VLC 网页播放器(PageHi VLC Web Player)—

网页中低延迟、多路原生播放 RTSP 首选(2024 年版)

开发者手册

索引

撰写人	WangZuoHuai
版本	V2.2.15.1
公司	成都佐罗软件有限公司
日期	2024 年 05 月
销售热线电话	**4006831589 / 18081958957**
技术支持微信号	**ZorroSoft**
技术交流 QQ 群	**23126938**

一、概述

在遍地都是摄像头的今天，往往需要在各种信息化、数字化、可视化 B/S 系统中集成 RTSP(实时视频流)播放等功能，海康威视、浙江大华、深圳华为等厂家摄像头或录像机等设备一般也都遵循安防行业标准，支持国际标准的视频流传输协议 RTSP 输出，IE 等浏览器通过 ActiveX 控件或 NPAPI 插件方式也能正常播放。不幸的是 Chrome、Edge、Firefox 等新一代浏览器从 2015 年开始不再支持 NPAPI 插件加载运行，直接导致 RTSP 视频流从此无法在高版本浏览器网页中原生播放。对于绝大部分没有影视频处理经验的前、后端工程师来说是一个非常棘手的问题，专业性强，技术门槛高，而对做 B/S 系统集成的大多数公司来说，为了这部分功能单独招聘专职音视频研发人员来负责的话，成本高昂不说，还未必做的好。

既然当前浏览器已都不支持原生播放 RTSP 流，为了能播 RTSP 流，市场上不少公司也各显神通出了多种商用及开源方案，不过总的说来大多是在后端先转码再转流给前端播放的方案，这也是号称无插件的技术方案。对浏览器可直接硬件加速播放的 H.264 编码视频流，现在一般是在服务器端将 RTSP 流转为 WebRTC 流，前端接收后经过一定处理通过 Vedio 标签播放，这样服务器的压力大；对于浏览器不能支持加速播放的其它编码视频流，要么在后端先转流转码为 H.264 编码后按前述方案播放，要么后端通过 Web Socket 转流到前端，前端再通过对应编码的 WASM 播放器利用 CPU 软解码播放。转流到前端 WASM 转码播放时，即使配置了性能不错的电脑，受限于 WASM 的固有缺陷，比如多线程支持差、能使用的内存大小始终受限，无法充分利用终端电脑显卡的硬件加速能力(GPU)等，这就导致同时播放多路或高清 RTSP 流时也会比较吃力，而且大量占用终端电脑的 CPU 和内存资源，其它操作基本无法正常进行，对音视频格式的兼容能力也很有限，不大适用于复杂的环境。

虽然无插件播放方案能够播放出画面，但是往往延迟高，高分辨率视频流基本上需要数秒之久，在一些对延迟敏感的场合客户需要的是毫秒级延迟，显然无插件技术方案是无法满足的；而且首屏画面显示很慢，这就导致切换播放源时迟迟看不到有画面出来，播放体验差；况且无插件技术方案，需要在后端或前端持续运行高负荷运转的视频转码转流服务，虽然也有部分采用 WebSocket 直接从摄像头拉流播放的方案，不过需要设备支持才行，兼容性不够。如果摄像头路数多、高清甚至是 4K 视频或需要在线播放的终端比较多，服务器的压力就会很大，播放卡顿、花屏、黑屏、断播等现象就会时常出现，，要解决这些现象，最好搭配高端专用硬件解码器配合使用，相关硬件、软件的投入和持续不断的带宽占用，成本高，让客户难以接受。现在越来越多的客户追求高大上的视频播放效果，采用高清摄像头的越来越多，而播放显示器 1080P 已是低配，2K 甚至 4K 大屏正在成为主流之选。这种无插件技术方案，在中高配的屏幕上如果只能播放出慢如蜗牛的画面，想不让客户吐槽实在是太难了。

无插件方案网上宣传比较多的是国产的 EasyPlayer.js，宣传工具版(编译好的程序)免费，商用集成其实也是要收费的。EasyPlayer.js 在多路同时低延迟播放高清视频流时还是比较吃力的，首屏画面得几秒才能出来，延迟始终比较高，至少 1 秒以上，这里有其官方人员在 QQ 客服群里的对话记录为证：

如果是高清视频流如 2K、4K 屏的延迟就更高了，因为转码转流的技术路线不管如何优化，肯定还是无法和原生直接播放相提并论的，性能就是其最大瓶颈。在安防和直播等行业中，尤其是直播讲究的就是低延时，越低越好。一般来说，视频监控的目的就是尽早发现异常和风险，好及时进行处理，有时早一秒发现问题，就能避免重大事故发生，规避可能的人员伤亡及高价值设备的报废，因此通过技术方案做到低延迟从而尽可能避免巨大的经济损失，具有重要的社会意义。

基于当前市场对网页中多路低延迟原生播放 RTSP 的迫切需要，佐罗软件的 VLC 网页播放器 PageHi VLC Web Player 应运而生，基于跨浏览器的原生小程序系统-PluginOK 中间件开发，通过借助 PluginOK 中间件提供的 ActiveX 控件可内嵌网页运行的独家专利技术(专利号：**ZL 2019 1 1323165.1**)，在 Chrome 等现代浏览器主流版本中完全模拟实现了原来 ActiveX 控件和 NPAPI 插件的播放效果，底层调用 VLC(是一款自由、开源的跨平台多媒体播放器及框架，可播放大多数多媒体文件，以及 DVD、音频 CD、VCD 及各类流媒体协议)桌面客户端的 ActiveX 控件实现在网页中低延迟直接播放海康、大华、华为等监控设备的 RTSP 流，由于实际调用的是 VLC 本地原生播放控件，因此可充分利用本机硬件加速能力实现高效硬解码播放多路、高清和 4K 视频，最多可支持 25 路同时播放(具体能播多少路，取决于终端电脑硬件性能、网络带宽和 RTSP 流分辨率，多路播放时建议分屏小窗口都用子码流)，最低可用在 Chrome 41、Firefox 50、Edge 80(Chromium 内核)、360 极速/安全、IE、Opera、Electron、Vivaldi、Brave、QQ、华为、联想等浏览器，

也兼容运行于这些浏览器的最新版本。

总的来说，采用佐罗软件 VLC 网页播放器具有如下优点：

1、在支持所有主流版本浏览器的前提下，是当前市场中可实现低延迟(200-500 毫秒)稳定播放的不二之选技术方案，而且首屏画面显示很快；

2、支持多路播放、动态批量切换播放源，支持定时抓图和指定时常录像保存、支持添加水印和字幕显示、支持全屏播放、局部放大播放、旋转播放、单帧播放、慢播、快播、双击全屏播放及热键控制播放、指定显卡或双显卡加速播放等，尤其是高分辨率和 H.265 编码视频播放体验比其它转码转流方案好；

3、底层播放采用的是开放源代码的 VLC ActiveX 控件，播放器引擎后续更新支持有保障，VLC 对播放流和视频格式的兼容能力在同类产品中无与伦比，同时支持 H.264 和 H.265 编码无压力；

4、VLC 网页播放器额外付费后可获得其源代码进行定制开发，实现源代码级别的自主可控，也可基于其它的视频播放控件委托开发出类似的网页播放器；

5、支持海康、大华、华为、宇视、科达等各厂家的摄像头及硬盘录像机等视频监控设备，只要能提供标准的 RTSP 实时流或回放流，都可以正常稳定播放，也可以播放远程或本地的 MP4 等多种格式视频；

6、此方案简单高效，在原有技术路线下实现平滑升级兼容各种浏览器，降低了方案大改造带来的技术路线和延期交付风险，可节省大量研发成本；

7、提供了丰富的前端开发接口，前端集成简单，支持纯 JS、VUE2 及 VUE3 等主流框架，无需视频专业知识，无需关心系统和浏览器是 32 位还是 64 位，一套代码即可全兼容；

8、移动端可选择基于开放源代码的 VLC 手机 APP 工程定制出自己的版本来使用，苹果和安卓系统都有对应版本，投入少见效快；也可以搭配转码转流方案和本方案一起使用，实现全平台和全终端的兼容播放。

终端电脑采用 VLC 网页播放器时需部署 PluginOK 中间件并搭配 VLC 网页播放器小程序包，还需安装 VLC 的桌面播放器并注册其 ActiveX 控件，因此为简化部署，可采用 VLC 绿色版程序包（3.0.*版本绿色版，如果运行在 Windows 7 系统中，记得先删除下 VLC\plugins\codec\ libd3d11va_plugin.dll 文件）解压文件放入 VLC 网页播放器目录后，再将这些程序文件按目录结构统一做成 MSI 或 EXE 安装包放到 B/S 服务器上，提供给用户下载安装或实施工程师统一部署。一般来说，需要看视频监控的地方基本上集中于监控室，电脑数量不是很多，只需安装一个程序包即可快速完成部署，还是很简单的。之后的升级和配置可借助 PluginOK 中间件的在线升级机制，还可彻底解决传统客户端软件升级维护的难题，一箭双雕。

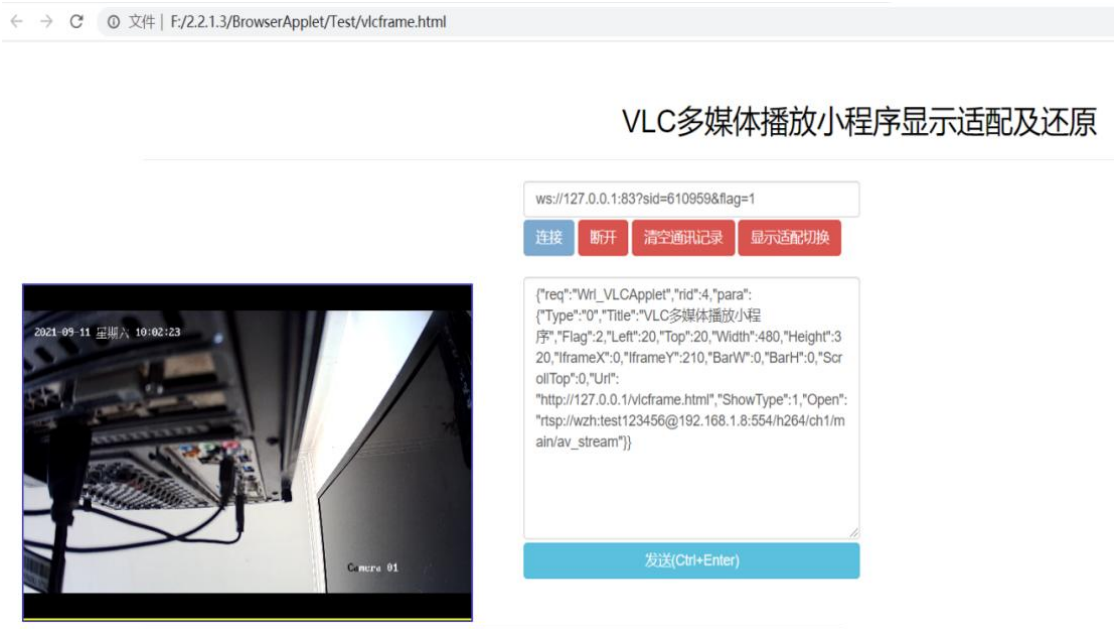
一个好的 RTSP 网页播放器，首先是要能做到持续稳定播放多路视频，还需同时支持 H.264、H.265 编码及其它编码格式，可以兼容 RTSP、RTMP、HLS 等流协议播放，最核心的是要做到低延迟、切换画面快，最好能支持高清甚至 4K 视频流畅播放。另外就是对当前主流版本的浏览器兼容能力要强，多路播放时分屏样式多，开发接口丰富并可定制，如果还能做到开源或底层采用免费的主流播放引擎实现，那就最好不过了，毕竟开源在商业领域的应用越来越多，是大趋势。从系统集成商的角度来说，尽可能采用成熟开源工程来使用意味着有更多的自主可控机会来降低整个系统的实施风险。

