

云直播
如何播放
产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

如何播放

获取播放地址 URL

- 频道模式获取播放地址

- 直播码模式获取播放地址

- 业务后台生成播放地址

移动端播放

Web 播放

- Web 播放器 TCPlayer Lite

 - TCPlayerLite

 - 常见问题

- Web 直播播放器 1.0

 - Web 直播播放器 1.0

 - 问题排查

如何播放

获取播放地址 URL

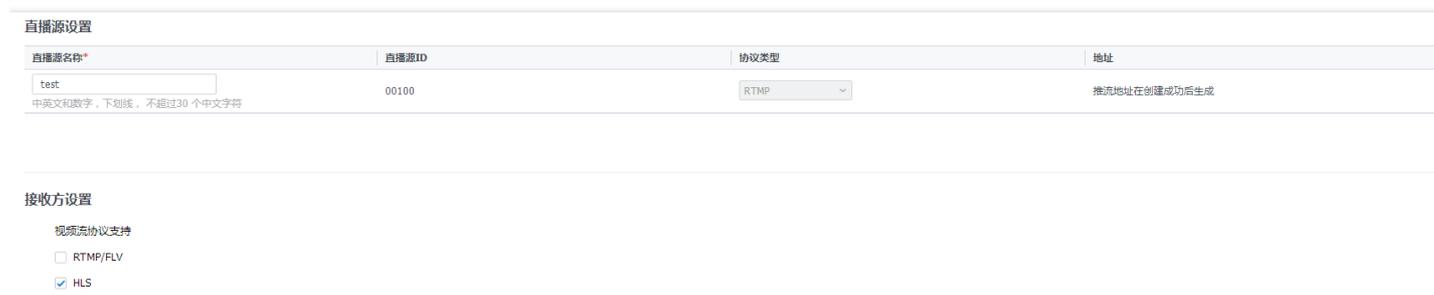
频道模式获取播放地址

最近更新时间：2018-03-29 15:33:44

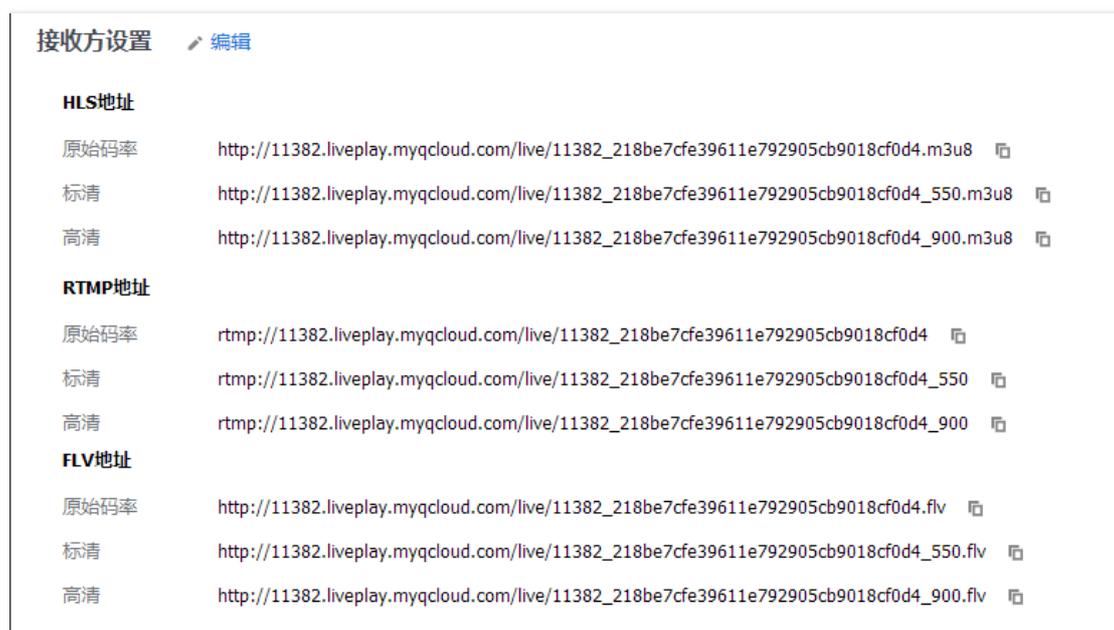
创建一个频道，选择需要的播放码率，如下图：



在接收方设置中，选择需要的播放协议，如下图：



单击【保存】，创建完成后在频道信息中可查看到各种码率和播放协议对应的播放 URL，如下图：



直播码模式获取播放地址

最近更新时间：2018-03-29 15:33:58

进入直播码模式，在推流生成器中输入直播码，可得到对应的推流 ID 和播放 ID。

- 推流生成器是一个推流播放地址生成的工具，仅是一个工具，工具展示出 URL 时该条流并未建立，实际推流成功后才会建立对应的一条流；
- 直播码即一条流的 ID，等同于频道模式的频道概念或者房间的概念。



如需多码率，则获取到播放 URL 后，在 URL 后携带码率后缀，例如：

```
rtmp 协议，标清 ( 550kbps ) 播放地址：rtmp://11382.liveplay.myqcloud.com/live/11382_test_550
rtmp 协议，高清 ( 900kbps ) 播放地址：rtmp://11382.liveplay.myqcloud.com/live/11382_test_900
flv 协议，标清 ( 550kbps ) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_550.flv
flv 协议，高清 ( 900kbps ) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_900.flv
hls 协议，标清 ( 550kbps ) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_550.m3u8
hls 协议，高清 ( 900kbps ) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_900.m3u8
```

业务后台生成播放地址

最近更新时间：2018-11-01 10:55:26

适用于直播码模式

不带防盗链的播放地址

在控制台上通过工具生成地址不够方便，您可根据播放地址的生成逻辑在您的业务后台用代码实现，推流地址和播放地址是有对应关系的，假设推流地址格式：

```
rtmp://bizid.livepush.myqcloud.com/live/bizid_id ?
```

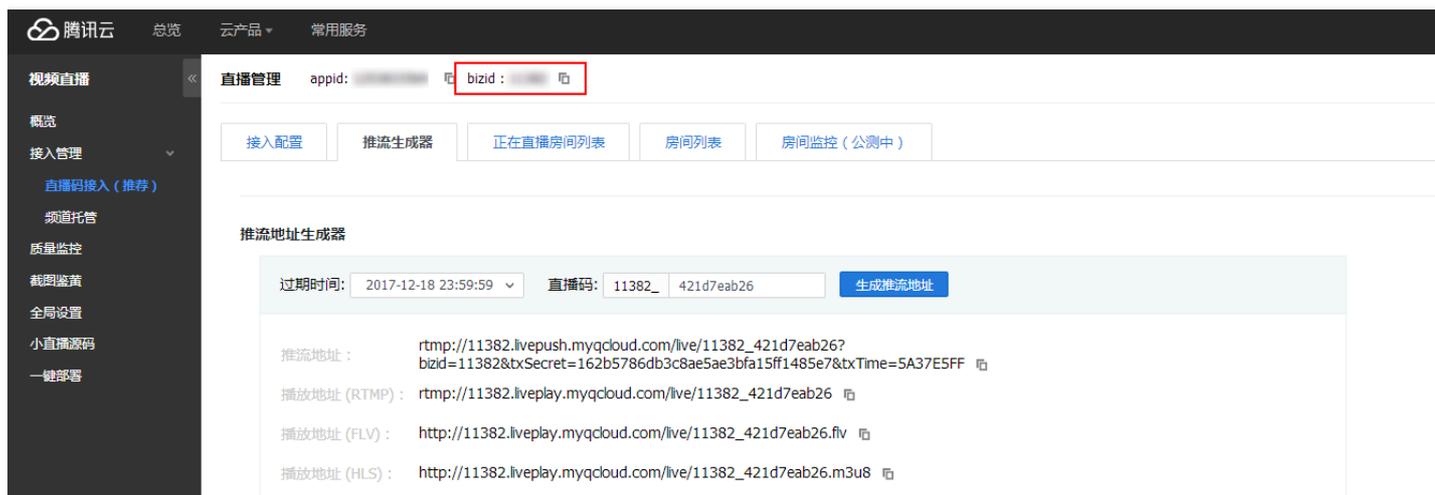
则播放地址格式：

```
rtmp://bizid.liveplay.myqcloud.com/live/bizid_id  
http://bizid.liveplay.myqcloud.com/live/bizid_id.flv  
http://bizid.liveplay.myqcloud.com/live/bizid_id.m3u8
```

例如：帐号对应的 bizid 为 8888，ID 为 test，则播放地址为：

```
http://bizid.liveplay.myqcloud.com/live/bizid_test.flv  
http://bizid.liveplay.myqcloud.com/live/bizid_test.m3u8
```

bizid 可在控制台上查看，ID 为自定义部分，一般由大小英文字母和下划线组成。



腾讯云 总览 云产品 常用服务

视频直播 << 直播管理 appid: [redacted] bizid: [redacted]

