

```
"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
"key": 93  
  
}
```

4.18.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d1925 cc	String		Http 请求密钥
key	93	Numbe r		调 用 /index/api/getMediaLi st 返回的 key 的值

4.18.3. 返回参数

4.18.3.1. 返回样例

```
{  
  
"code": 0,  
  
"memo": "success"  
  
}
```

4.18.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.19. 获取流媒体服务器所有输出流列表

4.19.1.基本信息

请求接口： /index/api/getOutList

请求方式： GET, POST

Content-Type: application/json

4.19.2. 请求参数

4.19.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/getOutList?secret=035c73f7-bb6b-4889-a715-d9eb2d19

25cc

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc"  
  
}
```

4.19.2.2. 参数说明

参数名	示例值	参 数 类 型	是 否 必 填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String		Http 请求密钥

4.19.3. 返回参数

4.19.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success",  
  
  "outList": [  
  
    {  
  
      "key": 103,  
  
      "app": "Media",  
  
    }  
  
  ]  
  
}
```

```
{
  "stream": "Camera_00001",

  "sourceURL": "rtsp://127.0.0.1:554/Media/Camera_00001",

  "videoCodec": "H264",

  "audioCodec": "AAC",

  "audioChannels": 1,

  "audioSampleRate": 16000,

  "networkType": 24,

  "dst_url": "127.0.0.1",

  "dst_port": 43801

}

]
```

【注释：可以根据 "networkType": 24, 这个字段值区分 媒体输出的类型 ， 具体详见网络类型的对照表 】

4.19.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息
outList				输出流列表

outList.key	103	Number		请求客户端的标识 ID
outList. app	Media	String		应用名称
outList. stream	Camera_00001	String		媒体流
outList. sourceURL	rtsp://127.0.0.1:554 /Media/Camera_00001	String		表示外界以 rtsp 方式向服务器请求码流
outList. videoCodec	H264			
outList. audioCodec	AAC			
outList. audioChannels	1			
outList. audioSampleRate	16000			
outList. networkType	24			网络类型为 24 ，标识为 rtsp 方式
outList. dst_url	127.0.0.1			请求码流客户端 IP
outList. dst_port	10000			请求码流客户端端口

4.20. 删除 某一个服务器所有往外部输出码流列表

4.20.1. 基本信息

请求接口： /index/api/delOutList

请求方式： GET, POST

Content-Type: application/json

4.20.2. 请求参数

4.20.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/delOutList?secret=035c73f7-bb6b-4889-a715-d9eb2d19

25cc&key=93

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "key": 93  
}
```

4.20.2.2. 参数说明

参 数 名	示例值	参 数 类 型	是 否 必 填	参数描述
secre t	035c73f7-bb6b-4889-a715-d9eb2d1925 cc	String		Http 请求密钥
key	93	Numbe r		调 用 /index/api/getMediaLi st 返回的 key 的值

4.20.3. 返回参数

4.20.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success"  
}
```

4.20.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表

				成功，其他都代表失败
memo	success	String		说明信息

4.21. 删除任意一个或一组或者全部媒体输入列表

请求接口：/index/api/close_streams

请求方式：GET，POST

Content-Type: application/json

4.21.1.请求参数

4.21.1.1. 请求样例

➤ GET

http://190.168.24.112:7088/index/api/close_streams?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&app=live&stream=Camera_00001&force=1

➤ POST

```
{
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",
  "vhost": "_defaultVhost_",
  "app": "live",
  "stream": "Camera_00001",
  "force": 1
}
```

}

4.21.1.2. 参数说明

参数名	示例值	参 数 类 型	是否 必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请 求 秘钥
vhost	_defaultVhost_	String	否	虚拟主机标 识
app	live	String	否	应用名称
stream	Camera_00001	String	否	媒体流名
force	1	Number	是	是否强制关 闭 (1 强 制 关闭，不管 是否有人在 观看、0 非 强制关闭， 当有人观看 时不关闭)

4.21.2. 返回参数

4.21.2.1. 返回样例

```
{  
  
  "code": 0,  
  
  "memo": "success"  
}
```

4.21.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.22. 获取系统配置参数

4.22.1. 基本信息

请求接口： /index/api/getServerConfig

请求方式： GET， POST

Content-Type： application/json

4.22.2. 请求参数

4.22.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/getServerConfig?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc"  
}
```

4.22.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String		Http 请求密钥

4.22.3. 返回参数

4.22.3.1. 返回样例

```
{  
  
  "code": 0,  
  
  "params": [  
  
    {  
  
      "enable_audio": 1,  
  
      "memo": "enable_audio = 1 open Audio , enable_audio = 0 Close Audio "  
  
    },  
  
    {  
  
      "httpServerPort": 7088,  
  
      "memo": "http api port "  
  
    },  
  
    {  
  
      "rtspPort": 554,  
  
      "memo": "rtsp port "  
  
    },  
  
    {  
  
      "rtmpPort": 1935,  
  
      "memo": "rtmp port "  
  
    },  
  
  ],  
  
}
```

```
{  
  
  "httpFlvPort": 8088,  
  
  "memo": "http-flv port "  
  
},  
  
{  
  
  "hls_enable": 1,  
  
  "memo": "hls whether enable "  
  
},  
  
{  
  
  "hlsPort": 9088,  
  
  "memo": "hls port"  
  
},  
  
{  
  
  "wsPort": 6088,  
  
  "memo": "websocket flv port"  
  
},  
  
{  
  
  "mp4Port": 5088,  
  
  "memo": "http mp4 port"  
  
},  
  
{  
  
  "hlsCutType": 2,
```

```
    "memo": "hlsCutType = 1 hls cut to Harddisk,hlsCutType = 2  hls cut Media to  
memory"
```

```
  },
```

```
  {
```

```
    "h265CutType": 1,
```

```
    "memo": " 1 h265 cut TS , 2 cut fmp4 "
```

```
  },
```

```
  {
```

```
    "RecvThreadCount": 128,
```

```
    "memo": " RecvThreadCount "
```

```
  },
```

```
  {
```

```
    "SendThreadCount": 128,
```

```
    "memo": "SendThreadCount"
```

```
  },
```

```
  {
```

```
    "GB28181RtpTCPHeadType": 1,
```

```
    "memo": "rtp Length Type"
```

```
  },
```

```
  {
```

```
    "rtpPacketOfRtsp": 1,
```

```
    "memo": "rtsp 1  rtp paylaod ES,2 rtp payload PS"
```

```
},  
  
{  
  
  "ReConnectingCount": 40320,  
  
  "memo": "Try reconnections times ."  
  
},  
  
{  
  
  "maxTimeNoOneWatch": 20,  
  
  "memo": "maxTimeNoOneWatch ."  
  
},  
  
{  
  
  "pushEnable_mp4": 0,  
  
  "memo": "pushEnable_mp4 ."  
  