二、在线体验

请先下载网络高级版的 RTSP 网页播放器安装包进行安装，一般来说，此安装包已经默认内置了 2 个不同 PID 的 VLC 网页播放器的程序包，分别演示不同播放配置情况下的播放效果，需要注意的是，本程序运行时依赖 VLC 桌面播放器，启动播放时如未安装，会自动打开 VLC 的官方网站下载页面，64 位系统建议下载 64 位版本的 VLC 软件进行安装，在实际给客户实施时，可把 VLC 的绿色版程序加入本中间件程序做一个统一的包进行安装。然后打开 RTSP 网页播放器在线演示进行体验，或者双击打开程序安装后子目录 Test 下的 VlcWebPlayer.html 网页，点击连接后再点击发送，即可尝试启动 VLC 网页播放器体验。如提示还未授权，请在测试网页中点击连接后复制粘贴以下内容：{"req":"Wrl_Version", "rid":1,"para":{"Mac" : 0,"More" : 0,"Config" : 0,"Router" : 0,"IP" : 0}} 到输入框中后点击执行，以便获取到测试电脑的 UniID，如下图所示：

如有需要，在启动播放发送的命令中，可以通过修改 ShowType 值(分屏风格)，或修改 Open 参数或 Web 参数，就是您需要播放的 RTSP 流地址，必要时请先进行 UrlEncode 编码。然后加上贵公司名称和联系人姓名及电话，再通过加微信 ZorroSoft 或扫描以下企业微信提交给客服人员，申请开通试用授权，一般很快会获得批准。

在线体验播放效果如下图所示：



如果启动播放后没有画面，并且浏览器自动打开了 VLC 的官方网站，就代表你的电脑上还未安装 VLC 桌面程序，请先从 VLC 官方网站下载对应版本并安装，推荐安装 VLC 最新版 3.0.20，经过测试验证，XP 系统中建议使用 2.2.8 版，最高可用 3.0.3 版本来实现网页中播放，具体请根据自己测试的结果来确定 VLC 的使用版本，毕竟不同版本的 VLC 对视频流或视频文件的兼容能力不一样。

VLC 网页播放器最低可支持在 Windows XP 系统中使用，建议在 Windows 10、11 及以上 X86 和 X64 系统使用，也支持在 Windows Server 2008 及以上版本服务器系统中使用。对于使用 ARM 芯片的电脑，如果安装的是 Windows 系统，一般也是可以兼容运行的。

本软件对浏览器的兼容性如下：

1、IE 8 及以上版本；
2、Chrome 41 及以上版本；
3、FireFox 50 及以上版本；
4、Edge(Chrome 内核) 80 及以上版本；
5、Opera 36 及以上版本；
6、Brave 浏览器；
7、Vivaldi 浏览器；
8、Electron 桌面程序；
9、360 极速浏览器(X) 9.5 及以上版本；
10、360 安全及企业安全浏览器；
11、QQ 浏览器 10 及以上版本；
12、搜狗浏览器；
13、华为浏览器；
14、奇安信浏览器；
15、海泰红莲花浏览器；
16、微信网页窗口；
17、联想浏览器...

如您另有特殊需求，也可以付费定制支持需要支持的专有浏览器适配版本。

三、个性化配置

找到 PluginOK 中间件程序的运行目录，其子目录下的 VLC 网页播放器配置文件 Plugins\ VLCWebPlayer \Config.json，对应的是通过 WS 连接中间件发送指令 Wrl_VLCWebPlayer (如果是用 Wrl_VLCApplet 指令启动播放器，那么这个对应的配置文件路径是 Plugins\90FC7E0E-0D2F-4C38-9875-B06407CE4556\Config.json)，可用记事本打开编辑，主要内容如下：{"COM":
"vlc/axvlc.dll,VlcOcx.dll,X64/VlcOcx.dll,VideoProxyPlayer.exe,
X64/VideoProxyPlayer.exe","Caching":300,"RTSPTCP":1,"ToolBar":1,"FillWnd":0,"Aut
oLoop":1,"SelfGPU":1,"Mute":0,"BlockFlag":7,"PercentTime":300,"ErrTryCount":3,"Er
rTrySecond":6, "PlayList":0, "OverTime":15,"AsynPlay":1,"LogFlag":7,"ClockSync":-1,
"DisableDBFull":0,"Avcodec":"none","AudioOut":"any","BackColor":"#000000","Stat
us":{"Text":"准备播放中，请等待","Err":"播放时出现错误
","Color":"#00FF7F","Opacity":255,"Size":16,"Name":"宋体"},"PORT": 930}
其中 COM 项是为 VLC 绿色版和中间件程序一起分发时需要自动注册的组件准备的，如果 VLC 绿色版程序单独放到子目录 vlc 了，那么 axvlc.dll 的路径也对应修改为 vlc/ axvlc.dll。PORT 设置播放器的 Web Socket 服务侦听默认端口，前端在请求连接到播放器端口时，不能写死这个值，因为可能启动多个实例，不同实例会采用不一样的端口进行侦听。前端在连接到中间件端口启动播放器的过程中，收到以下的 JSON 包中解析 Port 值，就是实际侦听端口。
{"event":"Wrl_AppletOK","aid":4,"rid":4,"data":{"SID":"221903","PID":"90FC7E0E-0D
2F-4C38-9875-B06407CE4556","Port":930}}
其中 aid 为当前播放器的实例 ID，前端可通过此 ID 对其进行各种控制，比如显示、隐藏、放大缩小播放窗口、滚动等等。
Caching 设置网络缓存时间，控制播放延迟的关键配置，单位毫秒，不建议设置太低，尤其是您的电脑配置或网络状态不太好的情况下，如遇到播放画面停滞，就需要适当调大这个值比如 300 毫秒后保存再启动播放看效果，关于 VLC 播放 RTSP 延迟问题，建议看看这个文章：https://developer.aliyun.com/article/243646。在甲方客户现场实际实施时，可进行针对性的测试，找到一个可正常播放又能保证尽量低延迟的数值。RTSPTCP 配置 Live555 流传输采用 HTTP 还是 TCP 方式，1 为 TCP 方式，0 为 HTTP 方式，如遇无法播放情况，请先检查播放源是否正常，另外就是调整此参数进行测试。关于检查播放源是否正常或调优播放效果，常见做法就是直接启动 VLC 桌面程序进行播放测试。以上这些播放参数只是默认值，也可在启动播放器后，再调用对应接口设置其它值来控制播放行为，通过接口调用的参数只在当时生效，不会修改默认配置中的数值。ToolBar 设置播放画面中是否显示控制播放的工具栏，FillWnd 设置播放画面是否充满整个窗口区(设置为 1 时充满窗口，这可能会导致画面拉伸)，AutoLoop 配置播放结束时是否自动再次播放，SelfGPU 配置播放器是否采用高性能 GPU 播放，Status 是启动播放时的状态提示文字信息，可设置字体大小、字体名称、颜色和透明度。Mute 设置是否初始播放时静音，BackColor 为播放窗口默认背景色，ErrTryCount 为网络出现故障时重试播放次数,，OverTime 连接超时秒数，AsynPlay 异步播放设置 1 避免

界面卡顿，LogFlag 设置 VLC 播放输出更多信息，ErrTrySecond 收到播放故障后延迟多少秒重试，PercentTime 为推送播放进度最短周期，PlayList 配置是否支持播放列表，默认不支持，ClockSync 配置是否启用音频和视频的时钟同步，-1 默认行为，0 代表不同步，1 代表同步.DisableDBFull 设置是否禁用双击全屏功能，默认不禁用，Avcodec 配置硬件加速类型，可指定的其它值 any、dxva2、d3d11va，不设置时默认用 any； AudioOut 配置音频输出模块名称，可指定 any、mmdevice、directsound、waveout、amem、afile、NDI、adummy、none 等值，默认用 any。BlockFlag 为播放器转发 VLC 控件的事件通知消息过滤掩码，由于几个消息发送的比较多，大多前端不需要，所以通过设置阻止发送不需要的通知，详细说明请参考这里：信息屏蔽掩码

在 VLC 桌面客户端主界面，点击媒体菜单->打开网络串流，弹出以下画面：

输入 RTSP 流地址，并点击显示更多选项，修改网络缓存时间的数值后确认即可播放，测试播放延迟时，请注意桌面端配置和 VLC 网页播放器配置文件中保持一致，这样才方便对比两个之间的播放效果是否一致 。

在工具菜单中点击偏好设置，切换到如下界面，配置 Live555 流传输方式：

只要在 VLC 桌面客户端验证你的 RTSP 流能够稳定低延迟播放出来，那么在 VLC 网页播放器中播放也就没有问题，需要做的就是配置好播放参数。

如果播放性能不佳，修改如下硬件加速播放选项测试看看：

四、开发集成

在测试网页中播放正常的情况下，可以尝试把其中的播放代码，迁移到自己的前端网页中。本身对前端并无特殊要求，也支持 VUE 等常用前端框架，只需要 JS 脚本处理即可，通信采用 HTML5 中的国际标准技术 Web Socket，收发数据包采用 JSON 打包。由于 PluginOK 中间件已经抽象处理了各种浏览器的兼容问题，所以前端调用起来就非常方便了，除了在 IE 浏览器中需要调用 PluginOK 中间件

提供的 Web Socket 连接 ActiveX 控件之外，其它都是一样的。测试网页中需要手工点击连接，编辑启动参数后，再点击发送才可启动播放窗口，而在实际的 B/S 项目中，是不可能让用户这样操作的，所以前端需要把点击连接和发送这两个手工执行的过程，通过编写 JS 脚本代码实现自动执行，从而可在网页加载后完成自动播放效果，集成后的 VUE 版在线演示网页请访问这里 RTSP 流播放在线演示。具体可查看测试网页中开源的 JS 代码和 HTML 网页中的写法。集成大体上有以下几步：

1、编写网页，嵌入VLC网页播放器需要的代码：

前端集成采用纯JS脚本时，请主要参考：https://gitcode.net/zorrosoft/pluginok/-/tree/master/Demo/VlcJS，需要嵌入用到的JS脚本程序是完全开源的，如果是VUE2的框架，请参考 https://gitcode.net/zorrosoft/pluginok/-/tree/master/Demo/VlcVue2，VUE3框架的请参考 https://gitcode.net/zorrosoft/pluginok/-/tree/master/Demo/VlcVue3。如果只是简单测试启动，可以参考在线体验网络版网页：http://local.zorrosoft.com/vlcWebPlayer.html 和http://local.zorrosoft.com/vlcwebfull.html 前者实现网页中局部加载播放器窗口，后者实现自动匹配网页窗口大小加载。如下以Wrl_VLCWebPlayer指令举例：