接入配置 推流生成器 正在直播房间列表 房间列表 房间监控 (公测中)

推流地址生成器

过期时间: 2017-12-18 23:59:59 直播码: 11382_ 421d7eab26 生成推流地址

推流地址: rtmp://11382.livepush.myqcloud.com/live/11382_421d7eab26?
bizid=11382&txSecret=162b5786db3c8ae5ae3bfa15ff1485e7&txTime=5A37E5FF

播放地址 (RTMP): rtmp://11382.liveplay.myqcloud.com/live/11382_421d7eab26

播放地址 (FLV): http://11382.liveplay.myqcloud.com/live/11382_421d7eab26.flv

播放地址 (HLS): http://11382.liveplay.myqcloud.com/live/11382_421d7eab26.m3u8

多码率播放地址

同上，如需多码率，在 URL 后携带码率后缀，例如：

```
rtmp 协议，标清 (550kbps) 播放地址：rtmp://11382.liveplay.myqcloud.com/live/11382_test_550  
rtmp 协议，高清 (900kbps) 播放地址：rtmp://11382.liveplay.myqcloud.com/live/11382_test_900  
flv 协议，标清 (550kbps) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_550.flv  
flv 协议，高清 (900kbps) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_900.flv  
hls 协议，标清 (550kbps) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_550.m3u8  
hls 协议，高清 (900kbps) 播放地址：http://11382.liveplay.myqcloud.com/live/11382_test_900.m3u8
```

带防盗链的播放地址

携带防盗链的播放地址格式如下：

```
http://bizid.liveplay.myqcloud.com/live/bizid_id.flv?txSecret=xxxx&txTime=xxxx  
http://bizid.liveplay.myqcloud.com/live/bizid_id.m3u8?txSecret=xxxx&txTime=xxxx
```

注：rtmp 暂不支持防盗链。

参数说明：

1. txTime：播放地址的有效截止时间戳，在该时间戳之前请求是合法请求，时间戳之后的请求是非法请求会被拒绝。
2. txSecret：校验串，算法为：txSecret = MD5(KEY+ StreamName + txTime)字母小写

其中

- o txTime：播放地址的有效截止时间戳，同上；
- o StreamName：取的是去掉后缀的文件名，如：

```
rtmp://5000.liveplay.myqcloud.com/live/test2016011415 StreamName 为 test2016011415
http://5000.liveplay.myqcloud.com/live/test2016011415_550.m3u8 StreamName 为 test2016011415_550
http://5000.liveplay.myqcloud.com/live/test2016011415_900.flv StreamName 为 test2016011415_900
```

- o KEY：为用户的密钥，由控制分配，进入【直播码接入】>【接入配置】>【应用信息】即可查看。



配置申请：如需配置播放防盗链，可 [提交工单](#) 或者联系腾讯商务人员，联系电话：4009-100-100。

移动端播放

最近更新时间：2019-01-11 16:44:37

功能快速集成

如果您只需要在现有的 App 里集成直播播放功能，那么按照如下步骤可以快速实现目标

- step1：[开通直播服务](#)
- step2：[下载 RTMP SDK 开发包](#)
- step3：参考对接文档 ([iOS](#) & [Android](#)) 完成接入

完整解决方案

[移动直播](#)是一套移动终端直播解决方案的集合，它以免费源码的形式向您展示：如何利用腾讯云直播（LVB）、点播（VOD）、云通信（IM）和对象存储（COS）等几项服务组合构建出适合您的直播解决方案。

直播延时几秒钟属于正常范围，若对低延时要求较高，可推荐使用[实时音视频](#)

Web 播放

Web 播放器 TCPlayer Lite

TCPlayerLite

最近更新时间：2018-11-28 10:39:40

功能介绍

腾讯云 Web 超级播放器 TCPlayerLite 主要解决在手机浏览器和 PC 浏览器上播放音视频流的问题，使您的视频内容可以不依赖用户安装 App 就可以在朋友圈、微博等社交平台进行传播，本文档适合有一定 Javascript 语言基础的开发人员阅读。

基础知识

对接之前先要了解如下的基础知识，非常有必要：

• 直播和点播

直播是指视频源是实时的，一旦主播停播了，这个地址就已经失去意义了，而且由于是实时直播，所以播放器在播直播视频的时候是没有进度条的。

点播是指视频源是一个服务器上的文件，文件只要没有被提供方删除，就随时可以播放，而且由于整个视频都在服务器上，所以播放器在播点播视频的时候是有进度条的。

• 协议的支持

TCPlayerLite 的视频播放能力本身不是网页代码实现的，而是靠浏览器的支持，所以其兼容性并不像我们想象的那么好，您必须要接受一个事实：**不是所有的手机浏览器都能有符合预期的表现**，有些手机浏览器甚至根本就不支持视频播放。最常见的用于网页直播的视频源地址是以 m3u8 结尾的地址，我们称其为 HLS (HTTP Live Streaming)，这是苹果推出的标准，由于苹果的影响力，目前各手机浏览器产品对这种格式的兼容性最好，但它有个天然的问题，就是延迟比较大，一般是 20-30 秒左右的延迟，在手机浏览器上我们并没有其它选择。

在 PC 上情况会好很多，因为 PC 上的浏览器目前还没有抛弃 flash 控件，而 flash 控件不追求洁癖，支持的视频源格式挺多的，而且各浏览器上的 flash 控件都是 Adobe 自己开发，所以兼容性非常好。

视频协议	用途	示例地址	PC浏览器	移动浏览器
HLS(m3u8)	可用于直播	http://2157.liveplay.myqcloud.com/2157_358535a.m3u8	支持	支持
HLS(m3u8)	可用于点播	http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.m3u8	支持	支持
FLV	可用于直播	http://2157.liveplay.myqcloud.com/2157_358535a.flv	支持	不支持
FLV	可用于点播	http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.flv	支持	不支持
RTMP	只适用直播	rtmp://2157.liveplay.myqcloud.com/live/2157_280d88	支持	不支持
MP4	只适用点播	http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.MP4	支持	支持

对接攻略

Step 1：页面准备工作

在需要播放视频的页面（包括 PC 或 H5）中引入初始化脚本。

```
<script src="//imgcache.qq.com/open/qcloud/video/vcplayer/TcPlayer-2.2.2.js" charset="utf-8"></script>
```

注意：

直接用本地网页是调试不了的，因为腾讯云 Web 播放器处理不了这种情况下的跨域问题。

Step 2：HTML 里放置容器

在需要展示播放器的页面位置加入播放器容器，也就是放一个 div，然后给它取个名字，比如：id_test_video，之后视频的画面都会在这个容器里渲染，容器的大小控制您可以使用 div 的属性进行控制，示例代码如下：

```
<div id="id_test_video" style="width:100%; height:auto;"></div>
```

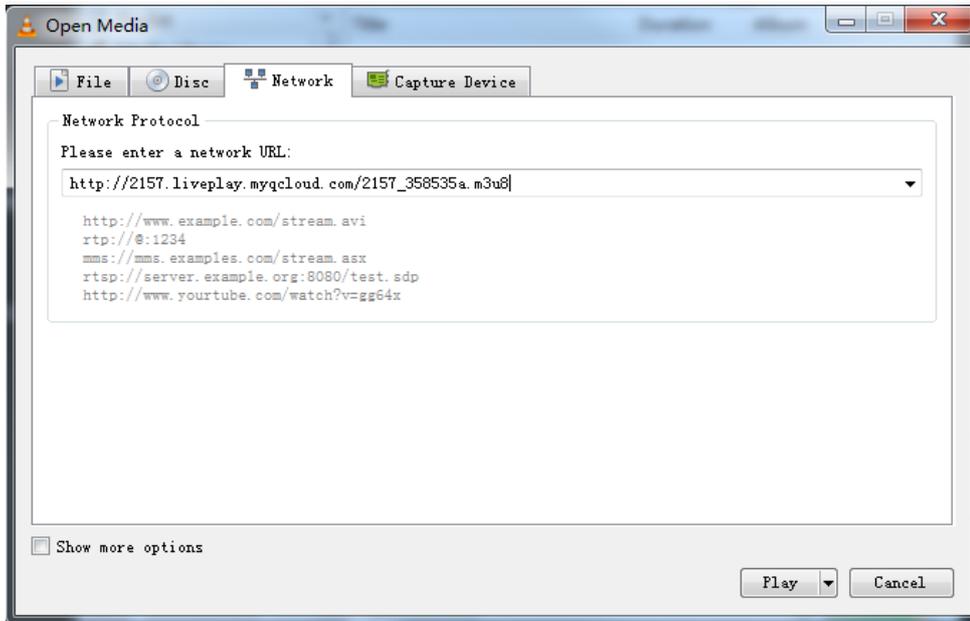
Step 3：对接视频的播放

接下来就要写 javascript 代码，这些代码的作用是指定的 URL 地址拉取音视频流，并将视频画面呈现到 Step2 中添加的容器里。

3.1 一次简单的播放

如下是一条典型的直播 URL 地址，它是 HLS (m3u8) 协议的，如果主播还在直播中的话，那么用 VLC 等播放器是可以直接打开该 URL 进行观看的：

```
http://2157.liveplay.myqcloud.com/2157_358535a.m3u8 // m3u8播放地址
```