},  
  
{  
  
  "fileSecond": 120,  
  
  "memo": "fileSecond ."  
  
},  
  
{  
  
  "fileKeepMaxTime": 1,  
  
  "memo": "fileKeepMaxTime ."  
  
},  
  
{
```



```
"httpDownloadSpeed": 6,  
  
"memo": "httpDownloadSpeed ."  
  
}  
  
]  
  
}
```

4.22.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息
params				配置详细

4.23. 图片抓拍

4.23.1. 基本信息

请求接口： /index/api/getSnap

请求方式： GET, POST

Content-Type: application/json

4.23.2. 请求参数

4.23.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/getSnap?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&vhost=_defaultVhost_&app=Media&stream=Camera_00001&timeout_sec=6

➤ POST

```
{
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",
  "vhost": "_defaultVhost_",
  "app": "Media",
  "stream": "Camera_00001",
  "timeout_sec": 10
}
```

4.23.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥
vhost	_defaultVhost_	String	否	虚拟主机标识

app	Media	String	否	媒体流标识
Stream	Camera_00001	String	是	媒体流名
timeout_sec	10	Int	是	即本次抓拍最大超时 时长 单位 秒
captureReplayType	1	Int		1 返回 url , 2 直接返回抓拍的 jpeg 图片, 默认从配置文件读取这个参数值 【可选参数】

4.23.3. 返回参数

4.23.3.1. 返回样例

```
{
  "code": 0,
  "memo": "success , Catpuring takes time 219 millisecond .",
  "url": "http://10.0.0.239:7088/Media/Camera_00001/2022031910034501.jpg"
}
```

4.23.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功, 其他都代表 失败
memo	success	String		说明信息

url		String		图片地址
-----	--	--------	--	------

4.24. 图片列表查询

4.24.1. 基本信息

请求接口： /index/api/queryPictureList

请求方式： GET, POST

Content-Type: application/json

4.24.2. 请求参数

4.24.2.1. 请求样例

➤ GET

http://10.0.0.239:7088/index/api/queryPictureList?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&app=Media&stream=Camera_00001&starttime=20220317081201&endtime=20220319231201

➤ POST

```
{
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",
  "vhost": "_defaultVhost_",
  "app": "Media",
  "stream": "Camera_00001",
  "starttime": "20220317081201",
  "endtime": "20220319231201"
}
```

4.24.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-488 9-a715-d9eb2d1925 cc	String	是	Http 请求秘钥
vhost	_defaultVhost_	String	否	虚拟主机标识
app	Media	String	否	媒体流标识
stream	Camera_00001	String	是	媒体流名
starttime	20220317081201	String	是	开始时间
endtime	20220319231201	String		结束时间

注意：1、开始时间必须小于 当前时间减去切片时长的时间 2、从 开始时间 至 结束时间 不能超过 7 天

4.24.3. 返回参数

4.24.3.1. 返回样例

```
{
  "code": 0,
  "app": "Media",
  "stream": "Camera_00001",
  "starttime": "20220317081201",
  "endtime": "20220319231201",
  "PictureFileList": [
    {
      "file": "2022031816153857.jpg",
      "url": "http://10.0.0.239:7088/Media/Camera_00001/2022031816153857.jpg"
    },
    {
      "file": "2022031816153958.jpg",

```

```

        "url": "http://10.0.0.239:7088/Media/Camera_00001/2022031816153958.jpg"
    },
    {
        "file": "2022031816154059.jpg",
        "url": "http://10.0.0.239:7088/Media/Camera_00001/2022031816154059.jpg"
    }
]
}

```

4.24.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
app	success	String		说明信息
stream		String		图片地址
PictureFileList		Array		图片列表
file		String		文件名称
url				http 地址

4.25. 获取当前转码的数量

4.25.1. 基本信息

请求接口： /index/api/getTranscodingCount

请求方式：GET，POST

Content-Type: application/json

4.25.2. 请求参数

4.25.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/getTranscodingCount?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc

➤ POST

```
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc"}
```

4.25.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求秘钥

4.25.3. 返回参数

4.25.3.1. 返回样例

```
{  
  "code": 0,  
}
```

```
    "currentTranscodingCount": 2
}
```

4.25.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
currentTranscodingCount	5	Number		转码数量

4.26. 列举流媒体服务器所有占用端口

4.26.1. 基本信息

请求接口： /index/api/listServerPort

请求方式：GET, POST

Content-Type: application/json

4.26.2. 请求参数

4.26.2.1. 请求样例

➤ GET

<http://127.0.0.1:7088/index/api/listServerPort?secret=035c73f7-bb6b-4889-a715-d9eb2d192>

5cc

➤ POST

```
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc"}
```

参数填写样例说明：

样例 1（app、stream 都不填写）
secret 035c73f7-bb6b-4889-a715-d9eb2d1925cc
返回所有在线的媒体源

样例 2（只填写 app）
secret 035c73f7-bb6b-4889-a715-d9eb2d1925cc
app rtp
返回 app 等于 rtp 的所有媒体源

样例 3（填写 app = rtp，stream = 44303403343034243200234）
secret 035c73f7-bb6b-4889-a715-d9eb2d1925cc
app rtp
stream 44303403343034243200234
返回 app 等于 rtp、并且 stream 等于 44303403343034243200234 的所有媒体源

样例 4（填写 stream = 44303403343034243200234）
secret 035c73f7-bb6b-4889-a715-d9eb2d1925cc
stream 44303403343034243200234
返回 stream 等于 44303403343034243200234 的所有媒体源

4.26.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-488 9-a715-d9eb2d1925 cc	String	是	Http 请求秘钥
app	Media	String	否	应用名
stream	Camera_00001	String	否	媒体流名

4.26.3. 返回参数

4.26.3.1. 返回样例

```
{
  "code": 0,
  "memo": "success",
  "data": [
    {
      "key": 478,
      "app": "Media",
      "stream": "Camera_00003",
      "networkType": 23,
      "port": 34988
    },
    {
      "key": 477,
      "app": "Media",
      "stream": "Camera_00002",
      "networkType": 23,
      "port": 34986
    },
    {
      "key": 476,
      "app": "Media",
      "stream": "Camera_00001",
      "networkType": 23,
      "port": 34984
    },
    {
      "key": 456,
      "app": "",
      "stream": "",
      "networkType": 68,
      "port": 10000
    }
  ]
}
```

【注释：可以根据 "networkType": 23, 这个字段标识对象的网络类型，具体详见网络类型的对照表】

4.26.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功, 其他都代表失败
currentTranscoding Count	5	Number		转码数量

4.27. 控制代理拉流

4.27.1. 基本信息

请求接口： /index/api/controlStreamProxy

请求方式：GET, POST

Content-Type: application/json

4.27.2. 请求参数

4.27.2.1. 请求样例

➤ GET

http://190.15.240.11:7088/index/api/controlStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=pause 暂停

http://190.15.240.11:7088/index/api/controlStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=resume 继续

http://190.15.240.11:7088/index/api/controlStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=scale&value=2 2 倍速回放

http://190.15.240.11:7088/index/api/controlStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=seek&value=360 拖拽到 360 秒处继续回放

http://190.15.240.11:7088/index/api/controlStreamProxy?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=93&command=seek&value=clock=20230625T113000Z-19700101T000001Z

➤ POST

http URL :

http://190.15.240.11:7088/index/api/controlStreamProxy

body 参数 Json 格式