在HTML网页中加入PluginOK的脚本

<script src="PluginOK/base.js"></script> 此脚本主要实现Web Socket

<script src="PluginOK/wrl.js"></script> 此脚本中有一些测试网页的元素处理，可以自行删除。主要实现收发Web Socket包并进行对应处理。

如还需兼容IE浏览器使用，网络版测试网页中 <object ID="WrlWS" CLASSID="CLSID:C0971B90-4513-4E2D-A0B6-15B915FE748A" width="0" height="0"></object>

的CLSID需要替换为：21ADE2E6-B4DD-4F3E-8BD5-9DDAD1785F3A，单机版不动。

前端集成可参考以上测试网页进行。

播放窗口选中状态时单击或按字母F即可全屏显示，全屏显示状态通过ESC、Windows按键、ESC或字母F取消，全屏切换会发送请求为VLC_FullScreen的JSON包通知。

2、请求启动播放器的接口JSON包详细参数说明：

Type为浏览器类型，传0自动判断(前提是当前浏览器已启动并显示在最前端，Flag指定当前页加载时必须是0) 可强制指定浏览器类型Type(1代表IE 2代表Chrome 4代表Firefox 8代表Opera 16代表Edge(Chromium内核) 20代表Electron 32代表360极速浏览器 33代表360安全浏览器 34代表360企业安全浏览器 50代表QQ浏览器 60代表搜狗浏览器)

Title：网页标题中的关键词

Url：加载小程序所在的网页实际地址，在网页中指定小程序的显示位置和大小，分屏多窗口播放地址和选项等，不建议使用了，建议改用Web参数

Flag掩码标记：1指定新标签加载(1和16都不指定时为当前页加载) 2小程序显示窗口边框 4不自动裁剪越界窗口 8自动适配网页高度和宽度显示 64启用Web参数 128防截屏 256强制显示到副屏 512允许同一网页加载多实例

Web：播放配置(新增方式)，可代替Url使用，Flag值+64使用此配置，此命令中必须指定Left、Top、Width、Height的值

Version播放小程序版本，0在播放小程序中播放，1在独立进程中播放
ShowType 播放窗口分屏类型，默认1只显示一个播放窗口，支持1到31的数值，多达31种分屏，注意此值不代表分屏个数
IframeX和IframeY分别为iframe嵌套的横竖偏移修正坐标
BarW和BarH分别是网页右侧和底部预留区域，ScrollTop为顶部滚动预留高度
小程序实际显示首先会基于Web或Url中指定的坐标和大小，再根据IframeX、IframeY、BarW、BarH设定的值做修正
Option：播放参数，多个参数中间用空格区分
:rtsp-tcp 指定TCP连接播放
:network-caching=300 指定网络缓存时间，毫秒为单位，如果是播放文件时，设置:file-caching=300
:start-time=3.000 指定起始时间，3秒，小数点后面是毫秒，只支持回放及文件视频
:stop-time=300.000 指定停止时间，300秒，小数点后面是毫秒，只支持回放及文件视频
LogFlag=1 指定VLC错误日志输出，默认1
ClockSync=-1 指定音频是否与视频同步，默认-1播放引擎默认值
AudioOut=any 指定音频输出类型 可选mmdevice、directsound、waveout、amem、afile、NDI、adummy、none
Avcodec=any 指定硬件加速类型，none不用硬件加速，可指定dxva2、d3d11va、any，any时让播放引擎自己选合适的
Transform=none 指定VLC播放画面是否旋转，除默认none不旋转之外，设置支持90、180、270、hflip、vflip、transpose、antitranspose
Open ：启动后第一个窗口自动播放的流地址或本地多媒体文件路径，斜杠\替换成/再传，也可以在Web中指定，如Open中的密码包含+等一些特殊字符，需要改到Web中指定
注意：Open、Url、Web中如果有特殊字符= & 双引号或中文等，需要用URL编码处理后传递 如非本地全路径，默认使用中间件程序Data子目录作为根目录使用。
请求举例如下：
{"req":"Wrl_VLCWebPlayer","rid":4,"para":{"Type":"0","Title":"VLC Applet","Flag":66,"Left":20,"Top":230,"Width":480,"Height":320,"IframeX":0,"IframeY":0,"BarW":0,"BarH":0,"ScrollTop":0,"Version":0,"ShowType":4,"Open":"https://vjs.zencdn.net/v/oceans.mp4","Web":[{"ID":2,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Option":":rtsp-tcp"},{"ID":3,"Uri":"http://www.zorrosoft.com/Files/PluginOKBrowserApplet.mp4","Option":":file-caching=300"},{"ID":4,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/sub/av_stream","Option":":network-caching=200"}]}}

需要注意的是，在需要继续播放的情况下，到中间件的WS连接是不能释放的，否则会导致播放窗口消失。

由于依赖VLC桌面器，所以可以提前调用以下接口获得VLC等播放器的安装情况，如未安装，可先提示下载安装
如果遇到部分流播放时崩溃，请修改以下请求中的指令参数Fix为1调用一次

{"req":"Wrl_PlayerInfo","rid":1,"para":{"Fix":0}} 获得当前电脑VLC桌面播放器安装的情况，返回值
{"ret":0,"rid":1,"req":"Wrl_PlayerInfo","data":{"Info":[{"Type":1,"X64":1,"Ver":"3.0.17.4","Path":"C:/Program Files/VideoLAN/VLC/vlc.exe"}]}}
Type 1 代表VLC X64 代表是否为64位版本 Ver 代表文件版本信息 Path 安装路径

3、前端脚本处理
启动后会前后收到三个JSON数据包
A、{"ret":0,"rid":1,"data":{"ID":2}} 代表小程序实例化开始，返回的ID是实例化ID，可通过这个ID对其进行相关控制
B、
{"event":"Wrl_Listen","aid":2,"data":{"SID":"123","PID":"VLCWebPlayer","port":935}}
代表小程序WS侦听服务就绪。返回的侦听端口，可再建立一个Web Socket连接后，调用小程序自身实现的相关功能，比如重新指定播放源，暂停播放，抓图等。
C、
{"event":"Wrl_AppletOK","aid":2,"rid":4,"data":{"SID":"123","PID":"VLCWebPlayer","Port":935}}
代表小程序创建成功，返回ID为当前小程序运行ID，通过此ID，可执行Wrl_AppletControl、Wrl_AppletScroll、Wrl_AppletResize等命令。举例如下：
1)、请求控制VLC网页播放器：
当前端不再需要小程序时可指定关闭，或者显示/隐藏及全屏显示等
ID为启动小程序时返回的ID值，Code代表控制类型掩码：1正常关闭 128强制迅速关闭 2全屏显示 4自动隐藏 8还原显示 16自动适配网页高度和宽度显示模式切换 32强制隐藏。其中全屏显示2，可直接通过热键ESC或取消,设置4和32隐藏后可通过8恢复显示
{"req":"Wrl_AppletControl","rid"::2,"para":{"ID":"1","Code":32}}

2)、请求滚动VLC网页播放器：
当前端截取到网页滚动通知时，需要调用此接口实现小程序窗口和网页的滚动联动
ID为启动小程序时返回JSON中的ID值
Code为滚动方向1是水平直，2是垂直，3是同时
Left为横向滚动条位置，Top为纵向滚动条位置
{"req":"Wrl_AppletScroll","rid":3,"para":{"ID":"1","Code":2,"Left":0,"Top":100}}

3)、请求改变VLC网页播放小程序显示位置或大小：
当前端网页显示区域缩放时，可动态修改小程序的显示位置或大小
ID为Wrl_VLCWebPlayer启动小程序时返回的ID值，Width和Height分别为新的宽度和高度
X和Y分别为新的显示位置，不指定时保持不变，指定时原设置的IframeX和IframeY失效
{"req":"Wrl_AppletResize","rid":4,"para":{"ID":1,"Width":500,"Height":600}}
或，同时修改小程序显示起始坐标

{"req":"Wrl_AppletResize","rid":5,"para":{"ID":1,"Width":500,"Height":600,"X":20,"Y":20}}

4)、请求设置网页预留右侧宽度和底部高度，滚动条信息、垂直滚动及水平滚动位置：
当小程序显示区域超过当前网页时，需去除滚动条的显示影响
ID为启动小程序时返回的ID值，BarW为预留右侧宽度 BarH为预留底部高度
Code 1代表有水平滚动条，2代表有垂直滚动条，3代表都有
ScrollTop垂直滚动条位置 ScrollTop水平滚动条位置
{"req":"Wrl_ScrollBar","rid":6,"para":{"ID":"1","Code":2,"BarW":0,"BarH":0,"ScrollTop":0,"ScrollLeft":0}}

5)、请求对小程序窗口做Alpha透明处理，便于前端临时显示覆盖到小程序窗口的菜单等：
ID为Wrl_VLCWebPlayer启动小程序时返回的ID值，Alpha为透明度百分比，1-100
{"req":"Wrl_AppletAlpha","rid":7,"para":{"ID":1,"Alpha":30}}

6)、请求对小程序窗口内容进行截图：
ID为启动小程序时返回JSON中的ID值，
File为指定截图文件保存路径或扩展名
Base64指定为1时代表返回BASE64编码的图像内容
{"req":"Wrl_AppletSnap","rid":10,"para":{"ID":1,"Base64":1,"File":".png"}}

7)、请求缩放内嵌网页小程序，用于浏览器网页按比例缩放时对应调用处理：
ID为启动小程序时返回JSON中的ID值，不指定Scale时，获取当前小程序所用的缩放百分比，一般和系统缩放比例一致
{"req":"Wrl_AppletScale","rid":11,"para":{"ID":1,"Scale":120}}

需要注意的是：如需要启动 2 个播放小程序实例时，每个 WS 连接最好只负责启动一个播放小程序，这样关闭连接时，不影响另外一个播放器的运行。

4、请求VLC网页播放器功能
在第3步中得到VLC网页播放器的实际侦听端口后，新建一个WS连接过去，成功后，就可以请求VlcOcx控件提供的功能，截获其事件通知了。可以请求的功能列表如下：
4.1、VlcOcx接口功能请求：
支持创建多个VlcOcx控件窗口同时播放多路实时视频，所以请求参数里需要指定窗口序号ID，从1开始，序号原则是从左向右开始编号，然后从上到下开始顺序编号，如遇右侧多层排列窗口时，直到右侧窗口序号排序完成。
1) 取指定ID播放窗口属性AutoLoop(自动循环)值，举例：
请求：{"req":"VLC_GetAutoLoop","rid":41,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":41,"ID":1,"data":{"Ret":0,"AutoLoop":0}} 和
{"ret":0,"rid":41,"ID":2,"data":{"Ret":0,"AutoLoop":0}}
ret为请求返回值，0正常，非零不正常，不正常时请取和ret同级的错误描述err,

Ret为调用VlcOcx控件对应函数返回值，0代表正常 下同

2) 设置指定ID控件窗口AutoLoop(自动循环)值，举例：
请求：
{"req":"VLC_PutAutoLoop","rid":42,"para":[{"ID":1,"AutoLoop":1},{"ID":2,"AutoLoop":0}]}
分别返回：{"ret":0,"rid":42,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":42,"ID":2,"data":{"Ret":0}}