我们现在要在手机浏览器上播放这个 URL 的视频，javascript 代码如下：

```
var player = new TcPlayer('id_test_video',{
  "m3u8": "http://2157.liveplay.myqcloud.com/2157_358535a.m3u8", //请替换成实际可用的播放地址
  "autoplay": true, //iOS下safari浏览器，以及大部分移动端浏览器是不开放视频自动播放这个能力的
  "coverpic": "http://www.test.com/myimage.jpg",
  "width": '480', //视频的显示宽度，请尽量使用视频分辨率宽度
  "height": '320' //视频的显示高度，请尽量使用视频分辨率高度
});
```

这段代码就可以支持在 PC 以及手机浏览器上播放 HLS (m3u8) 协议的直播视频，不过 HLS (m3u8) 协议的视频虽然兼容性还不错（部分 Android 手机依然不支持），但其延迟非常高，大约 20 秒以上的延迟。

3.2 PC 上实现更低的延迟

PC 浏览器支持 flash，javascript 代码如下：

```
var player = new TcPlayer('id_test_video', {
  "m3u8": "http://2157.liveplay.myqcloud.com/2157_358535a.m3u8",
  "flv": "http://2157.liveplay.myqcloud.com/live/2157_358535a.flv", //增加了一个flv的播放地址, 用于PC平台的播放 请替换成实际可用的播放地址
  "autoplay": true, //iOS下safari浏览器, 以及大部分移动端浏览器是不开放视频自动播放这个能力的
  "coverpic": "http://www.test.com/myimage.jpg",
  "width": '480', //视频的显示宽度, 请尽量使用视频分辨率宽度
  "height": '320' //视频的显示高度, 请尽量使用视频分辨率高度
});
```

跟前一段代码的区别就在于我们增加了一个 FLV 的播放地址, 腾讯云 Web 播放器如果发现目前的浏览器是 PC 浏览器, 会主动选择 FLV 链路, 可以实现更低的延迟。前提条件是 FLV 和 HLS (m3u8) 这两个地址都是可以出流的, 如果您使用腾讯云的直播服务, 这个问题是不需要发愁, 因为腾讯云的每一条直播频道都默认支持 FLV、RTMP 和 HLS (m3u8) 三种播放协议。

3.3 播放不了怎么办？

如果您发现视频不能播放, 有以下几个原因：

• 原因一：视频源有问题

如果是直播 URL, 那一定要检查一下是不是主播已经停止推流, 状态不处于【直播中】的情况, 可以用浮窗提示一下观众：【主播已经离开】。
如果是点播 URL, 那要检查您要播放的文件是否还在服务器上, 比如要播放的地址是否已经被其它人从点播系统移除。

• 原因二：本地网页调试

目前 TCPlayerLite 是不支持本地网页去调试的 (即通过 file:// 协议来打开视频播放的网页), 主要是浏览器有跨域安全限制, 所以您如果是在 Windows 上随便放一个 test.html 文件然后测试是肯定播放不了的, 需要将其上传到一个服务器上进行测试。前端工程师可以通过反向代理的方式对线上的页面进行本地代理以实现本地调试, 这是主流可行的本地调试办法。

• 原因三：手机兼容问题

FLV 和 RTMP 这两种地址, 在普通的手机浏览器上都是不支持的 (最新版本的 QQ 浏览器支持 FLV 协议的播放, 但普及度还不高), 只能用 HLS (m3u8)。

• 原因四：跨域安全问题

PC 浏览器的视频播放是基于 Flash 控件实现的, 但做过 Flash 开发的同学都知道 **Flash 控件会做跨域访问检查**, 当您要播放的视频所存放的服务器没有部署跨域策略时才会碰到这个问题, 解决方法就是：在视频服务器的根域名下的跨域配置文件 crossdomain.xml 中增加 qq.com 域名：

```
<cross-domain-policy>
<allow-access-from domain="*.qq.com" secure="false"/>
</cross-domain-policy>
```

Step 4：给播放器设置封面

在前面的代码例子中, 您应该注意到 "coverpic" 这个参数了, 在这里将详细介绍这个属性的使用方法。

备注：封面功能有可能在部分移动端播放环境下失效, 由于移动端视频播放环境相对 PC 来说比较复杂, 各个浏览器和 App 的 Webview 对 H5 video 实现的方式并不统一, 如果遇到功能失效的情况, 欢迎向我们的技术支持反馈 (反馈内容包括系统、浏览器或 App 的版本等关键信息), 我们将尽可能去支持。

4.1 简单设置封面

coverpic 可以传入一个图片地址作为播放器的封面, 将在播放器区域内居中并且以图片的实际分辨率进行显示。

```
"coverpic": "http://www.test.com/myimage.jpg"
```

4.2 设置封面的展现样式

coverpic 同时支持传入一个对象, 对象中可以进行设置封面的展现样式 style 和图片地址 src。

style 支持的样式有 3 种：

- default：居中并且以图片的实际分辨率进行显示；
- stretch：拉伸铺满播放器区域, 图片可能会变形；
- cover：优先横向等比拉伸铺满播放器区域, 图片某些部分可能无法显示在区域内。

```
"coverpic": {"style": "stretch", "src": "http://www.test.com/myimage.jpg"}
```

4.3 实现用例

这里有一个线上的示例代码，里面使用了 cover 方式显示封面，在 PC 浏览器中右键【查看页面源码】即可查看页面的代码实现：

```
http://imgcache.qq.com/open/qcloud/video/vcplayer/demo/tcplayer-cover.html
```

注意：

在某些移动端设置封面会无效，具体说明请查看常见问题。

Step 5：多清晰度的支持

5.1 原理介绍

我们知道优酷、土豆、腾讯上的视频有些是有多清晰度选择的，这个效果如何实现呢？



播放器本身是没有能力去改变视频的清晰度的，在视频源产生的地方其实只有一种清晰度，我们称之为原画，而原画视频编码格式和封装格式有很多种，在 Web 端无法完全支持播放所有的视频格式，点播支持必须是以 H.264 为视频编码，以 mp4、FLV 为封装格式的视频。

那么多清晰度是怎么实现的呢？这里就是视频云发挥作用的地方了：

- 对于直播，来自主播那一端的原始视频会在腾讯云进行实时的转码，分出多路转码后的视频，比如我们常说的高清-HD 以及标清-SD，每一路视频都有其对应的地址：

```
http://2157.liveplay.myqcloud.com/2157_358535a.m3u8 // 原画  
http://2157.liveplay.myqcloud.com/2157_358535a_900.m3u8 // 高清  
http://2157.liveplay.myqcloud.com/2157_358535a_550.m3u8 // 标清
```

- 对于点播，一个视频文件上传到腾讯云以后，您可以操作对该视频文件进行转码，产生另外几种清晰度的视频，比如我们常说的高清-HD 以及标清-SD，需要注意的是上传后的原始视频是未经过腾讯云转码的，不能直接用于播放。

```
http://200002949.vod.myqcloud.com/200002949_b6ffc.f240.m3u8 // 原画，用转码后的超清替换  
http://200002949.vod.myqcloud.com/200002949_b6ffc.f230.av.m3u8 // 高清  
http://200002949.vod.myqcloud.com/200002949_b6ffc.f220.av.m3u8 // 标清
```

注意：

上传后的原始视频是未经过腾讯云转码的，不能直接用于播放。

5.2 代码实现

如下的代码是让播放器支持多种清晰度的支持，也就是在播放器的用户界面上展示多种清晰度线路的选择。

```
var player = new TcPlayer('id_test_video', {
  "m3u8": "http://200002949.vod.myqcloud.com/200002949_b6ffc.f240.m3u8", //请替换成实际可用的播放地址
  "m3u8_hd": "http://200002949.vod.myqcloud.com/200002949_b6ffc.f230.av.m3u8",
  "m3u8_sd": "http://200002949.vod.myqcloud.com/200002949_b6ffc.f220.av.m3u8",
  "autoplay": true, //iOS下safari浏览器，以及大部分移动端浏览器是不开放视频自动播放这个能力的
  "coverpic": "http://www.test.com/myimage.jpg",
});
```

5.3 实现用例

这里有一个线上的示例代码，里面使用了多种分辨率的设置以及切换功能，在 PC 浏览器中右键【查看页面源码】即可查看页面的代码实现：

<http://imgcache.qq.com/open/qcloud/video/vcplayer/demo/tcplayer-clarity.html?autoplay=true>