```
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc", "key": 93, "command": "pause"}
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc", "key": 93, "command": "resume"}
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc", "key": 93, "command": "scale", "value": "2"}
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc", "key": 93, "command": "seek", "value": "360"}
{"secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc", "key": 93, "command": "seek", "value": "clock=20230625T113000Z-19700101T000001Z"}
```

4.27.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥
key	93	String	是	调用 addStreamProxy 返回的 key 的值
command	pause	String	是	命令、值样例列表 command value 注释 pause 不需要 暂停回放 resume 不需要 继续回放 scale 2 2 倍速回放 seek 360 拖拽从开始点计算起到 360 秒处再回放 【ntp 方式拖拽，代表 华为 VCN\3800】 seek

				clock=20230625T113000Z-19700101T000001Z 拖拽到20230625T113000Z 处再回放 【clock 方式拖拽，代表 海康 NVR】
value	2	String	按需	比如 1、2、4、8、16(倍速播放)，npt 方式：360、1800、3600（拖动播放），clock 方式： clock=20230625T113000Z-19700101T000001Z（拖动到：2023-06-25 11:30:00 执行播放） value 为可选参数,当 command 为 pause,resume 时，value 不用，当 command 为 seek,sacale 是,需要填写 value 的值当 command 为 seek,sacale 是需要填写 value 的值

4.27.3. 返回参数

4.27.3.1. 返回样例

```
{
  "code": 0,          # 0 为操作成功，其他值为操作失败
  "memo": "success", # success 为成功，如果失败是其他值
}
```

4.27.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败

memo	success	String		说明信息
------	---------	--------	--	------

4.28. 修改某一路的水印相关参数

4.28.1. 基本信息

请求接口： /index/api/setTransFilter

请求方式： POST

Content-Type: application/json

4.28.2. 请求参数

4.28.2.1. 请求样例

➤ POST

```
{  
  
  "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc",  
  
  "app": "live",  
  
  "stream": "test",  
  
  "text": "ABL",  
  
  "size": 60,  
  
  "color": "red",  
  
  "alpha": 0.8,  
  
  "left": 40,  
  
  "top": 40,
```

```
"trans": 1  
}
```

4.28.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-488 9-a715-d9eb2d1925 cc	String	是	Http 请求密钥
app	rtp、gb28181、 Media 等等	String	是	应用名
stream	比如 Camera_00001、 4430340334303424 3200234 等等	String	是	媒体流名
text	水印内容 比如	String	是	某某市某某公安局
size		Int	是	字体大小 20、 30 、 40 、 50
color		String		字 体 颜 色 red,green,blue,white,black
alpha		String		透明度 0.1 ~ 0.9

left		String		水印 x 坐标 比如 5 、 10 、 20
top		String		水印 y 坐标 比 如 5 、 10 、 20
trans		String		是否转换 固定 为 1

4.28.3. 返回参数

4.28.3.1. 返回样例

```
{  
    "code": 0,           # 0 为操作成功，其他值为操作失败  
    "memo": "success",  # success 为成功，如果失败是其他值  
}
```

4.28.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功， 其他都代表 失败
memo	success	String		说明信息

4.29. 为调试视频需要增加了一个接口来打开代理拉流

为调试视频需要增加了一个接口来打开代理拉流，国标接入原始视频保存，默认是关闭

的【只用于调试，正式运行不需要此功能】

4.29.1. 基本信息

请求接口： /index/api/setConfigParamValue

请求方式： GET

Content-Type: application/json

4.29.2. 请求参数

4.29.2.1. 请求样例

➤ GET

比如：

打开保存国标 PS 标志

http://44.35.33.239:7088/index/api/setConfigParamValue?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=saveGB28181Rtp&value=1

关闭存国标 PS 标志

http://44.35.33.239:7088/index/api/setConfigParamValue?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=saveGB28181Rtp&value=0

打开保存代理拉 r t s p 流标志

http://44.35.33.239:7088/index/api/setConfigParamValue?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&key=saveProxyRtspRtp&value=1

关闭保存代理拉 r t s p 流标志

http://44.35.33.239:7088/index/api/setConfigParamValue?secret=035c73f7-bb6b-4889-
a715-d9eb2d1925cc&key=saveProxyRtspRtp&value=0

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求秘钥
vhost			否	比如 _defaultVhost_
app	rtp、gb28181、Media 等等	String	是	应用名
key		String	是	比如 saveGB28181Rtp (保存接入的国标 PS 流)、 saveProxyRtspRtp (保存 rtsp 代理拉流的 rtp 流)
value	水印内容 比如	String	是	参数值 1 为打开

				0 为关闭
--	--	--	--	-------

4.29.2.2. 参数说明

4.29.3. 返回参数

4.29.3.1. 返回样例

```
{
  "code": 0,          # 0 为操作成功，其他值为操作失败
  "memo": "success",  # success 为成功，如果失败是其他值
}
```

4.29.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	success	String		说明信息

4.30. 安全停止服务器

安全停止服务器，如果服务器正在录像、抓拍等等操作，需要调用该函数安全停止服务器，这样录制的 mp4 才能正常播放

4.30.1. 基本信息

请求接口： /index/api/shutdownServer

请求方式：GET，POST

Content-Type: application/json

4.30.2. 请求参数

4.30.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/shutdownServer?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc

➤ POST

```
{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc"}
```

4.30.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥

4.30.3. 返回参数

4.30.3.1. 返回样例

```
{
    "code": 0,
    "memo": "ABLMediaServer shutdown Succeeded !"
}
```

}

4.30.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，其他都代表失败
memo	ABLMediaServer shutdown Succeeded !	String		说明信息

4.31. 重新启动服务器

安全重启服务器，如果服务器正在录像、抓拍等等操作，需要调用该函数安全重启服务器，这样录制的 mp4 才能正常播放

4.31.1.基本信息

请求接口： /index/api/restartServer

请求方式：POST

Content-Type: application/json

4.31.2. 请求参数

4.31.2.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/restartServer?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc

➤ POST

`{"secret":"035c73f7-bb6b-4889-a715-d9eb2d1925cc"}`

4.31.2.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	Http 请求密钥

4.31.3. 返回参数

4.31.3.1. 返回样例