3) 取指定ID控件窗口AutoPlay(自动播放)值，举例：
请求：{"req":"VLC_GetAutoPlay","rid":43,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":43,"ID":1,"data":{"Ret":0,"AutoPlay":0}} 和
{"ret":0,"rid":43,"ID":2,"data":{"Ret":0,"AutoPlay":0}}

4) 设置指定ID窗口AutoPlay(自动播放)值，举例：
请求：
{"req":"VLC_PutAutoPlay","rid":44,"para":[{"ID":1,"AutoPlay":1},{"ID":2,"AutoPlay":0}]}
分别返回：{"ret":0,"rid":44,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":44,"ID":2,"data":{"Ret":0}}

5) 取指定ID窗口StartTime(开始时间)值，举例：
请求：{"req":"VLC_GetStartTime","rid":45,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":45,"ID":1,"data":{"Ret":0,"StartTime":0}} 和
{"ret":0,"rid":45,"ID":2,"data":{"Ret":0,"StartTime":0}}

6) 设置指定ID窗口StartTime(开始时间)值，举例：
请求：
{"req":"VLC_PutStartTime","rid":46,"para":[{"ID":1,"StartTime":0},{"ID":2,"StartTime":0}]}
分别返回：{"ret":0,"rid":46,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":46,"ID":2,"data":{"Ret":0}}

7) 取指定ID窗口MRL(播放源)值，举例：
请求：{"req":"VLC_GetMRL","rid":47,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":47,"ID":1,"data":{"Ret":0,"MRL":""}} 和
{"ret":0,"rid":47,"ID":2,"data":{"Ret":0,"MRL":""}}

8) 设置指定ID窗口MRL(播放源)值，举例：
请求：
{"req":"VLC_PutMRL","rid":48,"para":[{"ID":1,"MRL":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream"},{"ID":2,"MRL":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream"}]}
分别返回：{"ret":0,"rid":48,"ID":1,"data":{"Ret":0}} 和

{"ret":0,"rid":48,"ID":2,"data":{"Ret":0}}

9) 取指定ID窗口Visible(可见性)，举例：
请求：{"req":"VLC_GetVisible","rid":49,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":49,"ID":1,"data":{"Ret":0,"Visible":1}} 和
{"ret":0,"rid":49,"ID":2,"data":{"Ret":0,"Visible":0}}

10) 设置指定ID窗口Visible(可见性)，举例：
请求：
{"req":"VLC_PutVisible","rid":48,"para":[{"ID":1,"Visible":1},{"ID":2,"Visible":0}]}
分别返回：{"ret":0,"rid":50,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":50,"ID":2,"data":{"Ret":0}}

11) 取指定ID窗口Volume(音量)，举例：
请求：{"req":"VLC_GetVolume","rid":51,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":51,"ID":1,"data":{"Ret":0,"Volume":0}} 和
{"ret":0,"rid":51,"ID":2,"data":{"Ret":0,"Volume":0}}

12) 设置指定ID窗口Volume(音量)，举例：
请求：
{"req":"VLC_PutVolume","rid":52,"para":[{"ID":1,"Volume":0},{"ID":2,"Volume":0}]}
分别返回：{"ret":0,"rid":52,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":52,"ID":2,"data":{"Ret":0}}

13) 取指定ID窗口BackColor(背景色)，举例：
请求：{"req":"VLC_GetBackColor","rid":53,"para":[{"ID":1},{"ID":2}]}
BackColor返回的是长整型(COLORREF)
分别返回：{"ret":0,"rid":53,"ID":1,"data":{"Ret":0,"BackColor":0}} 和
{"ret":0,"rid":53,"ID":2,"data":{"Ret":0,"BackColor":0}}

14) 设置指定ID窗口BackColor(背景色)，举例：
BackColor可用长整型(COLORREF)，也可以用#000000这样的颜色值
请求：
{"req":"VLC_PutBackColor","rid":54,"para":[{"ID":1,"BackColor":0},{"ID":2,"BackColor":"0"}]}
分别返回：{"ret":0,"rid":54,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":54,"ID":2,"data":{"Ret":0}}

15) 取指定ID窗口Toolbar(工具栏)显示与否，举例：
请求：{"req":"VLC_GetToolbar","rid":55,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":55,"ID":1,"data":{"Ret":0,"Toolbar":0}} 和
{"ret":0,"rid":55,"ID":2,"data":{"Ret":0,"Toolbar":0}}

16) 设置指定ID窗口Toolbar(工具栏)显示与否，举例：

请求：
{"req":"VLC_PutToolbar","rid":56,"para":[{"ID":1,"Toolbar":0},{"ID":2,"Toolbar":0}]}
分别返回：{"ret":0,"rid":56,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":56,"ID":2,"data":{"Ret":0}}

17) 取指定ID窗口推送信息屏蔽掩码，默认7，举例：
请求：{"req":"VLC_GetBlockFlag","rid":57,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":57,"ID":1,"data":{"Ret":0,"BlockFlag":7}} 和
{"ret":0,"rid":57,"ID":2,"data":{"Ret":0,"BlockFlag":7}}

18) 设置指定ID窗口推送信息屏蔽掩码，举例：
由于以下几种信息通知比较多，所以增加阻止通知设置，比如阻止鼠标移动和位置通知，就设置9
1阻止鼠标移动通知 2阻止时间变化通知 4阻止缓存通知 8阻止进度百分比通知
请求：
{"req":"VLC_PutBlockFlag","rid":58,"para":[{"ID":1,"BlockFlag":0},{"ID":2,"BlockFlag":0}]}
分别返回：{"ret":0,"rid":58,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":58,"ID":2,"data":{"Ret":0}}

19) 取指定ID窗口版本信息，举例：
请求：{"req":"VLC_GetVersionInfo","rid":59,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":59,"ID":1,"data":{"Ret":0,"VersionInfo":""}} 和
{"ret":0,"rid":59,"ID":2,"data":{"Ret":0,"VersionInfo":""}}

20) 取指定ID窗口多媒体描述信息，举例：
请求：{"req":"VLC_MediaDescGet","rid":60,"para":[{"ID":1},{"ID":2}]}
分别返回：
{"ret":0,"rid":60,"ID":1,"data":{"Ret":0,"Title":"","Artist":"","Genre":"","Copyright":"","Album":"","TrackNumber":"","Desc":"","Rating":"","Date":"","Setting":"","Url":"","Language":"","NowPlaying":"","Publisher":"","EncodedBy":"","ArtworkURL":"","TrackID":""}} 和
{"ret":0,"rid":60,"ID":2,"data":{"Ret":0,"Title":"","Artist":"","Genre":"","Copyright":"","Album":"","TrackNumber":"","Desc":"","Rating":"","Date":"","Setting":"","Url":"","Language":"","NowPlaying":"","Publisher":"","EncodedBy":"","ArtworkURL":"","TrackID":""}}

21) 取指定ID窗口音频信息，举例：
取时自动获得全部属性
请求：{"req":"VLC_AudioGet","rid":61,"para":[{"ID":1},{"ID":2}]}
分别返回：
{"ret":0,"rid":61,"ID":1,"data":{"Ret":0,"Mute":0,"Volume":100,"Track":"","TrackNumber":"","Channel":"","Desc":""}} 和
{"ret":0,"rid":61,"ID":2,"data":{"Ret":0,"Mute":"","Volume":"","Track":"","TrackNumb

er":"","Channel":"","Desc":""}}

22) 设置指定ID窗口音频信息，举例：
设置时不要求全部属性，只设置需要设置的
请求：
{"req":"VLC_AudioPut","rid":62,"para":[{"ID":1,"Mute":0,"Volume":100,"Track":"","TrackNumber":"","Channel":"","Desc":""},{"ID":2,"Mute":"","Volume":""}]}
分别返回：{"ret":0,"rid":62,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":62,"ID":2,"data":{"Ret":0}}

23) 取指定ID窗口播放信息，举例：
取时自动获得全部属性
请求：{"req":"VLC_GetPlay","rid":63,"para":[{"ID":1},{"ID":2}]}
分别返回：
{"ret":0,"rid":63,"ID":1,"data":{"Ret":0,"IsPlay":1,"ItemCount":3,"CurrentItem":1}}
和
{"ret":0,"rid":63,"ID":2,"data":{"Ret":0,"IsPlay":1,"ItemCount":5,"CurrentItem":2}}

24) 指定ID窗口添加播放内容，举例：
Uri需要进行UrlEncode编码，Option可不设置用缺省值 Uri如本地非全路径，默认使用中间件程序Data子目录作为根目录使用
请求：
{"req":"VLC_AddPlay","rid":64,"para":[{"ID":1,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"test","Option":":network-caching=300"},{"ID":2,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"test","Option":":rtsp-tcp"}]}
分别返回：{"ret":0,"rid":64,"ID":1,"data":{"Ret":0,"Item":1}} 和
{"ret":0,"rid":64,"ID":2,"data":{"Ret":0,"Item":2}}

25) 指定ID窗口播放(指定多媒体)，举例：
ItemId是序号，可缺省
请求：{"req":"VLC_Play","rid":65,"para":[{"ID":1,"ItemId":1},{"ID":2}]}
分别返回：{"ret":0,"rid":65,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":65,"ID":2,"data":{"Ret":0}}

26) 控制指定ID窗口播放器，举例：
Type可以是这些值：Pause(播放暂停),Continue(播放继续),Play(开始播放),Stop(停止播放),Next(播放下一个视频),Prev(播放上一个视频),Clear(清理播放列表),TogglePause(暂停状态切换)
请求：
{"req":"VLC_Control","rid":66,"para":[{"ID":1,"Type":"Pause"},{"ID":2,"Type":"Play"}]}
分别返回：{"ret":0,"rid":66,"ID":1,"data":{"Ret":0,"Status":4}} 和
{"ret":0,"rid":66,"ID":2,"data":{"Ret":0,"Status":3}}
返回的Status代表播放状态，定义参考libvlc_state_t

27) 指定ID窗口移除多媒体内容，举例：
ItemId是当前已存在的序号
请求：
{"req":"VLC_RemoveItem","rid":67,"para":[{"ID":1,"ItemId":1},{"ID":2,"ItemId":2}]}
分别返回：{"ret":0,"rid":67,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":67,"ID":2,"data":{"Ret":0}}

28) 指定ID窗口解析选项，举例：
请求：
{"req":"VLC_Parse","rid":68,"para":[{"ID":1,"Option":"","Timeout":0},{"ID":2,,"Option":"","Timeout":0}]}
分别返回：{"ret":0,"rid":68,"ID":1,"data":{"Ret":0,"Status":0}} 和
{"ret":0,"rid":68,"ID":2,"data":{"Ret":0,"Status":0}}

29) 取指定ID窗口SubTitle信息，举例：
NameID指定时获取描述信息
请求：
{"req":"VLC_GetSubTitle","rid":69,"para":[{"ID":1,"NameID":0},{"ID":2}]}
分别返回：
{"ret":0,"rid":69,"ID":1,"data":{"Ret":0,"Spu":1,"SpuNumber":0,"Desc":""}} 和
{"ret":0,"rid":69,"ID":2,"data":{"Ret":0,"Spu":1,"SpuNumber":0}}