正常情况将看到这样的效果：



PC 端现已支持多种清晰度播放并支持切换的功能，移动端尚未支持。

Step 6：定制错误提示语

我们默认的提示语您可能觉得不符合您的需求，比如“网络错误，请检查网络配置或者播放链接是否正确”或者“视频解码错误”等，我们担心这些提示语在您看来可能太干瘪了，所以腾讯云Web播放器将支持提示语定制：

6.1 代码实现

如下是让播放器支持自定义提示语的核心代码，设置的提示语主要落在 wording 属性上。

```
var player = new TcPlayer('id_test_video', {
  "m3u8": "http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.m3u8", //请替换成实际可用的播放地址
  "autoplay": true, //iOS下safari浏览器是不开放这个能力的
  "coverpic": "http://www.test.com/myimage.jpg",
  "wording": {
    2032: "请求视频失败，请检查网络",
    2048: "请求m3u8文件失败，可能是网络错误或者跨域问题"
  }
});
```

6.2 实现用例

这里有一个线上的示例代码，该示例将播放失败，同时使用了自定义提示文案的功能，在 PC 浏览器中右键【查看页面源码】即可查看页面的代码实现：

http://imgcache.qq.com/open/qcloud/video/vcplayer/demo/tcplayer.html?m3u8=http://2527.vod.myqcloud.com/2527_b393eb1.f230.av.m3u8

6.3 错误码对照表

Code	提示语	说明
1	网络错误，请检查网络配置或者播放链接是否正确	(H5 提示的错误)
2	网络错误，请检查网络配置或者播放链接是否正确	视频格式 Web 播放器无法解码 (H5 提示的错误)
3	视频解码错误	(H5 提示的错误)
4	当前系统环境不支持播放该视频格式	(H5 提示的错误)
5	当前系统环境不支持播放该视频格式	播放器判断当前浏览器环境不支持播放传入的视频，可能是当前浏览器不支持 MSE 或者 Flash 插件未启用
10	请勿在 file 协议下使用播放器，可能会导致视频无法播放	
11	使用参数有误，请检查播放器调用代码	
12	请填写视频播放地址	
13	直播已结束，请稍后再来	RTMP 正常播放过程中触发事件 (NetConnection.Connect.Closed) (Flash 提示的错误)
1001	网络错误，请检查网络配置或者播放链接是否正确	网络已断开 (NetConnection.Connect.Closed) (Flash 提示的错误)
1002	获取视频失败，请检查播放链接是否有效	拉取播放文件失败 (NetStream.Play.StreamNotFound)，可能是服务器错误或者视频文件不存在 (Flash 提示的错误)
2032	获取视频失败，请检查播放链接是否有效	(Flash 提示的错误)
2048	无法加载视频文件，跨域访问被拒绝	请求 m3u8 文件失败，可能是网络错误或者跨域问题 (Flash 提示的错误)

备注：

Code 1 ~ 4 对应的是 H5 的原生事件；

由于 Flash 的黑盒特性以及 H5 视频播放标准的不确定性，错误提示语会不定期更新。

源码参考

这里有一个线上的示例代码，在 PC 浏览器中右键【查看页面源码】即可查看页面的代码实现：

<http://imgcache.qq.com/open/qcloud/video/vcplayer/demo/tcplayer.html?autoplay=true>

您可以用它来测试播放器的效果，在链接后面加上需要播放的视频地址，刷新后就会播放这个视频地址：

<http://imgcache.qq.com/open/qcloud/video/vcplayer/demo/tcplayer.html?autoplay=true&m3u8=http://1251132611.vod2.myqcloud.com/4126d3evodtransgp1251132611/8a592f8b9031868222950257296/f0.f240.m3u8>

参数列表

下面列出了播放器支持的所有参数，并进行了详细的说明。

参数	类型	默认值	参数说明
m3u8	String	无	原画 m3u8 播放 URL 示例：http://2157.liveplay.myqcloud.com/2157_358535a.m3u8
m3u8_hd	String	无	高清 m3u8 播放URL 示例：http://2157.liveplay.myqcloud.com/2157_358535ahd.m3u8
m3u8_sd	String	无	标清 m3u8 播放URL 示例：http://2157.liveplay.myqcloud.com/2157_358535asd.m3u8
flv	String	无	原画 flv 播放 URL 示例：http://2157.liveplay.myqcloud.com/2157_358535a.flv
flv_hd	String	无	高清 flv 播放 URL 示例：http://2157.liveplay.myqcloud.com/2157_358535ahd.flv
flv_sd	String	无	标清 flv 播放 URL 示例：http://2157.liveplay.myqcloud.com/2157_358535asd.flv
mp4	String	无	原画 mp4 播放 URL 示例：http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.mp4
mp4_hd	String	无	高清 mp4 播放 URL 示例：http://200002949.vod.myqcloud.com/200002949_b6ffc.f40.mp4
mp4_sd	String	无	标清 mp4 播放 URL 示例：http://200002949.vod.myqcloud.com/200002949_b6ffc.f20.mp4
rtmp	String	无	原画 rtmp 播放 URL 示例：rtmp://2157.liveplay.myqcloud.com/live/2157_280d88
rtmp_hd	String	无	高清 rtmp 播放 URL 示例：rtmp://2157.liveplay.myqcloud.com/live/2157_280d88hd
rtmp_sd	String	无	标清 rtmp 播放 URL 示例：rtmp://2157.liveplay.myqcloud.com/live/2157_280d88sd
width	Number	无	必选 ，设置播放器宽度，单位为像素 示例：640
height	Number	无	必选 ，设置播放器高度，单位为像素 示例：480
volume	Number	0.5	设置初始音量，范围：0~1 [v2.2.0+] 示例：0.6
live	Boolean	false	必选 ，设置视频是否为直播类型，将决定是否渲染时间轴等控件，以及区分点播直播的处理逻辑 示例：true
autoplay	Boolean	false	是否自动播放 备注：该选项只对大部分 PC 平台生效 示例：true
coverpic	String / Object	无	预览封面，可以传入一个图片地址或者一个包含图片地址 src 和显示样式 style 的对象。 style 可选属性： default 居中 1 : 1 显示 stretch 拉伸铺满播放器区域，图片可能会变形 cover 优先横向等比拉伸铺满播放器区域，图片某些部分可能无法显示在区域内 示例："http://www.test.com/myimage.jpg" 或者 { "style": "cover", "src": http://www.test.com/myimage.jpg }
controls	String	"default"	default 显示默认控件，none 不显示控件，system 移动端显示系统控件 备注：如果需要在移动端使用系统全屏，就需要设置为 system。默认全屏方案是使用 Fullscreen API + 伪全屏的方式 例子 示例："system"

参数	类型	默认值	参数说明
systemFullscreen	Boolean	false	开启后，在不支持 Fullscreen API 的浏览器环境下，尝试使用浏览器提供的 webkitEnterFullScreen 方法进行全屏，如果支持，将进入系统全屏，控件为系统控件 示例：true
flash	Boolean	true	是否优先使用 flash 播放视频， 备注：该选项只对 PC 平台生效 [v2.2.0+] 示例：true
flashUrl	String	无	可以设置 flash swf url 备注：该选项只对 PC 平台生效 [v2.2.1+]
h5_flv	Boolean	false	是否启用 flv.js 的播放 flv。启用时播放器将在支持 MSE 的浏览器下，采用 flv.js 播放 flv，然而并不是所有支持 MSE 的浏览器都可以使用 flv.js，所以播放器不会默认开启这个属性。[v2.2.0+] 示例：true
x5_player	Boolean	false	是否启用 TBS 的播放 flv 或 hls。启用时播放器将在 TBS 模式下(例如 Android 的微信、QQ浏览器)将 flv 或 hls 播放地址直接赋给 <video> 播放。 TBS 视频能力 [v2.2.0+] 示例：true
x5_type	String	无	通过 video 属性 "x5-video-player-type" 声明启用同层 H5 播放器，支持的值：H5 (该属性为 TBS 内核实验性属性，非 TBS 内核不支持)， TBS H5 同层播放器接入规范 示例："h5"
x5_fullscreen	String	无	通过 video 属性 "x5-video-player-fullscreen" 声明视频播放时是否进入到 TBS 的全屏模式，支持的值：true (该属性为 TBS 内核实验性属性，非 TBS 内核不支持)。 示例："true"
x5_orientation	Number	无	通过 video 属性 "x5-video-orientation" 声明 TBS 播放器支持的方向，可选值：0 (landscape 横屏)，1：(portrait 竖屏)，2：(landscape portrait 跟随手机自动旋转)。(该属性为 TBS 内核实验性属性，非 TBS 内核不支持) [v2.2.0+] 示例：0
wording	Object	无	自定义文案 示例：{ 2032: '请求视频失败，请检查网络'}
clarity	String	'od'	默认播放清晰度 [v2.2.1+] 示例：clarity: 'od'
clarityLabel	Object	{od: '超清', hd: '高清', sd: '标清'}	自定义清晰度文案 [v2.2.1+] 示例：clarityLabel: {od: '蓝光', hd: '高清', sd: '标清'}
listener	Function	无	事件监听回调函数，回调函数将传入一个 JSON 格式的对象 示例：function(msg){ //进行事件处理 }

