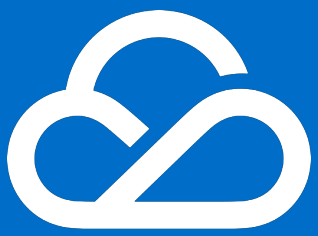
腾讯云移动直播

Web端集成

产品文档



腾讯云

【版权声明】

【商标声明】

【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

Web播放器

功能介绍

腾讯云Web播放器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滴。
  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哦。

- 协议的支持（手机）
  Web播放器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，没有办法，在手机浏览器上我们并没有其它选择。

- 协议的支持（PC）
  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控件的开发商 Adobe 并不追求设计上的洁癖，所以flash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它家自己开发，所以兼容性非常好（也是因为这个原因，意图推广 webrtc 和 HTML5 技术的 Google 在最新版的 Chrome 浏览器里默认禁用了 Flash）

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	不支持	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.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>
```

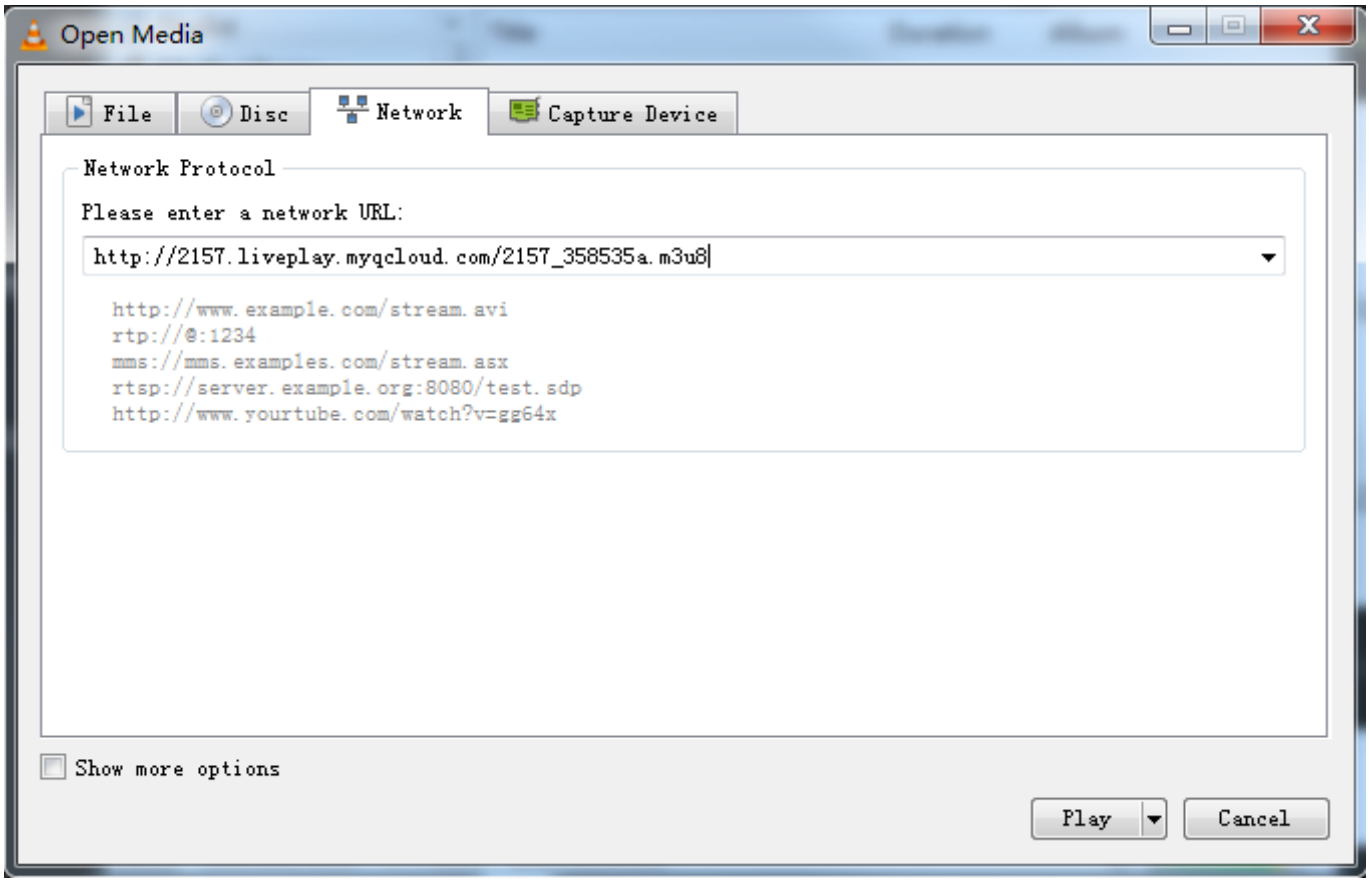
Step 3：对接视频的播放

接下来就要写 javascript 代码了，这些代码的作用是去指定的 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代码可以这样写：