```
{
  "code": 0,           # 0 为操作成功，其他值为操作失败
  "memo": "ABLMediaServer restartServer Succeeded！", # success 为成功，如果失败是其他值
}
```

4.31.3.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
code	0	Number		返回标识 0 代表成功，

				其他都代表 失败
memo	ABLMediaServer restartServer Succeeded !	String		说明信息

4.32. 消息通知（主动推送）

4.32.1. 功能说明

消息通知是流媒体服务器的一些消息比如无人观看、fmp4 录像切片完成、播放时流地址不存在，等信息能及时的通知到另外一个 http 服务器上，比如说无人观看消息通知，当收到无人观看消息时，国标服务器可以关闭国标发流，断开代理拉流，断开推流等等操作等，业务系统可以使用该功能及时了解流媒体服务处理状态。开启此功能需要在配置文件的参数 hook_enable 值设置为 1，同时通知的 http 服务器地址、端口号一定要设置对，下面列举出配置文件中的相关参数：

#事件通知部分,当 hook_enable=1 时，开启事件通知，hook_enable=0 时关闭事件通知，通知服务器的地址，端口号一定要配置准确比如：IP：192.168.1.158 端口 80

hook_enable=0

#某一路码流持续多少秒无人观看就会发送 无人观看通知消息，单位秒

noneReaderDuration=30

#当某一路的码流达到时会通知一次

on_stream_arrive=http://192.168.194.111:8080/index/hook/on_stream_arrive

#当某一路流无人观看时，会触发该通知事件，接收端收到后可以进行断流操作，比如关闭
国标发流，断开代理拉流，断开推流等等操作

on_stream_none_reader=http://192.168.194.111:8080/index/hook/on_stream_none_reade
r

#当某一路码流断开时会通知一次

on_stream_disconnect=http://192.168.194.111:8080/index/hook/on_stream_disconnect

#播放时，找不到播放的码流，通过配合 on_stream_none_reader 事件可以完成按需拉流

on_stream_not_found=http://192.168.194.111:8080/index/hook/on_stream_not_found

#录制完毕一段 mp4 文件通知

on_record_mp4=http://192.168.194.111:8080/index/hook/on_record_mp4 #录制完毕一
段 mp4 文件通知

4.32.2. 请求示例

4.32.2.1. 当某一路码流到达时会发送通知

```
# 根据 /index/hook/on_stream_arrive 这个可以判断是某一路码流到达
POST /index/hook/on_stream_arrive HTTP/1.1
Accept: */*
Accept-Language: zh-CN,zh;q=0.8
Connection: keep-alive
Content-Length: 105
Content-Type: application/json
Host: 127.0.0.1
Tools: ABLMediaServer-5.2.9(2022-03-28)
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/57.0.2987.133 Safari/537.36
```

```
{
  "app": "Media",
  # app
```



```

        "stream": "Camera_00001",          # stream
        "mediaServerId": "ABLMediaServer_00001", # 流媒体服务器编号，在
配置文件可以配置，如果集群流媒体服务器时，可以给每台流媒体服务器起个编号
        "networkType": 23, # 媒体流来源网络编号，可参考附表
        "key": 130          # 媒体流来源编号，可以根据这个 key 进行关闭流媒体
可以调用 delMediaStream 或 close_streams 函数进行关闭
    }

```

4.32.2.2. 无人观看消息通知样例

根据 /index/hook/on_stream_none_reader 这个可以判断是无人观看消息通知

```

POST /index/hook/on_stream_none_reader HTTP/1.1

Accept: */*

Accept-Language: zh-CN,zh;q=0.8

Connection: keep-alive

Content-Length: 105

Content-Type: application/json

Host: 127.0.0.1

Tools: ABLMediaServer-5.2.9(2022-03-28)

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.
36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{
    "app": "Media",          # app
    "stream": "Camera_00001",          # stream
    "mediaServerId": "ABLMediaServer_00001", # 流媒体服务器编号，在配置文件可以配置，
如果集群流媒体服务器时，可以给每台流媒体服务器起个编号
    "networkType": 23,          # 媒体流来源网络编号，可参考附表
    "key": 130          # 媒体流来源编号，可以根据这个 key 进行关闭流
媒体 可以调用 delMediaStream 或 close_streams 函数进行关闭
}

```

4.32.2.3. fmp4 录像切片录像完成一个文件时会发送一个消息通知

根据 /index/hook/on_record_mp4 这个可以判断是 mp4 录像切片完毕一个通知

POST /index/hook/on_record_mp4 HTTP/1.1

Accept: */*

Accept-Language: zh-CN,zh;q=0.8

Connection: keep-alive

Content-Length: 127

Content-Type: application/json

Host: 127.0.0.1

Tools: ABLMediaServer-5.2.9(2022-03-28)

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

```
{
  "app": "Media",                                # app
  "stream": "Camera_00001",                      # stream
  "mediaServerId": "ABLMediaServer_00001", # 流媒体服务器编号，在配置文件可以配置，如果集群流媒体服务器时，可以给每台流媒体服务器起个编号
  "networkType": 70,                             # 媒体流来源网络编号，可参考附表
  "fileName": "20220312212546.mp4"              # 录像切片完成的文件名字
}
```

4.32.2.4. 当某一路码流断开时会发送通知

根据 /index/hook/on_stream_disconnect 这个可以判断是某一路码流断开

POST /index/hook/on_stream_disconnect HTTP/1.1

Accept: */*

Accept-Language: zh-CN,zh;q=0.8

Connection: keep-alive

Content-Length: 105

Content-Type: application/json

Host: 127.0.0.1

Tools: ABLMediaServer-5.2.9(2022-03-28)

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

```
{
  "app": "Media",                                # app
  "stream": "Camera_00001",                      # stream
  "mediaServerId": "ABLMediaServer_00001",      # 流媒体服务器编号，在配置文件可以配置，如果集群流
  媒体服务器时，可以给每台流媒体服务器起个编号
  "networkType": 23,                             # 媒体流来源网络编号，可参考附表
  "key": 130                                     # 媒体流来源编号，
}
```

4.32.2.5. 当播放一个 url，如果不存在时，会发出一个消息通知

```
# 根据 /index/hook/on_stream_not_found，Http 接收服务器得知流不不存在
POST /index/hook/on_stream_not_found HTTP/1.1
Accept: */*
Accept-Language: zh-CN, zh;q=0.8
Connection: keep-alive
Content-Length: 127
Content-Type: application/json
Host: 127.0.0.1
Tools: ABLMediaServer-5.2.9(2022-03-28)
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/57.0.2987.133 Safari/537.36
```