实例方法列表

下面列出了播放器实例支持的方法：

方法	参数	返回值	说明	示例
play()	无	无	开始播放视频	player.play()
pause()	无	无	暂停播放视频	player.pause()
togglePlay()	无	无	切换视频播放状态	player.togglePlay()
mute(muted)	{Boolean} [可选]	true,false {Boolean}	切换静音状态，不传参则返回当前是否静音	player.mute(true)

方法	参数	返回值	说明	示例
volume(val)	{int} 范围：0~1 [可选]	范围：0~1	设置音量，不传参则返回当前音量	player.volume(0.3)
playing()	无	true,false {Boolean}	返回是否在播放中	player.playing()
duration()	无	{int}	获取视频时长 备注：只适用于点播，需要在触发loadedmetadata事件后才可获取视频时长	player.duration()
currentTime(time)	{int} [可选]	{int}	设置视频播放时间点，不传参则返回当前播放时间点 备注：只适用于点播	player.currentTime()
fullscreen(enter)	{Boolean} [可选]	true,false {Boolean}	调用全屏接口 (Fullscreen API)，不支持全屏接口时使用伪全屏模式，不传参则返回值当前是否全屏 备注：移动端系统全屏没有提供API，也无法获取系统全屏状态	player.fullscreen(true)
buffered()	无	0~1	获取视频缓冲数据百分比 备注：只适用于点播	player.buffered()
destroy()	无	无	销毁播放器实例 [v2.2.1+]	player.destroy()
switchClarity()	{String} [必选]	无	切换清晰度，传值 "od"、"hd"、"sd" [v2.2.1+]	player.switchClarity('od')
load(url)	{String} [必选]	无	通过视频地址加载视频 备注：该方法只能加载对应播放模式下支持的视频格式，Flash 模式支持切换 rtmp、flv、hls、mp4，H5 模式支持 mp4、hls、flv (hls、flv 取决于浏览器是否支持) [v2.2.2+]	player.load(http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.mp4)

注意：

以上方法必须是 Tcplayer 的实例化对象，且需要初始化完毕才可以调用（即 load 事件触发后）。

进阶攻略

这里介绍一些视频播放器 SDK 的进阶使用方法。

使用广告SDK

TcPlayerLite 提供了集成 IMA SDK 的版本，若需使用广告功能，需在页面中引入以下代码

```
<!-- Google IMA SDK -->
<script type="text/javascript" src="//imasdk.googleapis.com/js/sdkloader/ima3.js"></script>
<!-- 使用集成 IMA SDK 的版本 -->
<script type="text/javascript" src="//restcplayer.qcloud.com/sdk/tcplayer-web-1.0.1.js"></script>
```

通过 adTagUrl 和 auth 参数使用广告功能，帐号及 License 信息可登录 <https://tcplayer.qcloud.com> 注册申请，或联系 tcplayer@tencent.com 咨询反馈。

```
var player = new TcPlayer('id_test_video', {
  /* Advertisement-related parameter */
  "adTagUrl": "http://ad_tag_url", //VAST,VMAP,VAPID 视频广告 Tag
  "auth": {
    "user_id": "your_user_id", //广告帐户 ID
    "app_id": "your_app_id", //应用 ID
    "license": "your_license" //应用 license
  }
});
```

注意：

TcPlayer 2.2.0 之后的文档描述不适用于集成 IMA SDK 的版本，tcplayer-web-1.0.1 为独立的分支

ES Module

TcPlayerLite 提供了 ES Module 版本，module name 为 TcPlayer，下载地址：

<http://imgcache.qq.com/open/qcloud/video/vcplayer/TcPlayer-module-2.2.2.js>

开启优先 H5 播放模式

TcPlayerLite 是采用 H5 `<video>` 和 Flash 相结合的方式来来进行视频播放的，在不同的播放环境中，播放器会选择默认最合适的播放方案。

虽然目前浏览器厂商已经开始逐步放弃对 Flash 插件的支持，但是在国内仍有大量的浏览器不支持 MSE，在播放 FLV HLS (m3u8) 时无法切换到 H5 `<video>` 的播放模式，而播放 RTMP 必须使用 Flash 模式才可以播放，因此 TcPlayerLite 仍是默认优先启用 Flash 播放模式，如果在检测到 Flash 插件不可用的情况下，将采用 H5 `<video>` 进行播放。

默认 Flash 模式的原因是 Flash 支持的视频格式最广，而 H5 `<video>` 默认只支持 MP4 (h.264) (其他非腾讯云提供的视频格式这里就不列出)，在特定条件下才能支持HLS (m3u8)、FLV。

从 2.2.0 版本开始，提供了可以设置播放模式优先级的属性，如果想优先采用 H5 `<video>` 播放模式，需要把 Flash 属性设置为 false，这样在播放时默认将启用 H5 `<video>` 播放，如果 H5 `<video>` 不可用将采用 Flash 播放，没有检测到 Flash 插件将会提示“当前系统环境不支持播放该视频格式”。

监听事件

TcPlayerLite 是采用 H5 `<video>` 和 Flash 相结合的方式来来进行视频播放，由于两种方式播放视频时触发的事件不尽相同，所以我们以 H5 `<video>` 的规范为准，对 Flash 的播放事件做了一定程度的转换，以实现播放事件命名的统一，TcPlayer 对这两种播放方式所触发的原生事件进行了捕获和透传。

[H5 事件参考列表](#)

[Flash 事件参考列表](#)

统一后的事件列表：

```
error
timeupdate
```

```
load
loadedmetadata
loadeddata
progress
fullscreen
play
playing
pause
ended
seeking
seeked
resize
volumechange
```

注意：

如果通过系统控制栏进行全屏，将无法监听到 fullscreen 事件。

Flash 模式下特有的事件：netStatus

备注：由于 Flash 的黑盒特性以及 H5 视频播放标准在各个平台终端的实现不一致性，事件的触发方式和结果会有差异，开发过程中可以留意这些差异。

在非自动播放的条件下，加载视频至待播放状态，移动端和 PC Flash 触发的事件区别。

移动端：

```
Object {type: "load", src: H5Video, ts: 0, detail: Object}
Object {type: "resize", src: H5Video, ts: 1150.5800000000002}
Object {type: "loadedmetadata", src: H5Video, ts: 1150.5850000000003}
Object {type: "volumechange", src: H5Video, ts: 1156.19}
Object {type: "seeking", src: H5Video, ts: 1168.665}
Object {type: "timeupdate", src: H5Video, ts: 1256.8400000000001}
Object {type: "seeked", src: H5Video, ts: 1256.85}
Object {type: "loadeddata", src: H5Video, ts: 1256.865}
Object {type: "timeupdate", src: H5Video, ts: 1256.9}
Object {type: "progress", src: H5Video, ts: 1408.7800000000002}
```

PC Flash：

```
Object {type: "load", src: FlashVideo, ts: 27, detail: Object}
Object {type: "loadedmetadata", src: FlashVideo, ts: 166, detail: Object}
Object {type: "volumechange", src: FlashVideo, ts: 184}
Object {type: "progress", src: FlashVideo, ts: 1741}
```

备注：以上是两种平台的差异，然而在移动端的各种设备和 App 之间同样存在差异。

应用案例：通过事件监听，可以进行播放失败重连，[在线例子链接](#)。

更新日志

TCPlayerLite 在不断的更新以及完善中，为了方便大家了解版本情况，下面列出的是 TCPlayerLite 发布的主版本介绍，而一些历史 bug 修复，小功能版本没有列出。

日期	版本	更新内容
2016.12.28	2.0.0	首个版本

日期	版本	更新内容
2017.3.4	2.1.0	至 2017.6.30, 经历数次的迭代开发逐步趋于稳定, 目前文档的功能描述中, 如果没有特殊说明, 皆基于此版本。
2017.6.30	2.2.0	<ol style="list-style-type: none"> 1. 增加控制播放环境判断的参数: Flash、h5_flv、x5_player; 2. 调整播放器初始化逻辑, 优化错误提示效果; 3. 增加 flv.js 支持, 在符合条件的情况下可以采用 flv.js 播放 flv; 4. 支持 x5-video-orientation 属性; 5. 增加播放环境判断逻辑, 可通过参数调整 H5 与 Flash 的优先级, 以及是否启用 TBS 播放; 6. 启用版本号发布方式, 避免影响旧版本的使用者; 7. 优化事件触发的时间戳, 统一为标准时间; 8. bug 修复。
2017.12.7	2.2.1	<ol style="list-style-type: none"> 1. 增加 systemFullscreen 参数; 2. 增加 flashUrl 参数; 3. 修复音量 max 后进行静音切换的 UI 问题; 4. 修复 ios11 微信下需要单击两次才能播放的问题; 5. 修复 safari 11 系统样式被遮挡的问题; 6. 适配在 x5 内核会触发 seeking, 但不会触发 seeked 的情况; 7. 修复进度条拖拽到起始位置, 设置 currentTime 失败的问题; 8. 切换清晰度保持音量不变; 9. 修复页面宽度为 0, 播放器宽度判断失败问题; 10. destroy 方法增加完全销毁播放器节点。
2017.12.20	2.2.1	<ol style="list-style-type: none"> 1. 增加可配置清晰度文案功能; 2. 设置默认清晰度; 3. 支持切换清晰度方法。
2018.5.3	2.2.2	<ol style="list-style-type: none"> 1. 优化 loading 组件; 2. 优化 Flash destroy 方法; 3. 默认使用 H5 播放。 4. 修复已知问题

