

插件播放器Web SDK开发文档

目录

- 1 修改历史
- 2 功能特性
- 3 快速上手
 - 3.1 插件安装
 - 3.2 SDK引入
 - 3.2.1 script标签方式引入
 - 3.2.2 模块化方式引入（推荐）
 - 3.3 实例化
- 4 JavaScript接口 - 交互类
 - 4.1 createPlayer - 创建播放器窗口（异步）
 - 4.2 getToken - 获取鉴权token（异步）
 - 4.3 bindChannel - 绑定通道
 - 4.4 showWnd - 显示窗口
 - 4.5 hideWnd - 隐藏窗口
 - 4.6 setPlayMode - 切换播放模式
 - 4.7 openDownloadTaskDlg - 打开下载管理器
 - 4.8 uninit - 插件销毁
 - 4.9 captureView - 视频抓图
 - 4.10 setPreviewPullStreamConfig - 设置实时预览默认拉流策略
 - 4.11 setImmediatelyPlaybackConfig - 设置即时回放策略
 - 4.12 setPlaybackConfig - 设置录像回放策略
 - 4.13 cuttingPartWindow - 窗口部分区域切除
 - 4.14 repairPartWindow - 清除区域切除
 - 4.15 refreshPosition - 刷新播放器定位
 - 4.16 preload - 预加载（异步）
 - 4.17 getAppInfo - 获取插件下载地址（异步）
 - 4.18 setPreviewStream - 切换实时预览高清/标清
 - 4.19 setPlaybackSpeed - 设置录像回放的播放速度
 - 4.20 videoRecord - 视频录像控制
 - 4.21 setVideoPlayerFullScreen - 全屏/退出全屏
 - 4.22 downloadMosaicTask - 下载拼接任务
 - 4.23 setPlaybackTimePos - 录像回放跳播
 - 4.24 startRecordDownload - 下载回放录像
- 5 JavaScript接口 - 云台控制
 - 5.1 startPTZControl - 开启云台控制
 - 5.2 stopPTZControl - 结束云台控制
 - 5.3 startPTZMove - 开始云台移动
 - 5.4 stopPTZMove - 结束云台移动
 - 5.5 startPTZZoom - 开始云台变倍
 - 5.6 stopPTZZoom - 结束云台变倍
 - 5.7 startPTZChange - 开始变焦 光圈改变
 - 5.8 stopPTZChange - 结束变焦 光圈改变
 - 5.9 getPTZPresets - 获取云台预置点（异步）
 - 5.10 locatePTZPreset - 调用云台预置点（异步）
- 6 JavaScript接口 - 事件监听
 - 6.1 tellCurSelectedWnd - 窗口选中
 - 6.2 tellPluginClosedByUser - 插件被用户手动关闭
 - 6.3 tellPlaybackPlayTimePos - 录像回放播放进度
 - 6.4 tellVideoStateChanged - 视频状态发生变化
 - 6.5 tellPluginClosed - 插件进程退出

1 修改历史

日期	内容
2024/11/13	插件增加在客户端绑定显示的配置，createPlayer，preload方法参数options增加loadMode选项
2024/07/26	窗口的全屏和退出全屏增加监听事件6.6/6.7
2024/04/12	插件为解决跨域增加部分参数，createPlayer，preload方法增加topWindow，iframePosition参数
2024/04/03	插件增加中英文配置，createPlayer，preload方法参数options增加language选项
2023年06月3日	修复currentFramePosition在计算播放区域的位置时的错误计算方式
2023年04月8日	getToken增加请求异常的错误捕捉，反馈到preload，createPlayer
2023年03月24日	在初始化finishInit监听中增加判断是否需要水印的操作，如需要就调接口获取水印图片传给插件显示在画面上；
2023年03月17日	新增4.24 startRecordDownload - 下载回放录像
2022年07月12日	新增4.23 setPlaybackTimePos - 录像回放跳播
2022年07月02日	新增4.22 downloadMosaicTask - 下载拼接任务
2022年06月22日	(1)4.1 createPlayer-创建播放器窗口参数playMode增加枚举值 splice-拼接视频回放 rtmp-RTMP推流播放 (2)4.1 createPlayer-创建播放器窗口 和 4.3 bindChannel-绑定通道中对deviceSn参数说明修改，除可支持设备序列号，也可以支持拼接任务ID，RTMP推流的live ID，具体对应取决于playMode (3)6.3 tellPlaybackPlayTimePos-录像回放播放进度增加参数 wndMode pastTime durationTime (4)6.4 tellVideoStateChanged-视频状态发生变化增加参数 wndMode
2022年06月07日	(1)6.5 tellPluginClosed新增枚举值(见参数reason的说明) (2)新增4.21 setVideoPlayerFullScreen - 全屏/退出全屏
2022年05月30日	(1)新增4.18 setPreviewStream - 切换实时预览高清/标清 (2)新增4.19 setPlaybackSpeed - 设置录像回放的播放速度 (3)新增4.20 videoRecord - 视频录像控制 (4)3.3 实例化增加参数portCount