```
{
  "app": "Media",                                # app    不存在的 app
  "stream": "Camera_00001",                      # stream  不存在的 stream
  "mediaServerId": "ABLMediaServer_00001"      # 流媒体服务器编号，在配置文件可以配置，如
  果集群流媒体服务器时，可以给每台流媒体服务器起个编号
}
```

4.32.2.6. ABLMediaServer 启动时会发送上线通知

```
POST /index/hook/on_server_started HTTP/1.1
Accept: */*
Accept-Language: zh-CN, zh;q=0.8
Connection: keep-alive
Content-Length: 105
Content-Type: application/json
Host: 127.0.0.1
Tools: ABLMediaServer-6.3.5(2023-04-30)
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36
```

```
{"localIpAddress":"44.35.33.249","mediaServerId":"ABLMediaServer_00001","datetime":  
"2023-04-18 10:04:37"}
```

4.32.2.7. 某一个码流接入流媒体服务器时会产生 发布 事件通知

POST /index/hook/on_publish HTTP/1.1

Accept: */*

Accept-Language: zh-CN, zh;q=0.8

Connection: keep-alive

Content-Length: 149

Content-Type: application/json

Host: 127.0.0.1

Tools: ABLMediaServer-6.3.5(2023-04-30)

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

```
{"app":"Media","stream":"Camera_00001","mediaServerId":"ABLMediaServer_00001","netw  
orkType":50,"key":470,"ip":"44.35.33.248","port":554,"params":""}
```

4.32.2.8. 当播放流媒体服务器里面某一个码流时，会触发 播放事 件，可以通知 parmas 参数值进行播放鉴权

POST /index/hook/on_play HTTP/1.1

Accept: */*

Accept-Language: zh-CN, zh;q=0.8

Connection: keep-alive

Content-Length: 135

Content-Type: application/json

Host: 127.0.0.1

Tools: ABLMediaServer-6.3.5(2023-04-30)

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

```
{"app":"Media","stream":"Camera_00001","mediaServerId":"ABLMediaServer_00001","netw  
orkType":30,"key":472,"ip":"","port":0,"params":"user=admin&password=safsa234234&token  
=@#$@#$@#ASDFasfdsa@#$23"}
```

4.32.2.9. 如果某一个码流进行 MP4 录像 (enable_mp4=1) , 会触发录像进度通知事件

```
POST /index/hook/on_record_progress HTTP/1.1
    Accept: */*
    Accept-Language: zh-CN, zh;q=0.8
    Connection: keep-alive
    Content-Length: 204
    Content-Type: application/json
    Host: 127.0.0.1
    Tools: ABLMediaServer-6.3.5(2023-04-30)
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{"app": "Media", "stream": "Camera_00001", "mediaServerId": "ABLMediaServer_00001", "networkType": 472, "key": 470, "fileName": "20230418100440.mp4", "currentFileDuration": 49, "TotalVideoDuration": 49}
```

4.32.2.10. 某一个接入码流当 i 帧到达会发出通知 , 总共通知默认 30 次, 也可以修改总共通知次数

```
POST /index/hook/on_stream_iframe_arrive HTTP/1.1
    Accept: */*
    Accept-Language: zh-CN, zh;q=0.8
    Connection: keep-alive
    Content-Length: 161
    Content-Type: application/json
    Host: 127.0.0.1
    Tools: ABLMediaServer-6.3.6(2024-04-02)
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{"app": "Media", "stream": "Camera_00001", "mediaServerId": "ABLMediaServer_00001", "networkType": 23, "key": 18, "ip": "44.35.33.249", "port": 65183, "NotificationNumber": 1}
```

4.32.2.11. 如果某一个码流开启 hls (enable_hls=1) 并且设置为切片到硬盘 (hlsCutType=1 在配置文件中) , 每当切片完成一个 ts 文件会触发该事件

```
POST /index/hook/on_record_ts HTTP/1.1
    Accept: */*
    Accept-Language: zh-CN, zh;q=0.8
    Connection: keep-alive
    Content-Length: 246
    Content-Type: application/json
    Host: 127.0.0.1
    Tools: ABLMediaServer-6.3.5(2023-04-30)
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{"app": "Media", "stream": "Camera_00001", "mediaServerId": "ABLMediaServer_00001", "networkType": 30, "key": 470, "createDateTime": "2023-04-18 10:05:47", "currentFileDuration": 3, "fileName": "D:\\WorkDir\\ABLMediaServer\\x64\\Debug\\www\\Media\\Camera_00001\\23.ts"}
```

4.32.2.12. 流媒体服务器每隔 60 秒会触发心跳事件通知

```
POST /index/hook/on_server_keepalive HTTP/1.1
    Accept: */*
    Accept-Language: zh-CN, zh;q=0.8
    Connection: keep-alive
    Content-Length: 105
    Content-Type: application/json
    Host: 127.0.0.1
    Tools: ABLMediaServer-6.3.5(2023-04-30)
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{"localIpAddress": "44.35.33.249", "mediaServerId": "ABLMediaServer_00001", "datetime": "2023-04-18 10:25:40"}
```

4.32.2.13. 当代理拉流、国标接入等等 码流不到达时会发出 码流不到达的事件通知

```
POST /index/hook/on_stream_not_arrive HTTP/1.1
    Accept: */*
    Accept-Language: zh-CN, zh;q=0.8
    Connection: keep-alive
    Content-Length: 155
    Content-Type: application/json
    Host: 127.0.0.1
    Tools: ABLMediaServer-6.3.5(2023-04-30)
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{"mediaServerId":"ABLMediaServer_00001","app":"gb28181","stream":"44030012343220234
2349","networkType":60,"key":470}
```

4.32.2.14. 如果某一个码流进行 MP4 录像 (enable_mp4=1) , 当某个 MP4 文件被删除会触发该事件通知

```
POST /index/hook/on_delete_record_mp4 HTTP/1.1
    Accept: */*
    Accept-Language: zh-CN, zh;q=0.8
    Connection: keep-alive
    Content-Length: 204
    Content-Type: application/json
    Host: 127.0.0.1
    Tools: ABLMediaServer-6.3.5(2023-04-30)
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_1)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36

{"app":"Media","stream":"Camera_00001","mediaServerId":"ABLMediaServer_00001","file
Name":"20230418100440.mp4"}
```

4.33. 批量设置服务器的配置参数

请求接口: /index/api/setServerConfig

请求方式：POST
Content-Type: application/json

4.33.1. 请求参数

4.33.1.1. 请求样例

➤ GET

http://127.0.0.1:7088/index/api/setServerConfig?secret=035c73f7-bb6b-4889-a715-d9eb2d1925cc&noneReaderDuration=20&on_server_started=http://10.0.0.238:4088/index/hook/on_server_started&iFrameArriveNoticCount=30

➤ POST