30) 设置指定ID窗口SubTitleSpu，举例：
请求：
{"req":"VLC_PutSubTitle","rid":70,"para":[{"ID":1,"Spu":0},{"ID":2,"Spu":0}]}
分别返回：{"ret":0,"rid":70,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":70,"ID":2,"data":{"Ret":0}}

31) 取指定ID窗口视频信息，举例：
TrackID指定时获取描述信息
请求：{"req":"VLC_VideoGet","rid":71,"para":[{"ID":1,"TrackID":1},{"ID":2}]}
分别返回：
{"ret":0,"rid":71,"ID":1,"data":{"Ret":0,"FullScreent":0,"Width":0,"Height":0,"SubTitle":0,"Track":0,"TrackCount":0,"TeleText":0,"Scale":1,"AspectRatio":"","Crop":"","Desc":""}} 和
{"ret":0,"rid":71,"ID":2,"data":{"Ret":0,"Width":0,"Height":0,"SubTitle":0,"Track":0,"TrackCount":0,"TeleText":0,"Scale":100,"AspectRatio":"","Crop":""}}
AspectRatio为视频显示长宽比例 Crop为裁剪视频信息

32) 设置指定ID窗口视频信息，举例：
需要的才设置 FullScreen设置是否全屏显示，Scale比例 AspectRatio视频显示宽高尺寸 Crop裁剪信息 SubTitle子标题
请求：

{"req":"VLC_VideoPut","rid":72,"para":[{"ID":1,"FullScreen":1,"Scale":1,"AspectRatio":"","SubTitle":0,"TeleText":0,"Track":0},{"ID":2,"FullScreen":0}]}
分别返回：{"ret":0,"rid":72,"data":{"Ret":0}} 和
{"ret":0,"rid":72,"ID":2,"data":{"Ret":0}}
设置充满指定窗口大小显示：
{"req":"VLC_VideoPut","rid":72,"para":[{"ID":1,"Scale":0,"AspectRatio":"480:320"}]}

33) 切换指定ID窗口视频全屏，一次只能请求一个分屏窗口，举例：
请求：{"req":"VLC_VideoToggleFullscreen","rid":73,"para":[{"ID":1}]}
分别返回：{"ret":0,"rid":73,"ID":1,"data":{"Ret":0}}

34) 切换指定ID窗口视频图文，举例：
请求：
{"req":"VLC_VideoToggleTeletext","rid":74,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":74,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":74,"ID":2,"data":{"Ret":0}}

35) 取指定ID窗口字幕信息，举例：
Text指定时获取文字信息 Position指定时获取位置信息
请求：
{"req":"VLC_MarqueeGet","rid":75,"para":[{"ID":1,"Text":1,"Position":1},{"ID":2}]}
分别返回：
{"ret":0,"rid":75,"ID":1,"data":{"Ret":0,"Color":0,"Opacity":0,"Refresh":0,"Size":0,"Timeout":0,"X":0,"Y":"","Text":"","Position":""}} 和
{"ret":0,"rid":75,"ID":2,"data":{"Ret":0,"Color":0,"Opacity":0,"Refresh":0,"Size":0,"Timeout":0,"X":0,"Y":""}}

36) 设置指定ID窗口字幕信息，举例：
指定的参数，需要的才设置
Text 文字内容
Opacity 透明度 0-256 0完全透明 256完全不透明
Position 显示位置有center left right top bottom top-left top-right bottom-left bottom-right 不指定时由X和Y坐标决定
X 和 Y分别指定文字开始坐标Timeout 超时
Color 颜色 可用长整型(COLORREF)，也可以用#000000这样的颜色值
Size 字体大小
请求：
{"req":"VLC_MarqueePut","rid":76,"para":[{"ID":1,"Text":"Hello","Position":"TOP","Timeout":0,"Color":"16777215","Opacity":128,"Refresh":1,"Size":50,"X":50,"Y":50},{"ID":2,"Opacity":256}]}
分别返回：{"ret":0,"rid":76,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":76,"ID":2,"data":{"Ret":0}}

37) 控制指定ID窗口字幕，举例：

请求：
{"req":"VLC_MarqueeControl","rid":77,"para":[{"ID":1,"Enable":1},{"ID":2,"Enable":0}]}
分别返回：{"ret":0,"rid":77,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":77,"ID":2,"data":{"Ret":0}}

38) 指定ID窗口帧速控制工具，举例：
Mode为空时禁用，取值有blend bob discard linear mean x yadif yadif2x phosphor ivtc auto 请求：
{"req":"VLC_DeinterlaceControl","rid":78,"para":[{"ID":1,"Mode":"my_mode"},{"ID":2,"Mode":""}]}
分别返回：{"ret":0,"rid":78,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":78,"ID":2,"data":{"Ret":0}}

39) 动态改变分屏数量或播放内容
ShowType 指定分屏数量，如果不变就不设置
ForceDestroy VLC4.0且独立进程播放时默认1，其它时候默认0，指定0时切换播放不销毁原有控件
Play 重新指定播放内容：Uri中如果有特殊字符= & 双引号或中文等，需要用URL编码处理后传递 Uri如非本地全路径，默认使用中间件程序Data子目录作为根目录使用
请求范例1：{"req":"VLC_ChangePlay","rid":79,"para":{"ShowType":2}}
请求范例2：
{"req":"VLC_ChangePlay","rid":79,"para":{"Play":[{"ID":1,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"BrowserApplet1","Option":":rtsp-tcp"},{"ID":2,"Uri":"http://www.zorrosoft.com/Files/PluginOKBrowserApplet.mp4","Name":"BrowserApplet2","Option":":file-caching=300"}]}}
请求范例3：
{"req":"VLC_ChangePlay","rid":79,"para":{"ShowType":3, "ForceDestroy":0,
"Play":[{"ID":1,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"BrowserApplet1","Option":":rtsp-tcp :network-caching=300"},{"ID":2,
"Uri":"http://www.zorrosoft.com/Files/PluginOKBrowserApplet.mp4","Name":"BrowserApplet2","Option":":file-caching=400"},{"ID":3,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"BrowserApplet3","Option":":rtsp-tcp"}]}}
分别返回：{"ret":0,"rid":80,"data":{"Ret":0,"Play":[]}}

40) 设置当前焦点播放窗口
ID是当前播放窗口序号，ID是从1开始，需要确保不越界
{"req":"VLC_SetSelect","rid":80,"para":{"ID":2}}
返回：{"ret":0,"rid":80,"ID":2,"data":{"Ret":1}}

41) 获取当前焦点播放窗口序号
{"req":"VLC_GetSelect","rid":81,"para":{}}

返回：{"ret":0,"rid":81,"ID":2,"data":{"Ret":1}}
ID为当前播放焦点窗口

42) 指定ID窗口截图
Type 指定图像类型，默认4(PNG) 可指定1(BMP，不支持BASE64编码内容)、3(JPG)
Base64指定为1时代表返回BASE64编码的图像内容
AutoDel设置1 Base64为1时生效，自动删临时生成的图片文件，不删除设置0
Count截取图数量，默认截取一张，Delay指定延迟毫秒数开始，Interval多张时间隔毫秒数
FileName 指定截图文件名称,中文等需要先进行UrlEncode编码，可缺省使用默认值
CustomPath 指定截图保存路径，中文等需要先进行UrlEncode编码，不指定时由PathType决定
PathType 指定截图图片保存路径类型，0默认临时目录，1操作系统桌面 2当前登录用户我的图片目录 3中间件数据目录
多窗口播放情况下，只能指定某个窗口截图，不能同时指定多个窗口
请求：请求1：
{"req":"VLC_VideoSnapshot","rid":82,"para":[{"ID":1,"Type":3,"PathType":3,"Count":3,"Base64":1,"AutoDel":1,"Delay":1000,"Interval":200}]}
请求2：
{"req":"VLC_VideoSnapshot","rid":82,"para":[{"ID":1,"Type":3,"Count":3,"CustomPath":"D:/Zorro","FileName":"TestFileName"}]}
分别返回：
{"ret":0,"rid":82,"data":{"Ret":0,"Img":[{"Size":0,"Width":0,"Height":0,"File":""}]}} 和
{"ret":0,"rid":82,"ID":2,"data":{"Ret":0,,"Img":[{"Size":0,"Width":0,"Height":0,"File":""}]}}

43) 设置RTSP流网络方式，HTTP还是TCP
{"req":"VLC_SetRtspTcp","rid":83,"para":{"Tcp":1}}
返回：{"ret":0,"rid":83,"data":{"Ret":1}}

44) 获取RTSP流网络方式，HTTP还是TCP
{"req":"VLC_GetRtspTcp","rid":84,"para":{}}
返回：{"ret":0,"rid":84,"data":{"Tcp":1}}

45) 设置播放内容是否充满窗口
{"req":"VLC_PutFillWnd","rid":85,"para":[{"ID":1,"FillWnd":1},{"ID":2,"FillWnd":0}]}
分别返回：{"ret":0,"rid":85,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":85,"ID":2,"data":{"Ret":0}}

46) 获取播放内容是否充满窗口
{"req":"VLC_GetFillWnd","rid":86,"para":[{"ID":1},{"ID":2}]}
分别返回：{"ret":0,"rid":86,"ID":1,"data":{"Ret":0,"FillWnd":1}} 和
{"ret":0,"rid":86,"ID":2,"data":{"Ret":0,"FillWnd":0}}

47) 获取LOGO显示信息
{"req":"VLC_GetShowLogo","rid":87,"para":[{"ID":1},{"ID":2}]}
分别返回：
{"ret":0,"rid":87,"ID":1,"data":{"Ret":0,"File":"","Position":"","Delay":0,"Repeat":0,"Opacity":0,"X":0,"Y":0,"Enable":0}} 和 {"ret":0,"rid":87,"ID":2,"data":{"Ret":0}}

48) 设置水印显示
ID为窗口序号
File 水印文件路径，如有特殊符号请先进行UrlEncode，为空默认用小程序目录的Logo.png
Code 水印图片BASE64编码，可替代File
Position 显示位置有center left right top bottom top-left top-right bottom-left bottom-right，默认undefined未定义，显示位置有坐标X和Y控制
Delay 水印间隔图像时间为毫秒 0 - 60000 默认1000
Repeat 循环 水印动画的循环数量 -1继续(默认) 0关闭
Opacity 透明度 (数值介于 0(完全透明) 与 255(完全不透明，默认)
X 水印X坐标
Y 水印Y坐标
{"req":"VLC_PutLogoShow","rid":88,"para":[{"ID":1,"File":"VLC.png" ,"Delay":20,"Repeat":-1,"Opacity":128,"X":100,"Y":100},{"ID":2,"Position":" center ","Delay":100,"Repeat":-1,"Opacity":256,"X":200,"Y":0}]}
分别返回：{"ret":0,"rid":88,"ID":1,"data":{"Ret":0,"Enable":1}} 和
{"ret":0,"rid":88,"ID":2,"data":{"Ret":0,"Enable":0}}

49) LOGO显示控制
ID为窗口序号 Flag为1代表启用，0代表禁用
{"req":"VLC_LogoControl","rid":89,"para":[{"ID":1,"Flag":1},{"ID":2,"Flag":0}]}
分别返回：{"ret":0,"rid":89,"ID":1,"data":{"Ret":0,"Enable":1}} 和
{"ret":0,"rid":89,"ID":2,"data":{"Ret":0,"Enable":0}}