日期	内容
2022年05月27日	(1)新增6.5 tellPluginClosed - 插件进程退出
2022年05月26日	(1)3.3 实例化中的container参数改为非必填 (2)4.1 createPlayer增加参数container（若实例化不传入container，则此处必填） (3) 新增 4.17 getAppInfo - 获取插件下载地址（异步）
2022年05月24日	(1)4.1 createPlayer增加参数channelName (2) 4.3 bindChannel增加参数channelName
2022年05月23日	(1)4.13 cuttingPartWindow参数支持HTMLElement（DOM元素）类型 (2)4.14 repairPartWindow参数支持HTMLElement（DOM元素）类型
2022年05月20日	(1)新增4.16 preload - 预加载（异步）
2022年05月19日	(1)新增 6.3 录像回放播放进度 (2)4.1 createPlayer增加参数bottomToolBar、videoMenuBt (3)4.1 createPlayer参数videoRights增加 0x2000-连接控制（断开、重连、清除） (4)修改 6.1 tellCurSelectedWnd 窗口选中 status参数的说明 (5)新增 6.4 tellVideoStateChanged 视频状态发生变化 (6)4.15 refreshPosition增加参数players
2022年05月18日	(1)4.1 createPlayer删除channelNum参数 (2)4.3 bindChannel删除channelNum参数
2022年05月16日	(1)新增 6 JavaScript接口 - 事件监听
2022年05月14日	(1)3.3 实例化增加container参数 (2)4.1 createPlayer中的container参数更名为panel (3) 4.1 createPlayer增加参数playbackColor和videoWndConfig (4)新增4.15 刷新播放器定位
2022年05月05日	(1)调整文档目录结构，将云台控制相关接口放到第5部分 (2)4.3 bindChannel增加wndId参数

2 功能特性

1. 支持直播，支持播放时截图，截图保存格式为jpg；支持用户停止播放；支持全屏播放
2. 支持设备前端录像回放，支持按周期（最大周期为一天）播放；支持播放时截图，截图保存格式为jpg；支持用户录像回放控制，包含暂停、恢复、跳播，倍速播放（1/2/4/8）；支持用户停止播放；支持全屏播放，全屏播放时不支持跳播

功能项	描述
流媒体协议	JVMP私有流媒体协议
音频编码格式	支持G711音频解码

功能项	描述
视频编码格式	支持H.264（AVC） 支持H.265（HEVC）
支持的设备	公有云协议设备（中维世纪私协设备） 国标28181协议设备
音频控制	支持
视频下载	支持

3 快速上手

3.1 插件安装

运行插件安装包exe文件执行安装
安装完毕后自动运行
建议在系统界面上添加下载该文件的链接，以方便用户使用

3.2 SDK引入

3.2.1 script标签方式引入

```
<script src="./player-plugin.full.js"></script>
```

3.2.2 模块化方式引入（推荐）

```
import PlayerPluginGroup from './player-plugin.full.js'
```

3.3 实例化

参数

参数名称	类型	是否必填	说明
baseUrl	String	Y	请求服务地址根URL
ak	String	Y	app_key
sk	String	Y	app_secret
container	String HTMLElement	N	多个播放器窗口的外部容器
portCount	Number	N	WebSocket连接尝试轮询的端口数量(默认为5个)
httpErrCallback	Function	N	调用接口报错回调函数(接收错误码和错误信息)
调用示例			

```
const playerGroup = new PlayerPluginGroup({
  ak: 'xxx',
  sk: 'xxxx',
  baseUrl: '/',
  container: document.getElementById('container'),
  portCount: 3,
  httpErrCallback: (code, msg) => { console.error(`${code}: ${msg}`) }
})
```

由于目前采用的是多窗口的模式，该 `PlayerPluginGroup` 的实例，代表一个播放器组也就是其中可以包含多个播放器的实例

注意该处的 `container` 参数，区别于 4.1 中的 `panel`

4 JavaScript接口 - 交互类

4.1 createPlayer - 创建播放器窗口（异步）

```
createPlayer(options, topwindow?, iframePosition?)
```

options参数

参数名称	类型	是否必填	说明
playMode	String	Y	播放模式 preview-实况预览 playback-设备回放 splice-拼接视频回放 rtmp-RTMP推流播放
container	String HTMLElement	N	多个播放器窗口的外部容器
panel	String HTMLElement	Y	单个播放器窗口的定位元素, 可以为选择器或DOM对象
layout	Number	Y	分屏数量（多窗口模式下传1即可）
language	'zh' 'en'	N	插件的语言环境，默认值为 'zh' 中文
loadMode	'web' 'app'	N	插件绑定的环境，是浏览器还是客户端，默认为 'web' 浏览器
deviceSn	String	N	设备序列号/拼接任务ID/live Id（不传时仅加载播放器窗口，如果传入，加载窗口后立即执行视频拉流）
channelName	String	N	通道名称（如果不指定则获取实际通道名称）
wndTitle	String	N	播放器标题（不指定时将不显示标题栏，注意传值会直接显示在标题上）