常见问题

最近更新时间：2018-09-20 11:27:36

TcPlayer 支持哪些浏览器？

TcPlayer 经过测试并支持的浏览器有，桌面端：IE10+、Edge、Chrome、Firefox、QQ 浏览器，MAC Safari，移动端：Android4.4+，iOS8.0+，微信、手机 QQ、QQ 浏览器、Chrome、Safari。

如需支持 IE8、IE9 浏览器需要在引入播放器脚本前，引入 Polyfill 脚本，如下所示：

```
<!--[if lt IE 9]>
<script src="//imgcache.qq.com/open/qcloud/video/vcplayer/libs/es5-shim.js" charset="utf-8"></script>
<script src="//imgcache.qq.com/open/qcloud/video/vcplayer/libs/es5-sham.js" charset="utf-8"></script>
<![endif]-->
<script src="//imgcache.qq.com/open/qcloud/video/vcplayer/TcPlayer-2.2.1.js" charset="utf-8"></script>
```

为什么在移动端 TcPlayer 全屏后无法使视频全屏，仍然会显示浏览器的界面？

首先要了解一些背景知识，TcPlayer 提供的全屏方案是 Fullscreen API + 伪全屏的方式，如果浏览器支持 Fullscreen API 那么进入全屏时，视频容器将会铺满屏幕，控制栏是 TcPlayer 提供的控制栏，如图所示：



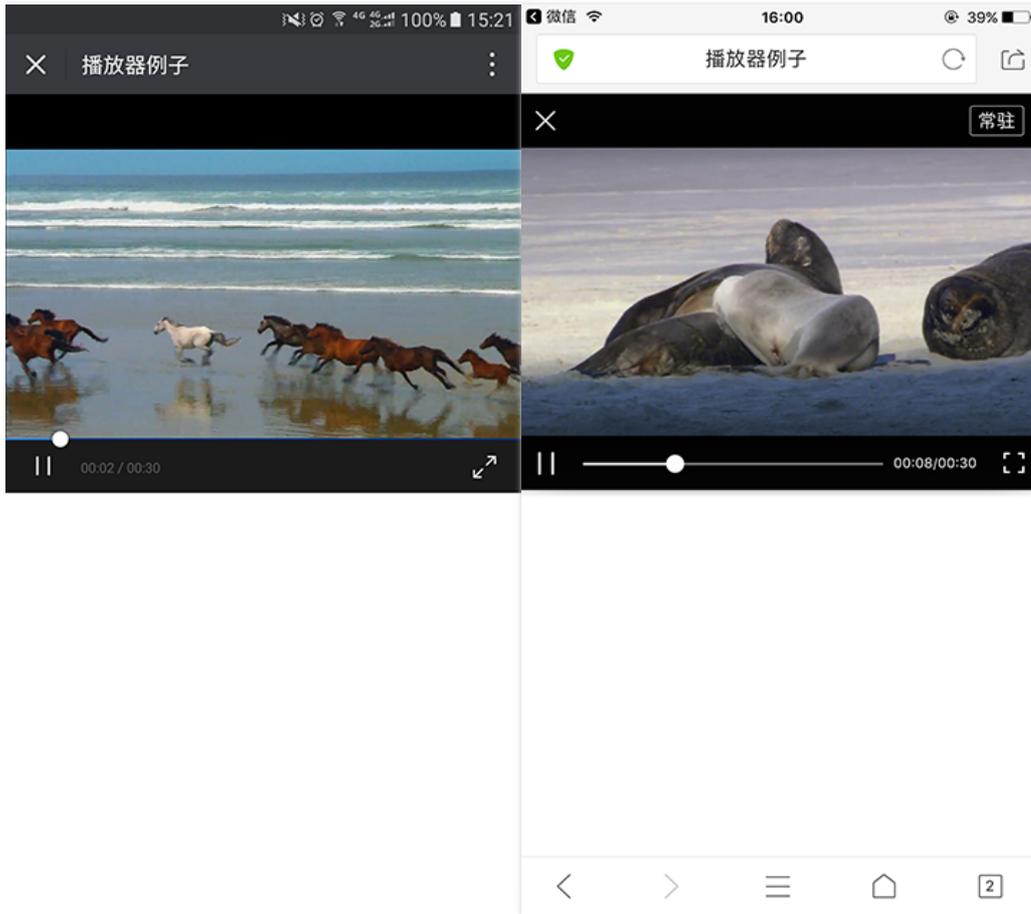
(Android Chrome)

如果浏览器不支持 Fullscreen API 则会进入伪全屏模式，如图所示：



(左：Android 微信，右：iOS 微信)

这两种全屏模式显示的控制栏都是 TcPlayer 提供的控制栏，进入方式都是通过单击控制栏的全屏按钮或者 TcPlayer 提供的全屏方法实现的，然而在移动端不一定可以显示 TcPlayer 提供的控制栏，大部分情况下移动端 webview 会劫持视频播放，并使用 webview 提供的控制栏，这样就无法显示 TcPlayer 的控制栏，也无法使用 TcPlayer 提供的全屏方案，如图所示：



(左：Android 微信，TBS 劫持视频播放，右：iOS QQ 浏览器劫持视频播放)
进入全屏后：



(左：Android 微信，右：iOS QQ浏览器)

可以看到控制栏与 TcPlayer 是不同的，而这种全屏方式是没有接口提供给 JS 调用的，所以 TcPlayer 无法实现这样的全屏方式。我们通常会把视频区域铺满屏幕的全屏模式称为系统全屏，而视频区域铺满浏览器的页面可视区域称为伪全屏，综上所述如果全屏后看到浏览器的界面则是进入了伪全屏，如果想在移动端全屏时必须进入系统全屏，只能使用系统提供的控制栏，选择控制栏类型可以通过 TcPlayer 的 controls 属性进行设置。

为什么 H5 播放视频拉伸变形了？

H5 并不具备拉伸视频的能力，请检查播放器的容器宽高是否设置正确。

为什么我自己的 div 无法在盖在视频上？

不同浏览器对于 `<video>` 标签的实现方案不同，比如您的网页如果是从 QQ 或者微信里打开的（这里说的是 Android 系统下），那么极高的概率会使用 QQ 或微信捆绑的 X5 浏览器内核，也就是 QQ 浏览器内核，该团队考虑当时处于某些原因的考虑，采用了“视频 `<video>` 标签一定要处于最上层”的实现方案（相关信息参考 [QQ 浏览器文档说明](#)），不过最近通过公司内部的各种协调，QQ 浏览器团队正在逐步修改这个策略，您在看到这个文档时，可能最新版本的 X5 浏览器内核已经解决了这个问题。

为什么设置封面无效？

这个问题的原因和上一个问题“div 无法在盖在视频上”是一样的，除非浏览器允许元素能够覆盖 `<video>` 标签，不然封面将会无法显示。

为什么在某些移动端浏览器视频会默认全屏播放？

如果您的视频是在 App 内实现内联播放（即您自己的 App 包装一个 iOS 的 webview 控件，用此控件显示网页，这种模式下您可以对 webview 进行一些细节定制，它的表现可以和标准 safari 浏览器有所不同），可以通过在 HTML 中的 `<video>` 标签设置 `webkit-playsinline` 属性（如果在 iOS10 下，则设置为 `playsinline` 属性），同时 WebView 需要设置 `allowsInlineMediaPlayback`，这样页面在 App 里打开时视频就能以非全屏模式（即内联的方式）播放。

如果您的页面是在 Safari 下打开的，目前 iOS10 以下版本的 Safari 是无法禁止视频自动全屏播放的，iOS10 可以通过前面说的方法（为 `<video>` 标签设置 `playsinline` 属性）实现非全屏模式（即内联的方式）播放，我们的播放器已经自动加上这个属性，只需要终端支持即可。

如果是 Android 终端，众所周知 Android 系统有各种各样的定制版本，每个版本对 `<video>` 标签的实现都有差别，并没有一个完整的统一标准，所以在 Android 上播放视频所展现的一致性，相比 iOS 要差很多，按照现有通用的办法，播放器已经自动加上 `webkit-playsinline` `playsinline` 属性，只需要终端支持即可。