50) 对指定RTSP流录像到文件
ID为分屏窗口，此分屏窗口必须有可播放的视频
Url为RTSP流地址，不指定ID和Url时取当前焦点窗口流的
File为录像目标文件不指定时自动生成 Second为录制限时时间(秒)，0代表由前端指定停止
{"req":"VLC_RecordFile","rid":90,"para":{"ID":1,"File":"D:/Zorro/test.mp4","Second":30}} 或
{"req":"VLC_RecordFile","rid":90,"para":{"Url":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","File":"D:/Zorro/test.mp4","Second":30}}
返回：{"ret":0,"rid":90,"data":{"Ret":0,"PID":1}}

51) 停止录像
传入指定录像时返回的PID

{"req":"VLC_StopRecord","rid":91,"para":{"PID":1}}
File为录像文件路径，Size为录像文件大小
返回：{"ret":0,"rid":91,"data":{"Ret":0,"File":"D:/Zorro/test.mp4","Size":321}}

52) 获取输入源信息
Title为1支持获取标题更多信息，标题中支持获取指定Track的描述
Chapter为1支持获取章节更多信息 章节中可指定TitleID获取TrackCount，再指定ChapterID获得章节描述
{"req":"VLC_GetInputInfo","rid":92,"para":[{"ID":1,"Title":1},{"ID":2,"Chapter":1}]}
分别返回：
{"ret":0,"rid":92,"ID":1,"data":{"Ret":0,"InputInfo":{"Status":3,"HasVout":1,"Len":0.0,"Pos":0.0,"Time":9667.0,"Rate":1.0,"Fps":25.000,"Title":{"Count":0,"Track":0,"Desc":""},"Chapter":{}}}}
和
{"ret":0,"rid":92,"ID":2,"data":{"Ret":0,"InputInfo":{"Status":3,"HasVout":1,"Len":117078.0,"Pos":0.075,"Time":8854.0,"Rate":1.0,"Fps":9.982,"Title":{},"Chapter":{"Count":0,"Track":0,"TrackCount":0,"Desc":""}}}}
返回值字段含义：
HasVout：当显示视频时返回1
State：作为枚举给出的输入链的当前状态 0闲 1打开 2缓冲 3播放 4暂停 5停止 6已结束 7错误
Rate：输入速度为float（1.0为正常速度，0.5为半速，2.0为两倍快）
Len：输入文件的长度，以毫秒为单位 返回0的是实时流或剪辑，其长度不能确定
Fps：以秒为单位返回的帧数
Pos：在多媒体流项目中播放位置，[0.0 - 1.0]
Time：以毫秒为单位给出的绝对位置，此属性可用于通过流来查找

53) 设置输入源信息
Pos是位置 Time是时间 Rate是速度 TitleTrack标题轨
{"req":"VLC_PutInputInfo","rid":93,"para":[{"ID":1,"Pos":0.075,"Time":9667.0,"Rate":1,"TitleTrack":0},{"ID":2,"Rate":1}]}
分别返回：{"ret":0,"rid":93,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":93,"ID":2,"data":{"Ret":0}}

54) 操作章节信息
Track 章节跟踪。该属性将整数作为输入值[0-0.65535]
Opt是操作类型，1是前进 2是后退
{"req":"VLC_PutChapterInfo","rid":94,"para":[{"ID":1,"Track":0},{"ID":2,"Opt":1}]}
分别返回：{"ret":0,"rid":94,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":94,"ID":2,"data":{"Ret":0}}

55) 设置状态信息显示
Text为空时或直接无Status节点时代表取消状态显示文字，打开视频播放时，由

Config.json中的Status节点配置状态，前端可对状态显示文字进行更改
{"req":"VLC_StatusShow","rid":95,"para":[{"ID":1, "Status":{"Text":""}
},{"ID":2, "Status":{"Text":"播放异常，请检查流地址或数据是否合法！
","Color":"#00FF7F","Opacity":255,"Size":16,"Name":"宋体"}
}]}
分别返回：{"ret":0,"rid":95,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":94,"ID":2,"data":{"Ret":0}}

56) 播放失败时(网络错误)尝试恢复设置
Count 尝试次数，大于0有效 Second 尝试间隔秒数，最低1秒
{"req":"VLC_PutErrTry","rid":96,"para":[{"ID":1,"Count":3,"Second":6},{"ID":2,"Count":5,"Second":15}]}
分别返回：{"ret":0,"rid":96,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":96,"ID":2,"data":{"Ret":0}}

57) 添加轮换播放内容
比如现在4分屏播放，执行后1分屏播放添加的轮换内容，原来1、2、3分屏播放内容切换到2、3、4分屏窗口播放
Uri中如果有特殊字符= & 双引号或中文等，需要用URL编码处理后传递 Uri如非本地全路径，默认使用中间件程序Data子目录作为根目录使用
请求：
{"req":"VLC_AddOrderPlay","rid":97,"para":{"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/sub/av_stream","Name":"OrderPlay","Option":":rtsp-tcp"}}
返回：{"ret":0,"rid":97,"data":{"Ret":0}}

58) 请求显示下一帧 执行后播放暂停，效果类似在VLC桌面程序中按快捷键E的效果
Count 为帧数量，不指定时默认为1 不支持实时流，支持文件和回放流支持
{"req":"VLC_NextFrame","rid":98,"para":[{"ID":1,"Count":10},{"ID":2,"Count":10}]}
分别返回：{"ret":0,"rid":98,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":98,"ID":2,"data":{"Ret":0}}

59) 取小程序Config.json中的配置内容，举例：
请求：{"req":"VLC_GetConfig","rid":99,"para":{}}
返回：{"ret":0,"rid":99,"data":{"COM":"axvlc.dll","Caching":300,"RTSPTCP":1,"ToolBar":1,"FillWnd":1,"AutoLoop":1,"SelfGPU":1,"Mute":0,"BlockFlag":7,"PercentTime":300,"ErrTryCount":3,"ErrTrySecond":6,"BackColor":"#000000","PORT": 930}}

60) 设置小程序Config.json中的配置内容，举例：
请求：{"req":"VLC_SetConfig","rid":100,"para":{"COM":"axvlc.dll","Caching":300,"RTSPTCP":1,"ToolBar":1,"FillWnd":1,"AutoLoop":1,"SelfGPU":1,"Mute":0,"BlockFlag":7,"PercentTime":300,"ErrTryCount":3,"ErrTrySecond":6,"BackColor":"#000000","PORT": 930}}

返回：{"ret":0,"rid":100,"data":{"Ret":0}}
此设置仅对本次播放实例有效，不会实际保存到文件中

61）HTTP同步方式上传视频到服务器或下载视频到本地，上传主要用于上传截图和录像到服务器保存
Type 1同步上传 0下载(支持异步，设置Asyn为1)
Agent、Cookie及Auth Cookie可传浏览器中的Cookie或通过Auth授权验证上传权限，默认为空，非空时需要先做UrlEnocde编码
Local 上传文件本地路径或下载文件保存路径，如果不指定默认用当前打开文档，需要做UrlEncode编码
Url 上传或下载的地址，需要做UrlEncode编码
Type为1时，可增加参数Para，对应POST数据包中需要传递的参数param 默认上传文件名为upfile，否则可通过NodeName设置自己的文件参数名称
Type为0时，可增加参数MD5及FileSize，下载文件的哈希值及大小，用于校验下载文件的完整性，可默认空或0
{"req":"VLC_NetFile","rid":101,"para":{"Type":1,"Agent":"","Cookie":"","Auth":"","Para":"","Local":"","Url":"http://zorrosoft.com/wp-admin/admin-ajax.php"}}
返回：{"ret":0,"rid":35,"data":{"Ret":"0","Info":{}}} Info为上传到服务器返回的信息

62) 指定多个同类视频文件进行合并，合并结果通过事件通知返回
Type 合并引擎 1:FFMpeg
Src 需要合并的原视频路径数组 文件名及路径需要先做UrlEncode编码
Out 合并生成目标 如设置了需要先做UrlEncode编码，没设置自动在默认路径生成一个mp4视频文件名
举例：
请求：
{"req":"VLC_MergeVideo","rid":102,"para":{"Type":1,"Src":["d:/Zorro/test5.mp4","d:/Zorro/test6.mp4"],"Out": "dest.mp4"}}
返回：{"ret":0,"rid":102,"data":{"Ret":0}}
事件通知合并结果
aid:对应VLC_MergeVideo的请求rid Size:合并目标视频文件大小 PID:合并进程
File:合并最终形成的目标视频文件
{"event":"VLC_MergeOK","aid":102,"PID":0,"data":{"Ret":0,"File":"dest.mp4","Size":0}}

63) 指定分屏添加轮播内容，一般用于大视频被切片时依次轮播
比如1分屏正在播放Url地址内容(要求是文件或回放视频)，当播放结束时马上切换到轮播内容，切换时没有延迟
Uri中如果有特殊字符= & 双引号或中文等，需要用URL编码处理后传递 Uri如非本地全路径，默认使用中间件程序Data子目录作为根目录使用
请求：
{"req":"VLC_AddPreparePlay","rid":103,"para":[{"ID":1,"Uri":"d:/zorro/play.mp4","Name":"PreparePlay","Option":":rtsp-tcp"}]}
返回：{"ret":0,"rid":103,"data":{"Ret":0}}

64) 设置指定分屏窗口是否禁用双击全屏功能
DisableDBFull 设置1时禁用，默认不禁用
{"req":"VLC_PutDisableDBFull","rid":104,"para":[{"ID":1,"DisableDBFull":1},{"ID":2,"DisableDBFull":0}]}
分别返回：{"ret":0,"rid":104,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":104,"ID":2,"data":{"Ret":0}}

65) 设置指定分屏窗口是否启用播放列表
EnablePlayList 设置1时启用，默认禁用
{"req":"VLC_PutEnablePlayList","rid":105,"para":[{"ID":1,"PlayList":1},{"ID":2,"PlayList":0}]}
分别返回：{"ret":0,"rid":105,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":105,"ID":2,"data":{"Ret":0}}

66) 设置指定分屏窗口指定区域的画面放大显示
Left及Top分别为播放窗口指定区域左上角坐标，Width及Height分别为指定区域宽度和高度
当指定Width和Height为0时代表取消放大显示，按字母Q或R也可退出放大显示，或点击鼠标右键也可取消
{"req":"VLC_PutCropWindow","rid":105,"para":[{"ID":1,"Left":50,"Top":50,"Width":300,"Height":300},{"ID":2,"Left":100,"Top":100,"Width":500,"Height":500}]}
分别返回：{"ret":0,"rid":106,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":106,"ID":2,"data":{"Ret":0}}

70) 设置指定ID窗口下方文字信息，定制功能，举例：
指定的参数，需要的才设置
Text 文字内容，必须参数
Color 颜色 可用长整型(COLORREF)，也可以用#000000这样的颜色值，默认白色
Size 字体大小，默认18
请求：
{"req":"VLC_CustomText","rid":110,"para":[{"ID":1,"Text":"Hello1","Color":"16777215","Size":50},{"ID":2,"Text":"Hello2"},{"ID":3,"Text":"Hello3"},{"ID":4,"Text":"Hello4"}]}
分别返回：{"ret":0,"rid":110,"data":{"Ret":0}}