参数名称	类型	是否必填	说明
videoRights	Number	N	视频权限（多种组合直接相加） 空表示拥有所有能力，0-表示没有任何能力 0x0001-高清、标清切换 0x0002-抓图 0x0004-录像 0x0008-声音控制 0x0010-对讲 0x0020-云台控制（弹窗） 0x0040-码流信息显示、隐藏控制 0x0080-原始比例、自适应切换 0x0100-回放下载 0x0200-回放快进、快退 0x0400-设置录像回放查询时间 0x0800-实时预览及时回放 0x1000-实时预览录像回放 0x2000-连接控制（断开、重连、清除）
startTime	String	N	回放开始时间（格式：yyyyMMddHHmmss）
endTime	String	N	回放结束时间（格式：yyyyMMddHHmmss）
playbackColor	Object	N	回放时间轴颜色设置
+timeAxisBGColor	Number	Y	录像/及时回放单条时间轴背景颜色
+timeAxisNormalColor	Number	Y	录像/及时回放单条时间轴正常录像颜色
+timeAxisAlarmColor	Number	Y	录像/及时回放单条时间轴告警录像颜色
videoWndConfig	Object	N	视频窗口颜色设置
+wndBGColor	Number	Y	视频窗口背景颜色
+wndBorderColor	Number	Y	视频窗口边框颜色
+selWndBorderColor	Number	Y	选中视频窗口边框颜色
+selWndBorderSize	Number	Y	选中窗口的边框宽度
bottomToolBar	Number	N	是否显示底部工具栏 0-不显示(默认) 1-显示
videoMenuBt	Number	N	是否显示视频中菜单按键 0-不显示(默认) 1-显示
playerId	String	N	播放器ID（不指定则自动生成）

topWindow参数

说明：在存在跨域的情形下为必传项，务必在取得数据后再调用该方法。参数的取值是iframe最顶层的window下的取值。

参数名称	类型	是否必填	说明
screenX	number	Y	最顶层的window的screenX属性
screenY	number	Y	最顶层的window的screenY属性
outerWidth	number	Y	最顶层的window的outerWidth属性
outerHeight	number	Y	最顶层的window的outerHeight属性
innerHeight	number	Y	最顶层的window的innerHeight属性
innerWidth	number	Y	最顶层的window的innerWidth属性
screen	number	Y	最顶层的window的screen属性window.screen
+height	number	Y	最顶层的window的screen属性下的height属性
+width	number	Y	最顶层的window的screen属性下的width属性
+availHeight	number	Y	最顶层的window的screen属性下的availHeight属性
+availWidth	number	Y	最顶层的window的screen属性下的availWidth属性
document	number	Y	最顶层的window的document属性
+title	string	Y	最顶层的window的document属性下的title属性

iframePosition参数

说明：在存在跨域的情形下为必传项，务必在取得数据后再调用该方法。参数的取值是iframe最顶层的window下的取值。

参数名称	类型	是否必填	说明
x	number	Y	在最顶层的window的下获取插件所在的iframe的位置横坐标 xxlframe.getBoundingClientRect().left
y	number	Y	在最顶层的window的下获取插件所在的iframe的位置纵坐标 xxlframe.getBoundingClientRect().top

调用示例

```
let topwindow = {
  //..得到的值
}
let iframePosition = {
  //..得到的值
}
```

```

playerGroup.createPlayer({
  playMode: 'preview', // 模式 preview或playback
  container: document.getElementById('container'), // 外部容器
  panel: this.$refs.player[index], // 定位元素
  layout: 1, // 播放面板数量
  deviceSn: '1001_0', // 设备序列号
  channelName: '通道1', // 通道名称
  wndTitle: '播放器1', // 播放器标题
  videoRights: 0x0001 + 0x0002 + 0x0004 + 0x0008 + 0x0010 + 0x0040 + 0x0100 +
0x0200, // 视频权限
  startTime: '20220425000000',
  endTime: '20220425235959',
  playbackColor: { // 回放时间轴颜色设置
    timeAxisBGColor: 0x6a2dd8,
    timeAxisNormalColor: 0x158f58,
    timeAxisAlarmColor: 0x8cc12f
  },
  videowndConfig: { // 视频窗口颜色设置
    wndBGColor: 0xc1952f,
    wndBorderColor: 0xce368d,
    selwndBorderColor: 0x3697ce,
    selwndBorderSize: 3
  },
  bottomToolBar: 0,
  videoMenuBt: 0,
  playerId: '1'
}, topWindow, iframePosition).then(player => {
  player.setPreviewPullStreamConfig(previewPullStreamConfig) // 设置实况预览默认拉流策略
  player.setImmediatelyPlaybackConfig(immediatelyPlayback) // 设置即时回放策略
  player.setPlaybackConfig(playbackConfig) // 设置录像回放策略
})

