



H5PlayerWEBSDK 开发向导 (NVR 扩展)

修订日期	
2023/3/24	
2023/7/11	

用户登陆

Method	RPC.login(username, password, httpOnly)			
Params				
Name	Type		Description	Example
username	string		账户名	Admin
password	string		账户密码	Test123
httpOnly	Boolean		是否在 cookie 中保存当前会话 session （已禁用）	
Request Method				

Response Params (multipart , key=value format in body , Heartbeat in body)				
Name	Type	R/O	Description	Example
isPwdOverdue	bool	否	用户密码是否过期。只有登陆成功后才返回这个字段	true
keepAliveInterval	int	空	保活周期，单位：秒 登陆成功后，由服务端返回给客户端的最大保活周期，如果消息中包含这个字段，则应以小于此值发送保活包；如果没有则按照原先的流程执行	30
Response Example				
{"result":true,"params":{"keepAliveInterval":60,"isPwdOverdue":false},"session":"6d5455a0783e3a315ce6ab232eeaab18","id":6}				

注销

Method	RPC.Global.logout()			
Params				
Request Method				

Response Params (multipart , key=value format in body , Heartbeat in body)				
Name	Type	R/O	Description	Example
Response Example				
{ "result":true,"params":{"keepAliveInterval":60,"isPwdOverdue":false},"session":"6d5455a0783e3a315ce6ab232eeaab18","id":6}				

PlayerControl 基本使用流程

拉流及事件注册

```
Const option = {} // 不同参数决定该对象不同作用：回放 预览 下载

Const player = new PlayerControl(option) // 实例化 player 对象

player.on('eventName',()=>{}) // 注册事件监听

player.init(canvasDom,videoDom) //挂载 dom 节点

player.connect(); // 开启拉流
```

方法调用

设置回放音量	player.setAudioVolume(vol)
暂停	Player.pause()
播放	Player.play()

PlayerControl 实例化

Method	实例化拉流对象		
Params			
Name	Type	Description	Example
Example	new PlayerControl(option)		
<div>options.wsURL;// websocket 连接</div> <div>options.rtspURL;// rtsp 连接</div> <div>options.isTalkService;//是否开启语言对接</div> <div>options.playback;//是否回放</div> <div>options.range; //回放开始播放的秒数 视频裁剪时间与视频的</div> <div>StartTime 时间差值</div> <div>options.isAudioFlag; //是否是纯音频播放</div> <div>options.decodeMode;//解码类型 video</div> <div>options.playbackIndex; // 回放视频的索引</div> <div>options.username;//用户名</div> <div>options.password;// 密码</div> <div>options.speed;// 倍速拉流</div> <div>options.isDownLoad;// 是否下载当前拉流</div> <div>Options:isPrivateProtocol // 是否私有协议</div>			

【注：具体使用场景及对应声明方式请查看 H5PlayerWEBSDK 开发向导】

PlayerControl 初始化

Method	init		判断是否存在文件
Params			
Name	Type	Description	Example
Canvas	DomEle	Canvas 对象	
Video	DomEle	Video 对象	
Example			
PlayerControl .init(\$canvas, \$video)			

PlayerControl 开启连接

Method	connect		开启连接
Params			
Name	Type	Description	Example
、		是否开启预览备份	
Example			
PlayerControl .connect()			

PlayerControl 注册自定义事件

[ResolutionChanged]	分辨率改变,返回具体分辨率
[PlayStart]	播放开始,指通过 canvas 开始播放第一帧或者 video 加载了 initsegment, 返回值为"PlayStart"
[DecodeStart]	开始解码,指通过 SPS/DH 帧头 获取到第一帧的分辨率, 仅 canvas 可用, 返回值为{width:xxx, height: xxx}
[UpdateCanvas]	每绘制一帧时触发一次,返回值为{width:xxx, height: xxx}
[GetFrameRate]	返回帧率, 在 SDP 信息中获取
[FrameTypeChange]	编码类型改变, 返回新类型, 支持"MJPEG", "H264", "H265"
[Error]	错误信息, 返回值为{errorCode: xxx} ,错误码类型如下 * *** 101 延时大于两秒 *** * *** 201 不支持当前音频格式 *** * *** 404 找不到流 ***
[MSEResolutionChanged]	由于 MSE 不能获取分辨率改变的事件, 故只能在收到一帧时进行判断
[audioChange]	音频改变事件, 当音频编码格式改变或者采样率改变时触发

【注：具体事件对应回调信息请查看 H5PlayerWEBSDK 开发向导】

```
/*  
  * @param eventName 事件名  
  * @param callback 回调函数  
  * @return player  
  */  
on: function (eventName, callback)  
  
  playerControl.on('playstart', ()=>{})
```


基于 SDK 事件的 PlaybackByPeriod 方法实现说明

在业务场景按时间段回放中，由于录像可以设置打包区间，即以时间段 a 作为录像打包时长，所以按时间回放存在跨录像段的情况

PlayerControl playback 参数配置中 range 可设置录像开始播放的开始时间

```
{
    wsURL: wsURL,
    rtspURL: _rtspURL,
    username: username,
    password: password,
    isPrivateProtocol: false,
    realm: RPC.realm,      //域信息

    speed: 16, // 倍速拉流

    range: 0 // 相对于录像开始时间的偏离量，单位秒

    playback: true, // 是否回放

    isDownLoad: true
}
```

'UpdateTimeStamp'回调事件可回调当前录像的时间戳

```
Player.on('UpdateTimeStamp', function (e) {
    console.log('UpdateTimeStamp: ' + JSON.stringify(e))
    // UpdateTimeStamp: {"timestamp":1679609103}
})
```

思路步骤：

- 1、录像文件搜索：参照 demo 中 `onSearchRecord` 方法，请求到当前时间段内录像列表
- 2、播放开始时间所在录像文件：根据开始时间判别文件列表中对对应起始播放文件，使用 `rang` 参数从开始时间处开始回放
- 3、递归播放剩余录像文件：按顺序播放所处时间段内完整录像文件
- 4、结束回放：通过注册'`UpdateTimeStamp`'事件，返回的时间戳大于等于结束时间时，结束播放