```
{ "secret": "035c73f7-bb6b-4889-a715-d9eb2d1925cc", "noneReaderDuration": 20, "iFrameArriveNoticCount": 30, "on_server_started": "http://10.0.0.238:4088/index/hook/on_server_started" }
```

4.33.1.2. 参数说明

参数名	示例值	参数类型	是否必填	参数描述
secret	035c73f7-bb6b-4889-a715-d9eb2d1925cc	String	是	服务器密码
noneReaderDuration	20	Number	否	多少秒无人观看通知一次 20
on_server_started	http://10.0.0.238:4088/index/hook/on_server_started	String		服务器启动通知
iFrameArriveNoticCount	30	Number		i 帧达到通知总数

5. 网络类型的对照表

1. 媒体输入网络类型对照表

序号	整形值	代表意义
1	21	以 rtmp 方式推送接入流媒体服务器
2	23	以 rtsp 方式推送接入流媒体服务器
3	30	服务器以 rtsp 方式主动拉流接入
4	31	服务器以 rtmp 方式主动拉流接入
5	32	服务器以 flv 方式主动拉流接入
6	33	服务器以 hls 方式主动拉流接入
7	60	服务器以国标 28181 的 UDP 方式接入
8	61	服务器以国标 28181 的 TCP 方式接入
9	80	服务器录像文件点播以读取 fmp4 文件输入
10	81	服务器录像文件点播以读取 TS 文件输入
11	82	服务器录像文件点播以读取 PS 文件输入
12	83	服务器录像文件点播以读取 FLV 文件输入

2. 媒体输出网络类型对照表

序号	整形值	代表意义
1	22	服务器以 rtsp 被动方式往外发送码流，即常见的 vlc 点播
2	24	服务器以 rtmp 被动方式往外发送码流，即常见的 vlc 点播

3	25	服务器以 flv 被动方式往外发送码流,即常见的 vlc 点播、浏览器播放
4	26	服务器以 hls 被动方式往外发送码流 , 即常见的 vlc 点播 、 浏览器播放
5	27	服务器以 ws-flv 被动方式往外发送码流 ,EasyPlayer.js 插件播放、浏览器播放
6	28	服务器以 http-mp4 被动方式往外发送码流 , 即常见的 vlc 点播 、 浏览器播放
7	40	服务器以 rtsp 主动方式往外发送码流 , 即常见的 rtsp 推流
8	41	服务器以 rtmp 主动方式往外发送码流 , 即常见的 rtmp 推流
9	65	服务器以国标 GB28181 主动 UDP 方式往外发送码流 , 即常见的国标以 UDP 方式往上级推流
10	66	服务器以国标 GB28181 主动 TCP 方式往外发送码流 , 即常见的国标以 TCP 方式往上级推流

6. 国标单端口接入

6.1. 服务器支持国标单端口接入特殊说明

为了方便某些特殊场合, 服务器支持 udp 的 10000 端口接入国标 PS 码流, 就是人们常说的单端口模式, url 的命名规则为 /rtp/ssrc, 其中 ssrc 为下级 rtp 打包的 16 进制的值转换为大小的字符串, 即可 `sprintf(url, "rtp/%X", ssrc)`, 具体接入的 url 名字可以调用 `getMediaList` 查询出接入的国标流。需要注意的是下级 rtp 打包时每路视频的 rtp 中的 ssrc 不能相同。

7. 版本记录

7.1. 2022-01-25 发布

- 1、完成 RTSP/RTMP/国标 GB28181 输入, rtsp、rtmp、http-flv 拉流输入
- 2、完成 RTSP/RTMP/国标 GB28181 输入, rtsp、rtmp、http-flv 、http-hls、http-mp4
websocket-flv 输出
- 3、流媒体录像，录像查询，录像点播，录像下载、录像自动删除，录像定时切片
- 4、支持按需拉流，没有播放自动关闭拉流
- 5、支持 视频：H264、H265 ， 音频：AAC、G711A、G711U

7.2. 2022-03-28 发布

- 1、增加国标 gb28181 以 udp 方式接收模式的 rtcp 回复保活
- 2、增加 rtsp 代理拉流自动判断负载是 ES 还是 PS 流类型，因为海康 NVR 的 rtsp 输出实况是 ES 流，录像回放是 PS 支持海康 NVR 录像回放以 rtsp 协议负载是 PS 码流接收接入媒体服务器。
- 3、优化 mp4 录像文件以（rtsp、rtmp、flv、ws-flv）点播回放更加流畅，流畅程度达到实况播放水准。
- 4、兼容停车场管理系统保有量较高的车牌识别摄像机（臻识）以 rtsp 协议代理拉流接入流媒体服务器。
- 5、在媒体流输出支持 g711 转 AAC 输出，这样以各种方式接入的 g711a、g711u 音频都可以转换为 AAC，在浏览器播放 http-flv、ws-flv、hls、mp4 也就会有声音录制出来的 mp4 录像文件音频格式也转为 AAC。需要此转码功能只要把配置参 G711ConvertAAC 的值修改为 1 即可，默认为 0 即不开启 g711 转 AAC 。

6、 增加事件通知功能（默认关闭，修改参数即可开启）支持 无人观看事件 通知支持 录像完毕一个 mp4 文件 通知支持 没有找到媒体源 通知。

7、 增加业界最快的图片抓拍功能，毫秒级返回抓拍图片支持图片按开始时间、结束时间查找抓拍图片支持下载、浏览已抓拍的图片支持设置每路流最大图片抓拍数量，超过了循环覆盖支持设置抓拍返回图片 url、图片数据两种返回类型。

7.3. 2022-04-11 发布

1、 代理拉流（rtsp、rtmp、flv）代理推流（rtsp、rtmp）的 url 中增加支持固定域名形式，也支持动态域名，定期 15 秒更新动态域名的 IP 地址。也就是说以前版本代理拉流、代理推流的 url 中只能是 ip 地址，现在的版本支持固定域名，也支持动态域名并会动态定期更新域名的 ip

2、 增加两个事件通知

1) 媒体源码流到达事件通知（关键字 app、stream）

2) 媒体源网络断开事件通知（关键字 app、stream）

3、 http 请求 post 方式中 Body 的数据类型 Content-Type 增加支持 application/x-www-form-urlencoded 这样格式的数据，如 key1=value1&key2=value2&key3=value3 方便用户用 wvp、wvp-pro 和 ABLMediaServer 对接，wvp 是 post 过来的 body 数据是本次发版新增加支持的格式。

4、 删除掉 http 请求 参数、值 的空格，包括 get、post 两个方式。用户在填写参数、值时误操作两端多了空格也不影响 http 请求的结果。

7.4. 2022-05-06 发布

一、增加功能

1、 http-flv、http-mp4、http-hls 支持跨域功能

- 2、增加抓拍输出图片宽、高可以让用户设置，统一大小输出。
- 3、增加设置最大抓拍并发数量，防止并发过多的抓拍造成服务器卡顿，媒体流输出缓慢。
- 4、支持图片抓拍对象复用功能，如果开启复用，最快能在 31 毫秒返回抓拍图片。
- 5、支持复用抓拍对象重用最大时长 120 秒，防止复用抓拍对象过多不释放对象对象造成服务器内存泄漏。

二、增加参数

maxSameTimeSnap=16

#图片抓拍最大并发数量限制，意思说服务器最大允许多少路在同时进行抓拍，因为抓拍很消耗硬件资源，需要解码视频为 YUV，再把 YUV 编码为 JPEG 图片，用户调大 maxSameTimeSnap 这个参数，看看服务器能否支撑的住大量并发抓拍，服务器 CPU 性能不同，这个参数也不同。

snapOutPictureWidth=0

#输出的抓拍图片宽，0 就是默认的宽，因为视频分辨率不知道，不同的摄像头各个不同，就使用 0 来代替默认原尺寸的宽。如果用户需要指定宽的大小可以填入 1920、1280、960、720、640、352。

snapOutPictureHeight=0

#输出的抓拍图片高，0 就是默认的高，因为视频分辨率不知道，不同的摄像头各个不同，就使用 0 来代替默认原尺寸的高。如果用户需要指定高的大小可以填入 1080、720、640、576、480、288

snapObjectDestroy=1

#抓拍对象是否销毁，如果针对 200 路以内的媒体流进行长期进行高频率的抓拍，可以

设置 0，就是每次抓拍完毕后不立即销毁对象，下一次再重新使用抓拍对象。比如针对 200 以内路视频每隔 2 秒进行抓拍图片，可以设置为 0，这样提供抓拍速度。

`snapObjectDuration=120`

#抓拍对象最大保存时长(单位：秒)，当 `snapObjectDestroy = 0` 时，该参数有效，意思说当抓拍对象复用时，最大有效时间内复用抓拍对象才有效，这样做是防止内存泄漏，当超出最长复用时长，对象会自动销毁，再抓拍时会重新创建抓拍对象

7.5. 2022-06-30 发布

1、新增配置参数（详见第 3 节）

#H265 视频转码部分，`H265ConvertH264_enable = 0` 未开启转码，`H265ConvertH264_enable = 1` 开启转码 注：当 `H265ConvertH264_enable = 0` 时，`H256DecodeCpuGpuType`、`convertOutWidth`、`convertOutHeight`、`convertMaxObject`、`convertOutBitrate` 这 4 个参数失效

`H265ConvertH264_enable=0`

#H265 转码类型 0 CPU,1 英伟达的 GUP 【注意：要电脑确定安装有英伟达显卡并且安装好驱动情况下才能开启，否则也会切换为 cpu】，2 AMD 的 GPU (尚未支持) 【注意：要电脑确定安装有 AMD 显卡并且安装好驱动情况下才能开启，否则也会切换为 cpu】

`H256DecodeCpuGpuType=0`

#转码输出宽，在 H265 转码为 H264 时，指定输出的宽，比如 1280、800、720、352、
`convertOutWidth=720`

#转码输出高，在 H265 转码为 H264 时，指定输出的宽，比如 720、480、480、288、
`convertOutHeight=480`

#支持最大的转码路数，由于防止服务器被转码功能过度使用，设置一个最大转码路数，

超过这个路数则 265 的视频不再转码，由于电脑 CPU、GPU 不太相同，这个最大转码路数也会不同，实际商用时需要用户实测之后再修改这个参数值，可能会变大，也可能会变小