```

该方法的异步回调执行代表插件窗口初始化完毕

如果有需要在初始化完毕后执行的操作，应在该异步回调中执行

由于云台控制在页面上，所以云台控制开启或释放时可能涉及一些页面的UI改变
 比如demo里面做的遮罩层的隐藏、预置点的加载等等
 这种情况下需要监听 `ptzControlChangeSuccess`

4.2 getToken - 获取鉴权token（异步）

参数

无

调用示例

```
await playerGroup.getToken()
```

在4.1 `createPlayer`中会自动调用该方法，如果在执行`createPlayer`之前需要先获取一下鉴权token用以执行其他操作
 可以在外部调用`getToken`

4.3 bindChannel - 绑定通道

参数

参数名称	类型	是否必填	说明
option	Object	Y	绑定设置
+deviceSn	String	Y	设备序列号/拼接任务ID/live Id
+channelName	String	N	通道名称（如果不指定则获取实际通道名称）
+startTime	String	N	回放开始时间（格式：yyyyMMddHHmmss）
+endTime	String	N	会放结束时间（格式：yyyyMMddHHmmss）
+wndId	Number	N	播放面板序号，从1开始，在指定面板绑定通道，0代表当前选中面板（默认）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.bindChannel({
  deviceSn: '1001',
  channelName: '通道1',
  startTime: '20220425000000',
  endTime: '20220425235959',
  wndId: 5
}, '1')
```

执行该方法后，依据当前播放器窗口模式的不同（实况预览或录像回放），会直接执行对应的视频拉流
在单分屏多窗口的模式下，不需要指定wndId或直接传0

4.4 showWnd - 显示窗口

参数

参数名称	类型	是否必填	说明
isAll	Boolean	N	是否显示全部窗口 true-显示全部 false-显示当前（默认）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.showWnd(false, '1')
```

若 isAll 为true，则 playerId 不起作用

4.5 hideWnd - 隐藏窗口

参数

参数名称	类型	是否必填	说明
isAll	Boolean	N	是否隐藏全部窗口 true-隐藏全部 false-隐藏当前（默认）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.hideWnd(false, '1')
```

若 isAll 为true，则 playerId 不起作用

当页面路由切换等操作需要隐藏或显示插件窗口时，可调用 `hideWnd` 或 `showWnd`

4.6 setPlayMode - 切换播放模式

参数

参数名称	类型	是否必填	说明
playMode	String	Y	播放模式 preview-实况预览 playback-设备回放 splice-拼接视频回放 rtmp-RTMP推流播放
wndTitle	String	N	播放器标题（不指定时将不显示标题栏）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.setPlayMode('playback', '播放器1-回放', '1')
```

4.7 openDownloadTaskDlg - 打开下载管理器

参数

无

调用示例

```
playerGroup.openDownloadTaskDlg()
```

4.8 uninit - 插件销毁

参数

参数名称	类型	是否必填	说明
isAll	Boolean	N	是否销毁全部窗口 true-销毁全部 false-销毁当前（默认）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

参数名称	类型	是否必填	说明
调用示例			

```
playerGroup.uninit(false, '1')
```

若 isAll 为true, 则 playerId 不起作用

4.9 captureView - 视频抓图

参数

参数名称	类型	是否必填	说明
imageType	Number	Y	图片类型 0-bmp,1-jpg,2-png
playerId	String	N	播放器ID (不指定则对当前选中窗口执行)
调用示例			

```
playerGroup.captureView(2, '1')
```

4.10 setPreviewPullStreamConfig - 设置实时预览默认拉流策略

参数

参数名称	类型	是否必填	说明
singleScreenStream	Number	Y	单分屏拉流默认策略 0-主码流 1-次码流
multiScreenStream	Number	Y	多分屏拉流默认策略 0-主码流 1-次码流

调用示例

```
playerGroup.setPreviewPullStreamConfig({
  singleScreenStream: 0,
  multiScreenStream: 0
})
```

该方法对播放器组中的所有播放器生效

4.11 setImmediatelyPlaybackConfig - 设置即时回放策略

参数

参数名称	类型	是否必填	说明
timePoint	Number	Y	即时回放时间, 单位为分钟, 最小5分钟
location	Number	Y	1-设备录像, 0或其他表示集中录像