为什么在移动端浏览器视频无法自动播放？

在移动端 Web 自动播放视频只有两个办法，通过设置 `<video>` 标签 `autoplay` 属性 或者调用 `<video>` 标签提供的 `play()` 方法，然而现实是在移动端 Web 中视频自动播放一直是被禁止的，目前通用的办法是通过用户手动触发播放（例如监听用户的单击事件并调用 `play()` 方法），除此之外不排除一些特殊定制的 webview 支持 `<video>` 标签 `autoplay` 属性或者通过其他特殊的函数调用实现自动播放，那么在这类 Webview 下打开页面就有可能自动播放，我们的播放器已经在 `autoplay` 设置为 `true` 的情况下，为 `<video>` 标签加上 `autoplay` 属性，只要终端支持即可。

为什么在 PC Chrome 中 Flash 播放器会有两个播放按钮？

从 Chrome 42 版本开始将不再自动播放 Flash（谷歌购买了 WebRTC 并进行开源并不是没有想法的），只对主要的 Flash 内容进行自动播放，其它的 Flash 内容将被暂停播放，除非用户决定去手动点开它。

为什么在 PC 浏览器中可以播放直播视频，移动端却不行？

在移动端浏览器中播放直播视频，目前只支持 `hls(m3u8)` 协议，因此需要确认直播拉流地址是否有 `hls(m3u8)` 拉流 url，如果您只给我们的播放器一个 `flv` 或者 `rtmp` 的地址，是没有什么办法在手机上观看的。

Web 直播播放器 1.0

Web 直播播放器 1.0

最近更新时间：2018-09-20 17:53:42

功能介绍

腾讯云视频直播播放器 Web SDK 解决方案，可帮助腾讯云视频用户直接使用经过验证的视频播放能力，通过灵活的接口，快速同自有 Web 应用集成，以实现桌面应用播放功能，同时 SDK 提供在 Web 端上传视频的能力。

该 SDK 所播放的文件受限于全局防盗链功能定义，详细内容请查看官网 FAQ 安全功能相关说明，该文档面向考虑使用腾讯云视频直播播放器 Web SDK 进行开发并具备 Javascript 语言基础的开发人员。

格式支持

播放格式：支持 Web SDK 直播视频格式。

播放格式	PC 浏览器	移动端浏览器
HLS (m3u8)	支持	支持
RTMP	支持	不支持
FLV	支持	不支持

Android 系统兼容性问题：不像 iPhone 上只有一个 Safari 浏览器，Android 上的系统标配浏览器有非常多的实现版本，所以 Android 手机浏览器的兼容是一个业界难题，故此表格中所示的手机浏览器格式支持情况比不适用于所有 Android 手机。

准备工作

Step 1：开通服务

在 [腾讯云官网](#) 注册腾讯云账号，并开通直播服务。

Step 2：创建频道

进入 [直播管理控制台](#) 并创建直播频道。

Step 3：获取 ID

您可以在直播管理控制台查找到直播频道并对该频道进行管理，在控制界面您可以找到 app_id，单击该直播频道可以在【基础设置】查询到该频道的频道 ID。





Step 4 : 页面准备

在需要播放视频的页面（包括 PC 或 H5）中引入初始化脚本。

```
<script src="//qzonestyle.gtimg.cn/open/qcloud/video/live/h5/live_connect.js" charset="utf-8"></script>;
```

注意：

播放页面需要挂 IP 或域名访问，如若直接打开本地静态页面将无法播放；
另外可以通过服务端 API 获取频道 ID 和 APPID，具体请参考 [文档链接](#)。

API 基础使用方法

Step 1 : 添加播放器容器

在需要展示播放器的页面位置加入播放器容器，例如：在 index.html 中加入如下代码：

```
<div id="id_video_container" style="width:100%; height:auto;"></div>
```

容器 ID 以及宽高都可以自定义。

Step 2 : 创建 Player 对象

接下来在页面底部引入的 Javascript 脚本中创建一个播放器对象，这时将使用播放器的构造函数：

```
var player = new qcVideo.Player("id_video_container", {  
  "channel_id": "16093104850682282611",  
  "app_id": "1251783441",  
  "width": 480,  
  "height": 320  
});
```

调用构造函数将会生成一个播放器对象，并且根据 channel_id 和 app_id 找到对应的直播视频流进行播放，如果没有 channel_id 和 app_id，您也可以使用直播地址的形式进行播放，具体示例如 [API 使用案例中的 case3](#)，您可以使用播放器对象 player 对播放器进行控制，播放器对象的参数选项下方 API 方法总览有详细

介绍。

完整实例代码

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimum-scale=1.0, maximum-scale=1.0"/>
<title>直播</title>
</head>
<body>
<div id="id_video_container" style="width:100%; height:auto;"></div>
<script src="//qzonestyle.gtimg.cn/open/qcloud/video/live/h5/live_connect.js" charset="utf-8"></script>
<script type="text/javascript">
(function () {
var player = new qcVideo.Player("id_video_container", {
"channel_id": "16093104850682282611",
"app_id": "1251783441",
"width": 480,
"height": 320
});
})();
</script>
</body>
</html>
```

API 使用案例

使用 API 需先完成 API 基础使用方法中的 [添加播放器容器](#) 部分，添加完成后再进行 API 使用。

Case 1：在 PC 或移动端（H5）中播放直播视频

直播 SDK 在 PC 和 H5 中的使用方式是一样的，SDK 会检测平台，自动选择最优的播放方案，例如在 PC 平台，SDK 会优先使用 Flash 播放器以适应多种视频格式的情况（需要 Flash 版本高于 10，否则将提示升级 Flash），而在移动端 H5 则会使用 video 标签进行播放（如果浏览器不支持 H5，则提示使用 QQ 浏览器），SDK 同时支持传频道 ID 或视频文件地址的方式播放。

注意：
两种播放方式不能混合使用。

Case 2：使用频道 ID 播放视频

```
var option = {
"channel_id": "16093425727656143421",
"app_id": "1251132611",
"width": 480,
"height": 320

//...可选填其他属性
};

var player = new qcVideo.Player("id_video_container", option);
```

注意：
使用频道 ID 播放视频不支持直播码的方式。

Case 3 : 使用直播视频地址播放视频

如果没有 app_id 及 channel_id，播放器也支持使用直播视频地址播放视频。

```
var option = {
  "live_url": "http://2157.liveplay.myqcloud.com/2157_358535a.m3u8",
  "live_url2": "http://2000.liveplay.myqcloud.com/live/2000_2a1.flv",
  "width": 480,
  "height": 320

  //...可选填其他属性
};

var player = new qcVideo.Player("id_video_container", option);
```

注意：

最多支持传入两个播放地址，live_url、live_url2，如果这两个地址都传了，那么会按平台支持最好的一个地址选择进行播放，例如当前环境是 PC，那么会优先选择其中为 rtmp 或 flv 的格式，当前环境为手机 H5，会优先选择 hls 格式进行播放。

Case 4 : 如何使用"弹幕"功能?

在播放器初始化完成后，调用播放器对象的 addBarrage(barrage) 方法，可以为视频添加弹幕，具体参数参考 [API 方法总览](#) 的说明，例如，给正在播放的视频添加两条弹幕：

```
var barrage = [
  {"type":"content", "content":"hello world", "time":"1"},
  {"type":"content", "content":"居中显示", "time":"1", "style":"C64B03;30", "position":"center"}
];
player.addBarrage(barrage);
```

注意：

弹幕功能仅在前端实现，后台支持请自行开发，弹幕只在 PC Flash 播放器中生效，H5 暂时不具备弹幕功能。

API 方法总览

构造函数