```
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）协议的直播视频了，不过，前面我们有说过，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

那么对于PC浏览器而言，我们是否可以做的更好呢？当然可以，因为PC浏览器支持flash，它可强大多了，现在我们的代码要这样写：

```
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，因为可以实现更低的延迟。

当然，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视频直播服务，这个问题是不需要犯愁的，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，基本上逃不出如下几个原因：

- （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了，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了。

- （原因二）本地网页调试
  目前腾讯云的 Web 播放器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。当然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
  ，不过只有当您要播放的视频URL不是隶属于腾讯云时，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

备注：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？

这里要特别科普一下：
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。

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，分出两路转码后的视频，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。

http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.m3u8 // 原画
http://200002949.vod.myqcloud.com/200002949_b6ffc.f230.av.m3u8 // 高清
http://200002949.vod.myqcloud.com/200002949_b6ffc.f220.av.m3u8 // 标清

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.f0.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，里面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	视频解码错误	视频格式WEB播放器无法解码(H5提示的错误)
3	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)
4	获取视频失败，请检查播放链接是否有效	(H5提示的错误)
5	当前系统环境不支持播放该视频格式	(H5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)

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

您也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://2527.vod.myqcloud.com/2527_
b3907044441c11e6a46d294f954f93eb.f230.av.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	示例
m3u8	String	无	原画m3u8 播放URL	http://2157.liveplay.myqcloud.com/2157_358535a.m3u8
m3u8_hd	String	无	高清m3u8 播放URL	http://2157.liveplay.myqcloud.com/2157_358535ahd.m3u8
m3u8_sd	String	无	标清m3u8 播放URL	http://2157.liveplay.myqcloud.com/2157_358535asd.m3u8
flv	String	无	原画flv 播放URL	http://2157.liveplay.myqcloud.com/2157_358535a.flv
flv_hd	String	无	高清flv 播放URL	http://2157.liveplay.myqcloud.com/2157_358535ahd.flv
flv_sd	String	无	标清flv 播放URL	http://2157.liveplay.

参数	类型	默认值	参数说明	示例
				myqcloud.com/2157_358535asd.flv
mp4	String	无	原画mp4 播放URL	http://200002949.vod.myqcloud.com/200002949_b6ffc.f0.mp4
mp4_hd	String	无	高清mp4 播放URL	http://200002949.vod.myqcloud.com/200002949_b6ffc.f40.mp4