调用示例

```
playerGroup.setImmediatelyPlaybackConfig({
  timePoint: 10,
  location: 1
})
```

该方法对播放器组中的所有播放器生效

4.12 setPlaybackConfig - 设置录像回放策略

参数

参数名称	类型	是否必填	说明
fastBack	Number	Y	快退时间 单位为秒，最小15秒，小于15秒自动设置为15秒
fastForward	Number	Y	快进时间 单位为秒，最小15秒，小于15秒自动设置为15秒
type	Number	Y	类型 1-全部 2-正常录像 3-报警录像
location	Number	Y	位置 1-设备录像 0-集中录像

调用示例

```
playerGroup.setPlaybackConfig({ // 录像回放策略
  fastBack: 15, // 快退
  fastForward: 15, // 快进
  type: 1, // 类型 1-全部 2-正常录像 3-报警录像
  location: 1, // 位置 1-设备录像 0-集中录像
})
```

该方法对播放器组中的所有播放器生效

4.13 cuttingPartWindow - 窗口部分区域切除

参数

参数名称	类型	是否必填	说明
param	Object HTMLInputElement	Y	可以为对象类型的坐标参数或用于定位的DOM元素，如果为坐标参数需包含以下值
+left	Number	Y	切除区域在浏览器视窗中的左侧距离
+top	Number	Y	切除区域在浏览器视窗中的上方距离
+width	Number	Y	切除区域宽度
+height	Number	Y	切除区域高度

调用示例

```
// 调用方式1
const rect = document.getElementById('menu').getBoundingClientRect()
playerGroup.cuttingPartWindow({
  left: rect.x,
  top: rect.y,
  width: rect.width,
  height: rect.height
})

// 调用方式2
playerGroup.cuttingPartWindow(document.getElementById('menu'))
```

说明：推荐用于较小区域被插件窗口遮挡的情况，比如弹出的浮动菜单、气泡提示信息等

4.14 repairPartWindow - 清除区域切除

参数

参数名称	类型	是否必填	说明
param	Object HTMLInputElement	N	可以为对象类型的坐标参数或用于定位的DOM元素，如果为坐标参数需包含以下值
+left	Number	Y	切除区域在浏览器视窗中的左侧距离
+top	Number	Y	切除区域在浏览器视窗中的上方距离
+width	Number	Y	切除区域宽度
+height	Number	Y	切除区域高度

传递上述参数代表还原指定区域，不传代表还原所有区域

调用示例

```
// 调用方式1
const rect = document.getElementById('menu').getBoundingClientRect()
playerGroup.repairPartWindow({
  left: rect.x,
  top: rect.y,
  width: rect.width,
  height: rect.height
})

// 调用方式2
playerGroup.repairPartWindow(document.getElementById('menu'))
```

4.15 refreshPosition - 刷新播放器定位

参数

参数名称	类型	是否必填	说明
container	String HTMLElement	N	多个播放器窗口的外部容器
players	Array	N	单个播放器窗口的定位元素
+playerId	String	N	播放器ID
+panel	String HTMLElement	N	单个播放器定位元素，如果不指定则继续使用createPlayer时的panel

调用示例

```
playerGroup.refreshPosition(document.getElementById('container'), [{
  playerId: '1',
  panel: document.getElementById('player1')
},{
  playerId: '2',
  panel: document.getElementById('player2')
}])
```

`container` 的含义与3.3实例化中的container相同
如果不传，则继续使用实例化时传入的执行定位获取
`panel` 的含义与4.1 createPlayer中的panel相同

4.16 preload - 预加载（异步）

由于首次加载视频窗口，需要执行WebSocket连接以及一些初始化步骤，需要的时间较长
所以可以在页面加载完成还不需要弹出视频窗口的时候，先执行该方法
以提高实际显示视频弹窗时响应的速度
(如果从未执行，那么将在首次createPlayer时自动执行)

```
preload(options,topwindow?,iframePosition?)
```