71) 指定ID窗口叠加网页中的渲染内容 Widows 8及以上版本操作系统正常支持，本机需要安装Microsoft Edge WebView2 Runtime
Url为透明网页地址，如有特殊符号请先进行UrlEncode，设置为空字符串代表取消叠加
Alpha 透明度(0-255) 默认255不透明，Windows 7系统及以下版本不支持透明
Rect节点设置叠加窗口位置和大小，不设置时代表全部区域，E代表边距默认1，X/Y/W/H分别代表叠加窗口在分屏窗口中显示的起始点和宽高，P为显示位置类型，大于0时替代X和Y，1左上 2右上 3左下 4右下

{"req":"VLC_FloatWebInfo","rid":120,"para":[{"ID":1,"Url":"https://output.jsbin.com/dopavun"},{"ID":2,"Alpha":70,"Url":"https://output.jsbin.com/dopavun"}]}
{"req":"VLC_FloatWebInfo","rid":120,"para":[{"ID":1,"Url":"https://output.jsbin.com/dopavun","Rect":{"X":0,"Y":0,"P":0,"W":300,"H":300}}]}
返回：{"ret":0,"rid":120,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":120,"ID":2,"data":{"Ret":0}}

72) 指定ID窗口叠加网页注入脚本运行
Script为脚本内容，请先进行UrlEncode，设置为空字符串代表取消叠加
Init 初始化
{"req":"VLC_FloatWebInjectScript","rid":112,"para":[{"ID":1,"Init":0,"Script":"window.getComputedStyle(document.body).backgroundColor"},{"ID":2,"Init":0,"Url":"if%20(window.parent%20!%3D%3D%20window.top)%7B%0A%09delete%20window.open%3B%0A%7D"}]}
返回：{"ret":0,"rid":112,"ID":1,"data":{"Ret":0}} 和
{"ret":0,"rid":112,"ID":2,"data":{"Ret":0}}

73）请求在嵌入窗口区域内弹出一个消息对话框
Title为弹窗标题，如果是中文或特殊符号需要先做UrlEncode
Content为弹窗内容，如果是中文或特殊符号需要先做UrlEncode
Wait为等待多少秒自动消失 0代表不自动消失
Type代表弹窗类型 0是MB_OK，具体见
https://learn.microsoft.com/zh-cn/windows/win32/api/winuser/nf-winuser-messagebox uType定义
{"req":"VLC_MsgBox","rid":52,"para":{"Title":"Alert","Content":"This is a test message","Wait":5,"Type":0}}
返回：{"ret":0,"rid":52,"data":{"Ret":"0"}}

4.2、VLC 网页播放器接收的通知
A、ActiveX 事件：
文档参考：https://wiki.videolan.org/Documentation:WebPlugin
1) vlc is in idle state doing nothing but waiting for a command to be issued 播放器空闲中
ID 是分屏序号 aid 是播放实例 ID 下同
{"event":"VLC_MediaPlayerNothingSpeciale","ID":1,"rid":1,"data":{}}

2) vlc is opening an media resource locator (MRL) 正在打开播放源
{"event":"VLC_MediaPlayerOpening","ID":1,"data":{}}

3) vlc is buffering 播放器正在缓存数据 通过 VLC_PutBlockFlag 设置掩码 4 可阻止通知，默认不阻止
{"event":"VLC_MediaPlayerBuffering","ID":1,"rid":1,"data":{"cache":0}}

4) vlc is playing a media 正在播放中，第一次 WS 连接过来会收到此通知，也可以

通过调用 VLC_GetPlay 获得 IsPlay 是否播放状态
{"event":"VLC_MediaPlayerPlaying","ID":1,"rid":1,"data":{"W":1530,"H":1078}}
W 和 H 分别为视频的高度和宽度

5) vlc is in paused state 播放暂停状态
{"event":"VLC_MediaPlayerPaused","ID":1,"rid":1,"data":{}}

6) vlc is fastforwarding 快进播放
{"event":"VLC_MediaPlayerForward","ID":1,"rid":1,"data":{}}

7) vlc is going backwards 后退播放
{"event":"VLC_MediaPlayerBackward","ID":1,"rid":1,"data":{}}

8) vlc has encountered an error and is unable to continue 出现错误不能继续，前端需捕获处理如执行重新播放等
{"event":"VLC_MediaPlayerEncounteredError","ID":1,"rid":1,"data":{}}

9) time has changed 时间改变 通过 VLC_PutBlockFlag 设置掩码 2 可阻止通知，默认不阻止，可通过持续监听此通知判断播放是否正常状态
{"event":"VLC_MediaPlayerTimeChanged","ID":1,"rid":1,"data":{"time":0}}

10) media position has changed 位置改变 通过 VLC_PutBlockFlag 设置掩码 8 可阻止通知，默认不阻止
同时返回总长度和当前播放时间
{"event":"VLC_MediaPlayerPositionChanged","ID":1,"rid":1,"data":{"position":0,"length":0,"time":0}}

11) Play stopped 播放停止
{"event":"VLC_MediaPlayerStopped","ID":1,"rid":1,"data":{}}

12) Play stop async done 异步停止播放完成
{"event":"VLC_MediaPlayerStopAsyncDone","ID":1,"rid":1,"data":{}}

13) media seekable flag has changed (true means media is seekable, false means it is not)
{"event":"VLC_MediaPlayerSeekableChanged","ID":1,"rid":1,"data":{"seekable":0}}

14) media pausable flag has changed (true means media is pauseable, false means it is not)
{"event":"VLC_MediaPlayerPausableChanged","ID":1,"rid":1,"data":{"pausable":0}}

15) media has changed 播放源改变
{"event":"VLC_MediaPlayerMediaChanged","ID":1,"rid":1,"data":{}}

16) chapter has changed (DVD/Blu-ray) 章节已更改（DVD/蓝光）
{"event":"VLC_MediaPlayerChapterChanged","ID":1,"rid":1,"data":{"chapter":0}}

17) the number of video output has changed 视频输出数量发生变化
{"event":"VLC_MediaPlayerVout","ID":1,"rid":1,"data":{"count":0}}

18) audio volume was muted 静音
{"event":"VLC_MediaPlayerMuted","ID":1,"rid":1,"data":{}}

19) audio volume was unmuted 取消静音
{"event":"VLC_MediaPlayerUnmuted","ID":1,"rid":1,"data":{}}

20) audio volume has changed 音量改变
{"event":"VLC_MediaPlayerAudioVolume","ID":1,"rid":1,"data":{"volume":0}}

21) KeyDown 按键按下
{"event":"VLC_KeyDown","ID":1,"rid":1,"data":{"KeyCode":0,"Shift":0}}

22) KeyPress 按键码
{"event":"VLC_KeyPress","ID":1,"rid":1,"data":{"KeyCode":0}}

23) KeyUp 按键弹起
{"event":"VLC_KeyUp","ID":1,"rid":1,"data":{"KeyCode":0,"Shift":0}}

24) MouseMove 鼠标移动通知 通过 VLC_PutBlockFlag 设置掩码 1 可阻止通知，默认阻止
{"event":"VLC_MouseMove","ID":1,"rid":1,"data":{"Button":0,"Shift":0,"X":0,"Y":0}}

25) MouseDown 鼠标按下
{"event":"VLC_MouseDown","ID":1,"rid":1,"data":{"Button":0,"Shift":0,"X":0,"Y":0}}

26) MouseUp 鼠标弹起
{"event":"VLC_MouseUp","ID":1,"rid":1,"data":{"Button":0,"Shift":0,"X":0,"Y":0}}

/// 以下几个非 ActiveX 控件事件通知
27) Selected 选中分屏窗口序号通知
{"event":"VLC_Selected","ID":1,"rid":1,"data":{}}

28) ToggleFullScreen 某个分屏窗口切换全屏通知
{"event":"VLC_ToggleFullScreen","ID":1,"rid":1,"data":{}}

29) 录像结束通知(限时录制时发出)
{"event":"VLC_StopRecord","PID":1,"rid":1,"data":{"Ret":0,"File":"D:/Zorro/test.mp4","Size":321}}

PID 是开始录像时返回的 ID

30) 定制图标被点击
CID 标识窗口序号
{"event":"VLC_CustomClicked","CID":1,"rid":1,"data":{}}

31) 叠加网页导航结果通知
{"event":"VLC_WebNavResult","ID":1,"rid":1,"aid":1,"data":{"Result":""}]}}
Result UrlEncode 后的请求结果

32) 叠加网页注入脚本结果通知
{"event":"VLC_WebScriptResult","ID":1,"rid":1,"aid":1,"data":{"Result":""}]}}
Result UrlEncode 后的请求结果

33) 播放错误日志通知
{"event":"VLC_PlayLog","ID":1,"rid":1,"aid":1,"data":{"Level":1,"Line":18,"Module":"
","Desc":""}]}}
Level 日志级别 1 错误日志 2 警告日志 4 通知日志
Module 模块名称
Line 代码行数
Desc 日志描述

34) 播放视频选中局部放大通知
{"event":"VLC_ZoomOut","ID":1,"rid":2,"data":{"X":50,"Y":50,"Width":1024,"Height":
768}}
data 中参数分别为点选 X、Y 坐标及选中区域宽度和高度

35) 播放视频局部放大结束通知
{"event":"VLC_ZoomEnd","ID":1,"rid":2,"data":{}}

B、PluginOK 支持的事件通知：

1) VLC_FullScreen 播放器是否响应了热键全屏
{"event":"VLC_FullScreen","data":{"FullScreen":0}} FullScreen 为当前是否全屏标记

2) Wrl_AppletExit 小程序退出通知
{"event":"Wrl_AppletExit","data":{"ID":1}}

请求参数的 JSON 数据包，请确保是 UTF-8 无签名的编码。
更多关于 PluginOK 中间件的请求说明，请参考安装目录中的文档 TestWrl.txt、SDK 包中的“PluginOK 开发者手册.pdf”及“PluginOK 中间件安全解决方案.pdf”。
如还有疑问请直接联系微信客服：ZorroSoft 咨询，或加 QQ 群：23126938 和大家交流。

五、中间件对接

在上一节中我们介绍了如何启动 VLC 网页播放器并对其控制的方法，中间件本身还提供其它一些功能，比如获取中间件和 VLC 程序的版本、判断指定版本是否需要升级、获取本机 MAC 地址、本机路由器 MAC 地址、列出本机已安装小程序列表、指定浏览器打开网页等等，详细说明请参考程序安装目录下的文档“testwrl.txt”。

关于如何实现在线升级中间件或 VLC 网页播放器，具体参考 SDK 包中的文档“PluginOK 中间件制作升级包说明.pdf”。

六、常见问题解决

1、 多路播放时如何设置播放源？

在第四章开发集成中，已经在 HTML 网页中展示了如何配置播放多个 RTSP 源，最新推荐是直接在启动参数中配置：

"Web":[{"ID":2,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Option":":rtsp-tcp"},{"ID":3,"Uri":"http://www.zorrosoft.com/Files/PluginOKBrowserApplet.mp4","Option":":file-caching=300"},{"ID":4,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/sub/av_stream","Option":":network-caching=300"}]