```
convertMaxObject=26
```

#输出转码视频的码流，单位 K/bps，可以填写 512、768、1024、2048、3072、4096，但是由于转码输出的分辨率不高，不需要填写太大的码率 建议 1024 左右即可

```
convertOutBitrate=512
```

#录制 hls ts 切片完成事件通知（当 TS 文件切片至硬盘时才通知，即 hlsCutType=1）

注意：如果不需要通知该事件，就把 http://10.0.0.238:7088/index/hook/on_record_ts 删掉

```
on_record_ts=http://10.0.0.238:7088/index/hook/on_record_ts
```

2、新增接口，控制代理拉流 详见 [4.21](#) 章节。

3、其他功能更新

1、加大底层网络库 udp 方式接收、发送缓冲区大小

2、国标 UDP 方式接收时主动发送 rtcp 接收者报告包

3、支持代理拉流接入海康 NVR 录像回放的 rtsp 流，支持 1、2、4、8、16 倍速播放、拖动播放

4、windows 平台版本首先支持转 cpu 软转码、gpu 硬转码

1) 支持设置 cpu、gpu 转码选项

2) 支持多张英伟达显卡同时硬解转码，每一块英伟达 2080 显卡能转码 25 路 1080p 的 h265 视频，两块能同时转码 50 路，达到商用数量级别。

3) 支持设置转码输出视频分辨率、宽高、码率大小实测 linux 平台能最大并发转码 40 路 H265（硬件环境 至强 E5 2650 V3）实测 windows 平台能最大并发转码 35 路

H265 (i9 + 英伟达 RTX 2080)

7.6. 2022-08-04 发布

1、支持转码输出的视频打入水印，水印字符的内容、字体大小、颜色、输出的位置可以在配置文件中设置。

2、视频水印同时支持修改任意一路的视频水印内容、字体大小、位置、颜色、透明度，就是说每一路的视频

水印都可以不相同，根据需要进行修改。

3、增加支持国标 10000 单端口接入流媒体服务器，url 命名规则为 /rtp/ssrc 的 16 进制转换为大写字符串，比

如 /rtp/DF13AC58，可以调用 getMediaList 函数获得单端口接入的国标流。

4、增加支持 ffmpeg 推 TS 流接入流媒体，url 命名规则和国标单端口接入一样。ffmpeg 推 TS 流命令如下：