options参数

参数名称	类型	是否必填	说明
layout	Number	Y	分屏数量（多窗口模式下传1即可）

参数名称	类型	是否必填	说明
videoRights	Number	N	视频权限（多种组合直接相加） 空表示拥有所有能力，0-表示没有任何能力 0x0001-高清、标清切换 0x0002-抓图 0x0004-录像 0x0008-声音控制 0x0010-对讲 0x0020-云台控制（弹窗） 0x0040-码流信息显示、隐藏控制 0x0080-原始比例、自适应切换 0x0100-回放下载 0x0200-回放快进、快退 0x0400-设置录像回放查询时间 0x0800-实时预览及时回放 0x1000-实时预览录像回放 0x2000-连接控制（断开、重连、清除）
language	'zh' 'en'	N	插件的语言，默认值 'zh'中文
loadMode	'web' 'app'	N	插件绑定的环境，是浏览器还是客户端，默认为 'web' 浏览器
playbackColor	Object	N	回放时间轴颜色设置
+timeAxisBGColor	Number	Y	录像/及时回放单条时间轴背景颜色
+timeAxisNormalColor	Number	Y	录像/及时回放单条时间轴正常录像颜色
+timeAxisAlarmColor	Number	Y	录像/及时回放单条时间轴告警录像颜色
videoWndConfig	Object	N	视频窗口颜色设置
+wndBGColor	Number	Y	视频窗口背景颜色
+wndBorderColor	Number	Y	视频窗口边框颜色
+selWndBorderColor	Number	Y	选中视频窗口边框颜色
+selWndBorderSize	Number	Y	选中窗口的边框宽度

topWindow参数

说明：在存在跨域的情形下为必传项，务必在取得数据后再调用该方法。参数的取值是iframe最顶层的window下的取值。

参数名称	类型	是否必填	说明
screenX	number	Y	最顶层的window的screenX属性
screenY	number	Y	最顶层的window的screenY属性
outerWidth	number	Y	最顶层的window的outerWidth属性

参数名称	类型	是否必填	说明
outerHeight	number	Y	最顶层的window的outerHeight属性
innerHeight	number	Y	最顶层的window的innerHeight属性
innerWidth	number	Y	最顶层的window的innerWidth属性
screen	number	Y	最顶层的window的screen属性window.screen
+height	number	Y	最顶层的window的screen属性下的height属性
+width	number	Y	最顶层的window的screen属性下的width属性
+availHeight	number	Y	最顶层的window的screen属性下的availHeight属性
+availWidth	number	Y	最顶层的window的screen属性下的availWidth属性
document	number	Y	最顶层的window的document属性
+title	string	Y	最顶层的window的document属性下的title属性

iframePosition参数

说明：在存在跨域的情形下为必传项，务必在取得数据后再调用该方法。参数的取值是iframe最顶层的window下的取值。

参数名称	类型	是否必填	说明
x	number	Y	在最顶层的window的下获取插件所在的iframe的位置横坐标 xxlframe.getBoundingClientRect().left
y	number	Y	在最顶层的window的下获取插件所在的iframe的位置纵坐标 xxlframe.getBoundingClientRect().top

调用示例

```
let topwindow = {
  //..得到的值
}
let iframePosition = {
  //..得到的值
}
playerGroup.preload({
  layout: 1, // 播放面板数量
  videoRights: 0x0001 + 0x0002 + 0x0004 + 0x0008 + 0x0010 + 0x0040 + 0x0100 +
0x0200 + 0x2000, // 视频权限
  playbackColor: { // 回放时间轴颜色设置
    timeAxisBGColor: 0x6a2dd8,
    timeAxisNormalColor: 0x158f58,
    timeAxisAlarmColor: 0x8cc12f
  },
  videowndConfig: { // 视频窗口颜色设置
```



```

        wndBGColor: 0xc1952f,
        wndBorderColor: 0xce368d,
        selWndBorderColor: 0x3697ce,
        selWndBorderSize: 3
    },
    },topWindow,iframePosition).then(result => {
        if(result) {
            console.log('预加载成功')
        } else {
            console.log('已执行过预加载，无需重复执行')
        }
    })
})

```

异步回调获得的返回值是一个布尔值，true代表预加载完毕，false代表已经执行过预加载
 参数当中的 videoRights、 playbackColor、 videowndConfig，如果已在手动调用该方法，
 createPlayer中再传入的将不再起作用
 如果需要设定该参数并且需要调用预加载方法，必须在preload中传入

4.17 getAppInfo - 获取插件下载地址（异步）

参数

无

调用示例

```

playerGroup.getAppInfo().then(response => {
    console.log('下载地址:' + response.data.downloadPath)
})

```

4.18 setPreviewStream - 切换实时预览高清/标清

参数

参数名称	类型	是否必填	说明
param	Object	Y	设定参数
+wndId	Number	N	不填或者为空，表示对所有播放面板有效，否则只针对某一个播放面板
+stream	Number	Y	0-主码流 1-次码流
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```

playerGroup.setPreviewStream({
    wndId: 1,
    stream: 1
}, '1')

```

4.19 setPlaybackSpeed - 设置录像回放的播放速度

参数

参数名称	类型	是否必填	说明
param	Object	Y	设定参数
+wndId	Number	N	不填或者为空，表示对所有播放面板有效，否则只针对某一个播放面板
+speed	String	Y	"1/16X" "1/8X" "1/4X" "1/2X" "1X" "2X" "4X" "8X" "16X", 注意大小写
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```
playerGroup.setPlaybackSpeed({
  wndId: 1,
  speed: '2X'
}, '1')
```