其中 ID 为窗口序号，Uri 指定具体的播放源，第一个窗口的播放源可以在 Open 参数中指定，也可以在这里配置，如在 Open 中指定而 Web 中没有配置其它分屏窗口，其它分屏窗口会自动使用 Open 源来播放。:network-caching=300 是指定缓存时间，控制播放延迟效果的关键参数，如在这里设置了，那么 Config.json 中的配置就无效，:rtsp-tcp 同理。

也可在启动播放器窗口后，新建一个 WS 连接到启动后的播放程序侦听端口，再调用以下指令指定播放源，ShowType 指定分屏数量，如果不变就不设置 Play 重新指定播放内容

请求范例 1：{"req":"VLC_ChangePlay","rid":79,"para":{"ShowType":2}}

请求范例 2：

{"req":"VLC_ChangePlay","rid":79,"para":{"Play":[{"ID":1,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"BrowserApplet1","Option":":rtsp-tcp"},{"ID":2,"Uri":"http://www.zorrosoft.com/Files/PluginOKBrowserApplet.mp4","Name":"BrowserApplet2","Option":":file-caching=300"}]}}

请求范例 3：

{"req":"VLC_ChangePlay","rid":79,"para":{"ShowType":3,"Play":[{"ID":1,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"BrowserApplet1","Option":":rtsp-tcp :network-caching=300"},{"ID":2,"Uri":"http://www.zorrosoft.com/Files/PluginOKBrowserApplet.mp4","Name":"BrowserApplet2","Option":":file-caching=400"},{"ID":3,"Uri":"rtsp://wzh:test123456@192.168.1.8:554/h264/ch1/main/av_stream","Name":"BrowserApplet3","Option":":rtsp-tcp"}]}}

分别返回：{"ret":0,"rid":80,"data":{"Ret":0,"Play":[]}}

2、 实际播放的网络延迟达不到要求如何做？

首先检查默认的缓存参数设置是否合适，推荐设置 300-500 毫秒。另外检查摄像头的硬件配置性能，如配置太低也影响画面延迟。还有就是播放电脑终端的硬件性能，尽量采用性能好一些的电脑进行测试。此外，确保网络是通畅的，没有其他程序大量占用网络带宽，尤其是查看显卡是否为独立显卡还是集成显卡，CPU 和内存不能配置太低，必要时可请求我们的远程技术支持。播放多路时，由于影响因素较多，把大的分屏窗口视频源设置为主码流播放，其它辅助预览窗口采用子码流播放，另外把网络缓存时间设置大一些。

3、 VLC 网页播放器是否收费？

大家都喜欢免费的东西，不过大家也常会听到一句话，免费的往往也是最贵的。作为一个正常的商业软件公司，需要有合理的利润来支持工程师们继续维护升级，确保软件的稳定性和可用性，所以商业用途是收费的，如果有正式的公益机构出具证明项目用于公益，可向我们申请免费授权使用。本软件的授权费用也不贵，平均下来一个电脑终端的费用不到 200 元，购买授权后能够得到持续、长期、优质的售后服务，无疑是值得的。

4、 播放引擎只能用 VLC 的吗？

当然不是，只不过 VLC 是当前市场上最流行的开源多媒体播放器，全球有大量的优秀工程师在不断维护升级，稳定性和播放性能无疑是值得信赖的。如果您有需要，可选择我们的多引擎网页播放器，底层除了支持 VLC 引擎播放之外，支持调用海康威视、浙江大华的私有协议播放，另外支持调用 FFPlayer 引擎播放，也可以提供华为等厂家的播放控件进行定制，海康威视、浙江大华的原生播放 SDK 往往功能更丰富，比如实现云台控制，语音对讲等。

5、 分屏播放样式都有哪些，没有我要的咋办？

目前 VLC 网页播放器提供了三十一种播放分屏风格，如您的需求特殊，可联系我们定制开发，也可以购买这个播放器的源代码，自行开发独有分屏样式。具体的分屏播放样式，从 1 到 31 的分屏样式都是支持的，还有一些特殊分屏效果，可以自行修改 ShowType 参数一一体验看效果，这里就不详细展开介绍了。

6、 VLC 网页播放器如何实现全屏播放？

在播放窗口中双击，或当前选中焦点视频中按字母 F 键，可自动切换为全屏播放，热键按键 ESC 可退出全屏播放或按字母 F 键退出全屏，也可通过建立 WS 连接到播放器的端口，请求执行某个播放分屏的全屏操作，请求 JSON 如下：
{"req":"VLC_VideoPut","rid":72,"para":[{"ID":1,"FullScreen":1},{"ID":2,"FullScreen":1}]}，也可以请求切换全屏指令 {"req":"VLC_VideoToggleFullscreen","rid":73,"para":[{"ID":1}]}
另外可以请求播放器整个窗口的全屏，请求指令是：
{"req":"Wrl_AppletControl","rid":2,"para":{"ID":"1","Code":2}} 需要注意的是，这个请求并不能让某个分屏窗口全屏。ID 是播放器的实例序号。

7、 VLC 网页播放器是否能提供免插件的版本？

免插件自然是好，但是首先是要满足甲方客户提出的播放性能指标。甲方希望是无插件，主要还是担心插件的安全性和后续升级麻烦问题，为此 VLC 网页播放器方案提供了对应的安全调用机制和前端可自行请求升级的机制，最大化的解决了这些后顾之忧。只需保证安装的播放器程序是安全的即可，必要的话可通过开放播放器源代码来打消客户对安全的顾虑。想要低延迟或多路流畅播放，只能

采用这样的技术方案。其实所谓的免插件是要下载 JS 或 WASM 版的播放器文件的，而且浏览器清理缓存，都会导致这些 JS 版浏览器程序重新下载。互联网上也有一些开源免费的无插件播放器可供体验，如果需要极致的播放效果，选择 PageHi VLC Web Player 无疑是不会有错的，因为您已经选择了市场上最好的网页播放器。

8、 和使用 WebRTC(网页即时通信)协议的播放有何差异？

虽然现代版浏览器已经对 WebRTC 支持的较好，但并不支持 H.265 编码的视频流，况且还是需先进行协议转换再播放，性能等各方面还是欠缺的。单路播放尚可接受，多路、高清和 H.265 编码视频流还是无法胜任甲方客户实际需要的。

9、 发送启动播放器的指令后没有看到播放画面？

首先检查本机是否已经安装了 VLC 网页播放器软件和对应的 VLC 桌面程序版本；其次在启动播放的过程中，确认下使用的浏览器版本和类型，一般套壳的小众浏览器是不支持的，另外启动时尽量确保浏览器在前景窗口显示；如果还不行，可尝试重启电脑后再试，也可把中间件运行目录下的整个 data 子目录打包给客服人员用来分析故障，自己也可以检查日志文件 data 目录下 Temp 子目录的日志 VlcWebPlayer.txt 或 VlcPlayerApplet.txt 中输出的播放地址，确保是可以正常播放的。

10、 如何监控播放状态，出现错误咋办？

前端可通过 WS 连接到播放器侦听端口后，收取 VLC_MediaPlayerBuffering 事件通知得到播放器网络请求数据情况，收取 VLC_MediaPlayerEncounteredError 通知得到播放错误信息，如果不是主动停止播放或内容本身播放结束，收取 VLC_MediaPlayerTimeChanged 或 VLC_MediaPlayerPositionChanged 获取播放时间、位置信息，没有通知了，也可以认为播放异常了，需前端处理。

11、 如何屏蔽中间件和播放器中的运行日志？

在前端请求 WS 连接的参数 flag 掩码中，改为 0 即可，测试网页中默认设置的 1，所以会输出比较多的日志信息，一旦系统运行稳定，可以屏蔽日志输出，提升系统运行速度。

12、 如何在同一个网页中启动播放器的多个实例？

默认情况下，在不同的网页中可以分别启动一个播放器的实例。如您需要在同一个网页中启动播放器的多个实例，方法如下：A、启动每一个播放器实例，都应该在单独到中间件的 WS 连接中进行，便于单独释放和控制；B、请求启动播放器的 Url 指定的网页中，不能设置多路播放的源和显示位置信息，统一改为启动播放的 JSON 参数中指定，也就是 Flag 需要在原来的基础上+64；C、请求启动播放器的 JSON 参数 Flag 值，Flag 需要在原来的基础上+512，最终支持启动多实例的 Flag 就是 578 的数字；D、启动多个实例时应先后进行，不能同时发起启动请求，等完成一个后再启动下一个，具体可参考我们提供的在线体验网页源码工程。

13、 需要播放的电脑无法连接外网时能用吗，如何测试？

无法连接外网的电脑也是可以用的，只需要终端电脑网络能访问内网的授权服务器和终端电脑能访问 RTSP 流即可，也可以申请单机版的授权来测试。默认开通的是网络版授权在线体验，需要终端电脑可以访问我们的公网授权服务器。

14、 启动播放后一会，播放窗口消失如何解决？

一般这种情况是电脑的硬件配置不能支撑太短的网络延迟时间，需要修改播放小程序子目录的 Config.json 配置文件中参数 Caching 值，调大，比如从 300

调到 500 甚至 1000 等，然后重启播放再试。

15、 在多路播放中，如何指定不一样的播放配置参数，比如网络延迟及网络连接方式？

播放配置参数，默认有播放小程序目录的 Config.json 决定，在启动播放或请求切换播放时，Option 没有指定，那么就用默认值，如果指定了，就用指定的配置。当有的视频流不支持 TCP 连接播放时，需要修改配置文件中的参数"RTSPTCP":1，改为 0 保存。然后在需要使用 TCP 连接播放的视频流中，指定"Option":":rtsp-tcp"即可。

16、 如何获得 VUE 框架的在线演示网页源代码工程？

请访问网址：https://gitcode.net/zorrosoft/pluginok/-/tree/master/Demo 获取，VlcJS 为纯 JS 脚本的，VlcVue2 为 VUE2 的，VlcVue3 为 VUE3 的。

17、是否支持 HTTPS 网站使用？

答：是支持的。默认授权是开通 HTTP 网站的，前端 WS 连接地址用 IP 即可，如果您的 B/S 是 HTTPS 的，请提前和我们沟通说明，需要您在 HTTPS 网站主域名下申请一个其它地方不会用到的二级域名，然后用这个二级域名申请其 SSL 证书并下载证书文件包发给我们，需要下载 Apache 的，我们将用这个证书包给您制作专用的授权文件。

18、 绑定 HTTPS 网站的 SSL 证书到期如何处理？

答：如果已经到期，需要按第 17 条的方式，重新申请新的证书文件包发给我们做授权文件。如果证书还没到期，最好是提前几天申请新的证书，然后自己可以根据 testwrl.txt 文档中的说明自行更新证书信息到授权文件，具体指令是 Wrl_UpdateSslCert，获得新的授权文件后，还可以根据文档中的指令 Wrl_UpdateAuth 让终端强制更新授权文件，或者制作一个中间件的升级包强制终端抓紧更新一下，否则证书到期后前端就无法连接到中间件服务端口，影响使用。更多详情请参考 SDK 包中的文档“HTTPS 网站中配置 WSS 连接说明书.pdf”