```
qcVideo.Player(id, option, listener);
```

ID : String，**必选**参数，页面放置播放器的容器 ID，可以自由定义；

option : Object，**必选**参数，播放器的参数可设置选项，具体选项：

参数	类型	默认值	参数说明
channel_id	String	无	用直播视频 ID 播放方式为 必选 参数
app_id	String	无	条件同上为 必选 参数，同一个账户下的视频，该参数是相同的
width	Number	无	必选 ，例如：640，设置播放器宽度，单位为像素
height	Number	无	必选 ，例如：480，设置播放器高度，单位为像素
cache_time	Number	0.3	直播画面开始播放前，最大缓存时间；这个属性可避免有效下行带宽不足导致免 rtmp 直播卡顿的情况（可选参数） 备注：该选项只对 PC 平台 Flash 播放器生效

参数	类型	默认值	参数说明
h5_start_patch	Object	无	H5 播放, 开始播放前贴片 (可选参数) { url: 图片地址, stretch: false //是否拉伸图片铺面整个播放器, 默认 false }
wording	Object	无	直播提示语自定义 (可选参数, 详细信息可见 错误码) { '1': '数据库错误', '2': '连接不到直播源, 直播源没有推送视频内容(hls)', '3': '直播已经结束啦(hls)', '113': '连接超时, 请稍后再试', '114': '连接超时, 请稍后再试', '1000': 'channelID 或者 APPID 错误', '1001': '无效参数, 获取 bizid 失败', '1009': '直播源已失效', '10000': '请求超时, 请检查网络设置', '10008': '密码错误, 请重新输入', //无效密码 '10020': '直播账户余额已不足, 请及时充值', '11044': '无效请求', '11045': '请求参数缺少 channelID', '11046': '密码错误, 请重新输入', '20110': '密码错误, 请重新输入', '20113': '直播已经结束啦, 请稍后再来', // 'downstream type is not exist' 拉流不存在 '20201': '直播已经结束啦, 请稍后再来', // 'get upstream info error' 查询推流错误 '20301': '直播频道不存在, 请确认频道 ID', 'TipReconnect': '重新连接中' 'TipRequireSafari': '当前浏览器不能支持视频播放, 请使用 safari 打开观看' 'TipRequireFlash': '当前浏览器不能支持视频播放, 可下载最新的 QQ 浏览器或者安装 Flash 即可播放' 'TipVideoinfoResolveError': '视频信息解析错误, 请检查参数是否正确', //接口没有返回 json 数据, 无法解析 'TipVideoinfoError': '视频信息错误, 请检查参数是否正确', 'TipConnectError': '连接服务失败, 请检查网络设置', 'TipConnectDeny': '连接服务被拒绝', //Flash 请求触发安全异常 'TipLiveEnd': '直播已经结束啦, 请稍后再来', // NetStream.Play.Stop 事件, 'TipStreamNotFound': '直播已经结束啦, 请稍后再来' //连接不到直播源, 直播源没有推送视频内容 }
live_url	String	无	直播地址, 支持 hls/rtmp/flv 三种格式 用视频地址播放方式为 必选参数
live_url2	String	无	直播地址, 同上 (可选参数)
volume	Number	0.5	设置音量初始值 0 到 1, 默认 0.5 (可选参数) 备注: 该选项只对 Flash 播放器生效
https	Number	0	设置对播放页的 https 支持, 0: 关闭, 1: 开启
hide_volume_tips	Number	0	是否隐藏音量提示, 0: 显示, 1: 隐藏 备注: 该选项只对 Flash 播放器生效
x5_type	String	无	通过 video 属性"x5-video-player-type"声明启用同层 H5 播放器, 支持的值: H5 (该属性为 TBS 内核实验性属性, 非 TBS 内核不支持, 具体介绍参照 TBS 文档)
x5_fullscreen	String	无	视频播放时将会进入到全屏模式, 支持的值: true: 开启 (该属性为 TBS 内核实验性属性, 非 TBS 内核不支持, 具体介绍参照 TBS 文档)
WMode	String	window	Window 模式不支持其他页面元素覆盖在 Flash 播放器上面, 如需要可以修改为 opaque 或其他 flash wmode 的参数值, 可以在 Flash 播放器上进行覆盖其他元素。 备注: 该选项只对 PC 平台 Flash 播放器生效

listener: Object, 可选参数, 播放状态变化回调函数列表。

函数名称	类型	说明
playStatus	function	播放状态变更时触发，回调函数的参数 status : String 返回值： ready: “播放器已准备就绪”，playing: “播放中”，playEnd: “播放结束”，error: “播放异常结束或者遇到错误” //由于移动端播放事件触发条件不一致，如果监听播放结束事件建议同时监听error，以免回调不能正确执行。 type : String 在遇到error时返回错误类型 例子：function(status, type){ ... }

设置和动作

构造函数返回的播放器对象具有以下设置方法

方法	说明
resize(width,height)	参数：width : int ; height : int 功能：设置当前播放器宽度高度 返回：无
play()	功能：开始播放 返回：int (统一返回码) 备注：该功能仅 PC 端 Flash 播放器支持
stop()	功能：停止播放 返回：int (统一返回码) 备注：该功能仅 PC 端 Flash 播放器支持
pause()	功能：暂停当前播放的视频 返回：int (统一返回码) 备注：该功能仅 PC 端 Flash 播放器支持
resume()	功能：恢复播放视频 返回：int (统一返回码) 备注：该功能仅 PC 端 Flash 播放器支持
addBarrage(barrage)	参数：barrage://Array 弹幕信息 <pre> [{ "type":"content", //消息类型, content : 普通文本 (必选) "content":"hello world", //文本消息 (必选) "time":"1.101", //单位秒 , 表示距离当前调用添加字幕的时间多久后, 开始显示该条字幕 (必选) "style": "C64B03;35", // 分号分割, 第一项颜色值, 第二项字体大小 (可选) "position":"center" //固定位置 center: 居中, bottom: 底部, up: 顶部 (可选), ...] </pre> 功能：添加弹幕 返回：int 返回码 备注：弹幕仅在前端实现，后台功能请自行开发，该功能只在 PC Flash 播放器中生效
closeBarrage()	功能：关闭弹幕，关闭后重新调用 addBarrage 可开启弹幕。 返回：int 返回码 备注：弹幕仅在前端实现，后台功能请自行开发，该功能只在 PC Flash 播放器中生效

这些设置方法的统一返回码是：

错误码	含义
200	操作成功
0	播放器未完全初始化
-2	未知操作命令

问题排查

最近更新时间：2018-07-10 18:12:05

错误码列表

SDK 使用过程中出现的异常 code 对照表

- 前端返回错误

提示语	说明
视频信息解析错误，请检查参数是否正确	接口没有返回 json 数据，无法解析
视频信息错误，请检查参数是否正确	视频信息解析错误
连接服务失败，请检查网络设置	获取视频频道信息失败
连接服务被拒绝	Flash 请求触发安全异常
缺少视频数据	视频源数据缺失
直播已经结束啦，请稍后再来	NetStream.Play.Stop 事件
直播已经结束啦，请稍后再来	连接不到直播源，直播源没有推送视频内容

- 后台返回错误

Code	提示语	说明
1	数据库错误	数据库连接超时或者查询错误
2	连接不到直播源，直播源没有推送视频内容 (hls)	无法获取 m3u8 文件，连接不到直播源
3	直播已经结束啦 (hls)	Manifest 不是合法的 m3u8 文件或直播源已结束
113	连接超时，请稍后再试	参数错误
1000	channelID 或者 APPID 错误	app_id 填写错误 (长度错误)
1001	无效参数，获取 bizid 失败	app_id 填写错误 (长度正确，内容错误)
1009	直播源已失效	无效播放地址，直播源已失效
10000	请求超时	拉取播放器配置信息与视频信息超时，请检查网络重试，超时时间为 10s
10001	数据解析失败	拉取播放器配置信息与视频信息获取到的数据解析失败，可能是网络问题或者服务器异常
10002	连接超时，请稍后再试	拉取播放器配置信息与视频信息失败，可能是网络问题或者服务器异常
10008	密码错误，请重新输入	无效密码
10020	直播账户余额已不足，请及时充值	余额不足
11044	无效请求	缺少 APPID
11045	请求参数缺少 channelID	缺少 Channel ID
11046	密码错误，请重新输入	缺少密码
20110	密码错误，请重新输入	无效密码
20113	直播已经结束啦，请稍后再来	downstream type is not exist 拉流不存在
20201	直播已经结束啦，请稍后再来	get upstream info error 查询推流错误

Code	提示语	说明
20301	直播频道不存在，请确认频道 ID	channel_id 错误 (app_id 填写正确)

注意：

如遇到未在列表中的异常 code，请联系我们的客服，客服会安排工程师进行解决。

常见问题
为什么 H5 播放视频拉伸变形了？

H5 并不具备拉伸视频的能力，请检查播放器的容器宽高是否设置正确。

为什么提示“直播已经结束啦，请稍后再来”

播放连接视频直播地址未收到响应并重试 5 次后仍未成功将出现这个提示，这时需要确认视频是否在进行推流，网络是否通畅。

QQ 浏览器下无法在盖住视频

浏览器接管了 H5 的视频播放功能，X5 内核视频播放使用了自研的播放器，考虑用户体验，浏览器使用了统一的播放界面，相关信息参考 [QQ 浏览器文档说明](#)。

iOS 系统下视频自动全屏播放

iOS 系统由于 webkit 设置原因，默认视频全屏播放，如果您的视频需要在 App 内实现内联播放，可以设置 webkit-playsinline 属性，目前 iOS10 以下版本的 Safari 无法禁止视频自动全屏。

为什么在 PC Chrome 中 Flash 播放器会有两个播放按钮？

从 Chrome 42 版本开始将不再自动播放 Flash，只对主要的 Flash 内容进行自动播放，其它的 Flash 内容将被暂停播放，除非用户决定去手动点开它。

为什么在 PC 浏览器中可以播放直播视频，移动端却不行？

在移动端浏览器中播放直播视频，目前只支持 hls 协议，因此需要确认直播拉流地址是否有 hls 拉流 url。