4.20 videoRecord - 视频录像控制

参数

参数名称	类型	是否必填	说明
param	Object	Y	设定参数
+wndId	Number	N	不填或者为空，表示对所有播放面板有效，否则只针对某一个播放面板
+start	Number	Y	1-开始，0-结束
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```
playerGroup.videoRecord({
  wndId: 1,
  start: 1
}, '1')
```

4.21 setVideoPlayerFullScreen - 全屏/退出全屏

参数

参数名称	类型	是否必填	说明
fullScreen	Boolean	Y	true-全屏 false-退出全屏
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```
playerGroup.setVideoPlayerFullScreen(true, '1')
```

4.22 downloadMosaicTask - 下载拼接任务

参数

参数名称	类型	是否必填	说明
tasks	Array	Y	数组
taskId	String	Y	拼接任务ID
taskName	String	N	拼接任务名称

调用示例

```
playerGroup.downloadMosaicTask([[  
  taskId: '1001',  
  taskName: '任务1'  
], [  
  taskId: '1002',  
  taskName: '任务2'  
]])
```

4.23 setPlaybackTimePos - 录像回放跳播

参数

参数名称	类型	是否必填	说明
param	Object	Y	设定参数
+wndId	Number	N	0/不填-当前选中窗口，大于0为指定的窗口
+timePos	String	N	回放结束时间（格式：yyyyMMddHHmmss），时间尽量为回放播放进度返回的开始、结束时间之间的时间
+pastTime	Number	N	单位为ms，拼接回放/按文件播放时使用
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```
playerGroup.setPlaybackTimePos({
  wndId: 0,
  timePos: '20220712000000'
}, '1')
```

4.24 startRecordDownload - 下载回放录像

仅在该窗口为录像回放模式的情况下有效

参数

参数名称	类型	是否必填	说明
param	Object	Y	设定参数
+wndId	Number	N	0/不填-当前选中窗口，大于0为指定的窗口
+startTime	String	Y	下载录像起始时间（格式：yyyyMMddHHmmss）
+interval	Number	Y	下载时长，单位为秒
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```
playerGroup.startRecordDownload({
  wndId: 0,
  startTime: '20230317103000',
  interval: 10
}, '1')
```

5 JavaScript接口 - 云台控制

必须在调用 startPTZControl 获取到云台控制权限之后使用其他功能

5.1 startPTZControl - 开启云台控制

参数

参数名称	类型	是否必填	说明
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.startPTZControl('1')
```

5.2 stopPTZControl - 结束云台控制

参数

参数名称	类型	是否必填	说明
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.stopPTZControl('1')
```

5.3 startPTZMove - 开始云台移动

参数

参数名称	类型	是否必填	说明
action	Number	Y	方向 1-上, 2-下, 3-左, 4-右, 5-左上, 6-左下, 7-右上, 8-右下, 9-自动
speed	Number	Y	移动速度 (0~254)
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.startPTZMove(3, 120, '1')
```

5.4 stopPTZMove - 结束云台移动

参数

参数名称	类型	是否必填	说明
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.stopPTZMove('1')
```

5.5 startPTZZoom - 开始云台变倍

参数

参数名称	类型	是否必填	说明
zoom	Number	Y	变化值 (-254~254)
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.startPTZZoom(-120, '1')
```

5.6 stopPTZZoom - 结束云台变倍

参数

参数名称	类型	是否必填	说明
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.stopPTZZoom('1')
```

5.7 startPTZChange - 开始变焦 光圈改变

参数

参数名称	类型	是否必填	说明
focus	Number	Y	焦距变化值（-254~254）
iris	Number	Y	光圈变化值（-254~254）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.startPTZChange(120, 0, '1')
```

如果只需要改变焦距，光圈变化值传0即可，反之亦然

5.8 stopPTZChange - 结束变焦 光圈改变

参数

参数名称	类型	是否必填	说明
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）
调用示例			

```
playerGroup.stopPTZChange('1')
```

5.9 getPTZPresets - 获取云台预置点（异步）

参数

参数名称	类型	是否必填	说明
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

参数名称	类型	是否必填	说明
调用示例			

```
const presetList = await playerGroup.getPTZPresets('1')
```

5.10 locatePTZPreset - 调用云台预置点（异步）

参数

参数名称	类型	是否必填	说明
presetId	Number	Y	预置点ID
movespeed	Number	Y	移动速度（0~254）
playerId	String	N	播放器ID（不指定则对当前选中窗口执行）

调用示例

```
await playerGroup.locatePTZPreset(2, 120, '1')
```

6 JavaScript接口 - 事件监听

事件监听主要用于接收和获取插件客户端传到web端的消息内容
绑定事件格式统一为