```
ffmpeg -re -i d:\1.mp4 -vcodec copy -f rtp_mpegts rtp://127.0.0.1:10000
```

5、国标接入流方面优化，在接入的连接、ip、端口、ssrc 等等参数进行检测，最大程度上杜绝接入的视

频出现串流。

6、增加码流未到达事件通知，在发送代理拉流、国标接收请求成功后，由于源流没有到达会触发该事件。如

果发送请求失败不会触发该事件。

7、增加支持强制 i 帧输出功能，当播放器播放点播某路流时，能强制发第一帧 i 帧，确保视频画面秒开，开启

参数为 ForceSendingIframe=1 即可开启，默认关闭。

7.7. 2022-10-28 发布

1、优化 rtmp 拉流接入功能，已支持从 srs、zlm 拉取 rtmp

2、优化国标模块接入、国标输出，提高国标模块稳定性

3、录像模块支持 mp4、fmp4 两种录像格式，可在配置文件中配置。

4、优化 http-mp4 码流输出模块，提高稳定性，循环压测超过 50 万次码流输出稳定可靠。

5、强制发送 i 帧参数默认为开启，视频画面可以达到秒开

6、优化录像文件点播，录像回放更加流畅、音视频更加同步回放。

参数增加

#录像文件采用的文件格式 1 为 fmp4 格式，2 为 mp4 格式
videoFileFormat=1

#录像进度:当前录像文件已经录制了多少秒，总录像时长已经到多少秒：如果不需要通知该事件，就把 http://10.0.0.238:7088/index/hook/on_record_progress 删除掉

on_record_progress=http://192.168.0.103:4088/index/hook/on_record_progress

7.8. 2022-11-30 发布

- 1、增加 linux 平台的英伟达专业显卡转码功能，支持硬件解码、硬件缩放、硬件编码功能。实测两张 T4 卡能并发转码 80 路 1080p 的 h265 输出为 960x640 的 h264，硬件转码达到专业级别、商用级别、业界领先。
- 2、代理拉流接入增加支持 hls 代理拉流接入流媒体服务器。
- 3、代理拉流接入、国标接入可以指定视频转码分辨率。
- 4、增加可以指定是否某一路 H265 转码、H264 降分辨率。
- 5、新增支持视频转码时可以针对某一路指定不同分辨率。
- 6、增加指定某一路进行 hls 切片、某一路不执行 hls 切片。
- 7、http 函数增加服务器重启功能。
- 8、http 函数增加服务器安全退出功能。
- 9、http 函数增加设置配置服务器任意一个参数值并立即起效。

7.9. 2023-03-29 发布

- 1、优化接收线程池、发送线程池模块，极大提高网络性能，降低 cpu 占用率
 - 2、支持 rtsp、rtmp、国标 gb28181 推流接入只有音频流
 - 3、支持 rtsp、rtmp、国标 gb28181 输出只有音频流
 - 4、支持国标 gb28181 对讲（已实现 udp 方式下一版支持 tcp）
 - 5、国标接入支持过滤视频流只保留音频用于设备之间对讲
 - 6、代理拉流接入支持过滤视频流只保留音频把摄像头当作一个麦克风使用，可以解决国标对讲中的音源输入问题
 - 7、支持各种流接入事件通知，为接入鉴权提供判断条件
 - 8、支持各种流播放事件通知，为播放鉴权提供判断条件
 - 9、国标接入 openRtpServer 支持接入 PS、ES 流，也可以扩展为私有格式码流接入、加密视频接入
 - 10、国标发送 startSendRtp 支持 PS、ES 流发送，也可以扩展为私有格式码流发送、加密视频发送
 - 11、重新设计读取 mp4 录像文件机制，使录像回放更为流畅
 - 12、修复高频率图片抓拍的少量的内存泄露问题
 - 13、修改配置文件中的参数默认把 g711a、g711u 转码为 aac
- 【注：如果不需要 g711 转码为 aac 的用户自行关闭】
- 注：ABLMediaServer 流媒体服务器各种网络协议
(rtsp、rtmp、flv、ws-flv、http-mp4、hls、国标)
流输出都经历过 100 万次高强度的连接、断开压力测试过。

7.10. 2023-05-30 发布

- 1、支持视频转码时设置原尺寸输出，转码出来的宽、高都设置为 -1 即可原尺寸转码，原尺寸转码模式下 linux

平台专业卡视频转码不再消耗 cpu 资源

- 2、转码分辨率新增加 720 x 576 分辨率，至今视频转码输出支持 -1 x -1、1920 x 1080、1280 x 720、960 x 640、800 x 480、720 x 576。（-1 x -1 为 原尺寸）
- 3、新增 TCP、UDP 方式国标上级、国标下级相互发送接收码流即国标接收同时支持国标发送、国标发送同时支持国标接收达到全双工，实现以国标协议达到上下级之间共享、对讲
- 4、rtsp 协议输出支持 udp。vlc、ffplay 播放不用再设置 tcp 方式
- 5、rtsp 接入默认开启 rtcp 保活包，完美对接海康视频管理平台
- 6、修改配置 h265CutType=2，hls 协议 h265 切片为苹果的 fmp4
- 7、代理拉流增加一个过滤掉音频的参数，不需要音频或敏感声音不让听到可以开启这个音频过滤参数
- 8、国标接收、国标发送都增加屏蔽视频、屏蔽音频可选参数
- 9、http-flv、ws-flv 增加仅有音频流（没有视频）输出
- 10、优化 tcp 方式的国标接收，码流输出达到极度程度流畅
- 11、增加 listServerPort 函数列举服务器所占用的所有端口
- 12、优化 http-mp4 协议输出，使视频画面基本达到秒开程度
- 13、http 请求函数先执行密码校验，能阻断绝大部分非法连接
- 14、修复 rtmp 流输出计算双通道音频时间戳增量值
- 15、修复 rtsp 流输出中 sdp 信息中的双通道音频参数值
- 16、修复流媒体服务器运行界面上点击鼠标卡住服务运行问题

7.11. 2023-07-28 发布

- 1、新增加支持国标 tcp 主动拉流接入服务器
- 2、新增加支持国标 tcp 被动连接发送媒体流
- 3、新增加支持国标对讲 rtp+aac
- 4、把 ffmpeg 升级到 6.0 版本，抓拍出来图片质量更高
- 5、修复 boost:asio 网络库一个线程安全问题，网络库更坚固

7.12. 2024-01-05 发布

- 1、新增录像文件 ts 格式，录像回放新增加 hls 格式画面秒开
- 2、新增支持 webrtc 播放，画面秒开、延时低、不积累延时
- 3、新增 JTT1078 协议码流接入，也支持 JTT1078 码流共享输出
- 4、新增代理拉流接入选择本地 mp4 文件接入流媒体服务器
- 5、新增支持浏览器里录制语音以 websocket 协议接入服务器
- 6、新增音频 mp3 推流接入，rtsp、rtmp、flv 纯 mp3 音频输出
- 7、新增 10000 单端口 tcp 方式 GB28181 国标码流接入服务器
- 8、播放 http-flv、ws-flv 时可以增加静音输出视频画面秒开
- 9、修复底层 boost:asio 一个线程安全，网络库已无比坚固可靠
- 10、windows 平台把 ffmpeg 升级到 6.0 版本
- 11、全平台把 faac 升级到 1.30 版本

7.13. 2024-05-06 发布

- 1、新增 ffmpeg 代理拉流接入 http-flv、http-mp4、http-hls
- 2、新增查询录像返回多段文件时立即产生可播放的媒体源
- 3、支持多个录像文件连续、无缝正常速度、倍速、拖拽播放
- 4、支持多个录像文件给国标上级平台推送、支持倍速、拖拽
- 5、为了方便 wvp 对接把 ABL 所有消息通知都修改为短连接方式
- 6、抓拍函数新增参数直接指定是返回 url、还是返回 jpeg 图片