```
playerGroup.on(eventName, callback)
```

解绑事件格式统一为

```
playerGroup.off(eventName, callback)
```

其中 `eventName` 表示事件名称，`callback` 是对应事件的回调函数
以下列出的均为**事件名称**，其中的参数列表表示回调函数的**入参**

注意: 如果需要解绑，传入的callback必须是和绑定时相同的函数对象

6.1 tellCurSelectedWnd - 窗口选中

当点击某个窗口时会执行

参数

参数名称	类型	是否必填	说明
uuid	String	Y	播放器ID（同createPlayer中传入的playerId，或不指定时自动生成的ID）
wndId	Number	Y	播放面板ID（在单分屏多窗口模式下，该值均为1）

参数名称	类型	是否必填	说明
status	String	Y	0-空闲, 1-获取播放地址失败, 2-连接中, 3-连接失败, 4/5/6-预留, 7-断开连接, 8-连接超时, 9-正在播放, 10-播放结束, 11-连接超过上限
channelId	String	N	已关联的通道 (如果窗口未关联通道, 该值为空)

调用示例

```
playerGroup.on('tellCurSelectedwnd', param => {
  console.log('已选中窗口ID:' + param.uuid)
})
```

6.2 tellPluginClosedByUser - 插件被用户手动关闭

当点击单个窗口右上角叉号时会执行

参数

参数名称	类型	是否必填	说明
uuid	String	Y	播放器ID (同createPlayer中传入的playerId, 或不指定时自动生成的ID)

调用示例

```
playerGroup.on('tellPluginClosedByUser', param => {
  console.log('窗口被关闭ID:' + param.uuid)
})
```

6.3 tellPlaybackPlayTimePos - 录像回放播放进度

录像回放播放进度

参数

参数名称	类型	是否必填	说明
uuid	String	Y	播放器ID (同createPlayer中传入的playerId, 或不指定时自动生成的ID)
wndMode	Number	Y	同createPlayer时传入的playMode
wndId	Number	Y	播放面板ID (在单分屏多窗口模式下, 该值均为1)
playTimePos	String	Y	当前播放时间
startTime	String	Y	录像开始时间
endTime	String	Y	录像结束时间

参数名称	类型	是否必填	说明
pastTime	Number	Y	已经播放的进度(用于拼接视频回放)
durationTime	Number	Y	总时长(用于拼接视频回放)

调用示例

```
playerGroup.on('tellPlaybackPlayTimePos', param => {
  console.log('当前进度:' + param.playTimePos)
})
```

6.4 tellVideoStateChanged - 视频状态发生变化

参数

参数名称	类型	是否必填	说明
uuid	String	Y	播放器ID（同createPlayer中传入的playerId，或不指定时自动生成的ID）
wndMode	Number	Y	同createPlayer时传入的playMode
wndId	Number	Y	播放面板ID（在单分屏多窗口模式下，该值均为1）
channelId	String	Y	当前绑定的通道
status	String	Y	0-空闲/清除视频，1-获取播放地址失败，2-连接中，3-连接失败，4/5/6-预留，7-断开连接，8-连接超时，9-正在播放，10-播放结束，11-连接超过上限

调用示例

```
playerGroup.on('tellVideoStateChanged', param => {
  console.log('播放器ID:' + param.uuid)
})
```

6.5 tellPluginClosed - 插件进程退出

参数

参数名称	类型	是否必填	说明
reason	Number	Y	-1未知（多半为插件崩溃） 0-没有保活（超过10分钟未收到web端心跳信号） 1-用户关闭主界面 2-外部发送了disconnect信号 3-websocket连接断开 4-同一页面单插件模式 5-插件服务未启动或未安装

调用示例

```
playerGroup.on('tellPluginClosed', param => {
  console.log('插件进程退出 原因:' + param.reason)
})
```

6.6 tellEnterFullScreen - 监听窗口的进入全屏

参数

参数名称	类型	是否必填	说明
uuid	String	Y	播放器ID（同createPlayer中传入的playerId，或不指定时自动生成的ID）
wndMode	Number	Y	窗口的模式
channelId	String	Y	当前绑定的通道

调用示例

```
playerGroup.on('tellEnterFullScreen', param => {
  console.log('窗口全屏状态发生变化')
})
```

6.7 tellQuitFullScreen - 监听窗口退出全屏

参数

参数名称	类型	是否必填	说明
uuid	String	Y	播放器ID（同createPlayer中传入的playerId，或不指定时自动生成的ID）
wndMode	Number	Y	窗口的模式
channelId	String	Y	当前绑定的通道

调用示例

```
playerGroup.on('tellQuitFullScreen', param => {
  console.log('窗口全屏状态发生变化')
})
```