mp4_sd	String	无	标清mp4 播放URL	http://200002949.vod.myqcloud.com/200002949_b6ffc.f20.mp4
rtmp	String	无	原画rtmp 播放URL	rtmp://2157.liveplay.myqcloud.com/live/2157_280d88
rtmp_hd	String	无	高清rtmp 播放URL	rtmp://2157.liveplay.myqcloud.com/live/2157_280d88hd
rtmp_sd	String	无	标清rtmp 播放URL	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	640
height	Number	无	必选，设置播放器高度，单位为像素	480
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，	true

参数	类型	默认值	参数说明	示例
			以及区分点直播的处理逻辑	
autoplay	Boolean	false	是否自动播放 备注：该选项只对大部分PC平台生效	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	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	"system"
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)	"h5"
x5_fullscreen	String	无	通过 video 属性 “x5-video-player-fullscreen” 声明视	"true"

参数	类型	默认值	参数说明	示例
			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)	
wodring	Object	无	自定义文案	{ 2032: '请求视频失败，请检查网络'}
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	function(msg){ //进行事件处理 }

进阶攻略

这里介绍一些视频播放器SDK的进阶使用方法

监听事件

视频云的播放器是采用 H5

<video>

和 Flash 相结合的方式来进行视频播放，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。

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

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

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，iOS 10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（例如监听用户的点击事件并调用 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。

如何在朋友圈中看直播？

功能介绍

如果您的产品中直播内容很好，那么仅靠 APP 的传播势必阻碍产品的发展速度，依靠社交平台（微信朋友圈、新浪微博等等）的网页链接分享能力能够很好地解决这个问题。

微信朋友圈分享
是目前最流行也是最有效果的社交传播方案，本文就以微信分享为例，介绍如何为您的产品构建起 Web 端的闭环体验。

主播分享

iOS页面

Android页面

技术难点

要在网页上观看直播，只需要在页面中添加一个 video标签
即可，这看似并没有什么难度。然而，要做到较好的用户体验，需要克服很多的技术难点：

- 如何在Android浏览器上做好兼容？
  Android 浏览器的差异就跟 Android 系统一样种类繁多，制式和标准各不统一，经常会遇到在某款手

机上效果调整OK，但到了另一款手机上就完全变了样，兼容性问题是个非常耗时，且依赖经验的过程。

- 如何在Web页面上跟主播文字互动？
  主播用 APP 做直播，观众在网页端观看，那么只有观看没有互动能有什么效果呢？所以我们必须要解决在网页上跟主播互动的难题。

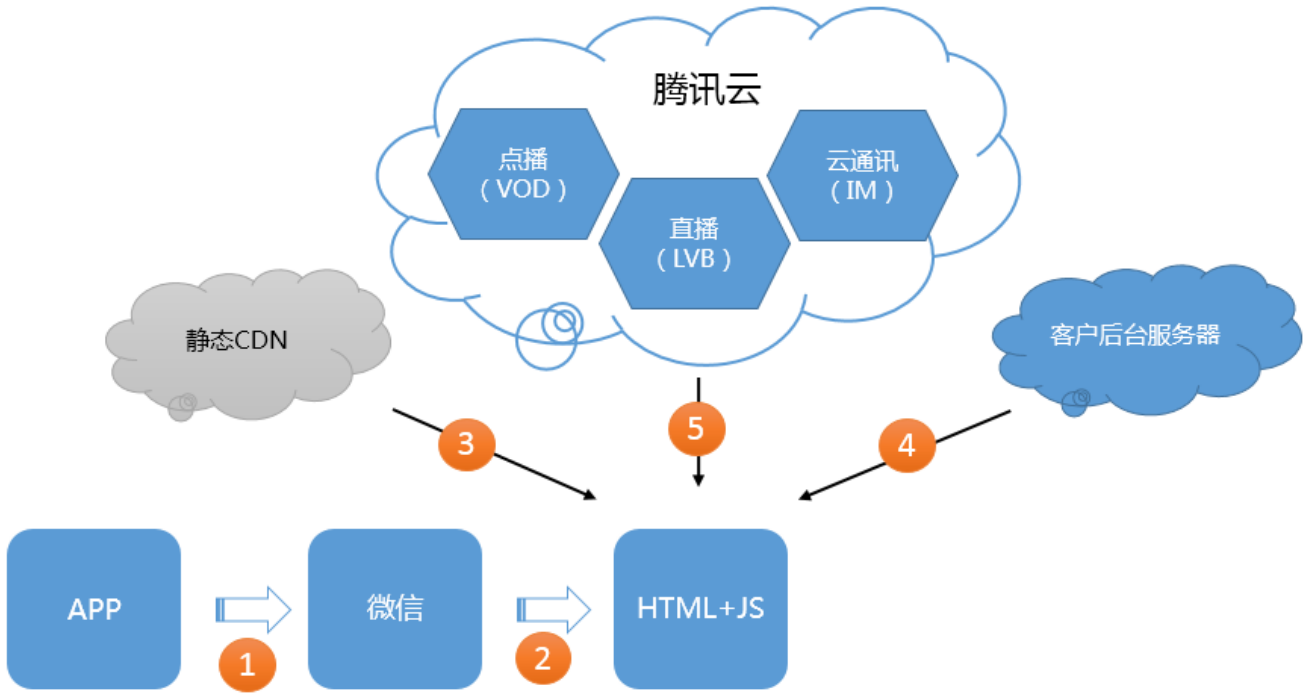
- 面对 X5 浏览器内核应当怎样应对？
  Android系统上，从微信或者QQ上点开的网页，基本都是QQ浏览器推出的 X5 内核，X5 内核在视频播放这里有自己的策略和规则，如果您不清楚，会面临很多的坑要踩。

- 面对 PC 浏览器内核应当怎样应对？
  PC 上的浏览器都不能原生支持直播视频流，所以如果 Web 页面被在 PC 上打开却看不了视频，也是非常尴尬的事情。

好，接下来就让我们通过如下的对接攻略，帮您解决上述提到的几个技术难题。

对接攻略

整体技术架构如下图所示：



对接步骤	步骤说明	参考指引
step1	您的APP在直播开始前，先要把分享URL先分享到微信朋友圈上去。	微信分享URL
step2	分享者好的好友在朋友圈中看到这条分享，并点击打开。	用户打开链接
step3	微信内嵌的WebView（浏览器）会从URL指定的静态CDN服务器上拉取Web页面（HTML+JS）	Web页面制作
step4	网页中的JS（javascript）代码通过ajax 异步请求方式到您的业务后台Server获取必要的展示信息。	Web后台搭建
step5	网页通过组合直播（LVB）、点播（VOD）和云通讯（IM）等服务，实现Web端的直播观看体验。	页面做了什么

Step1: 微信分享URL

我们在小直播的源码中，使用友盟的组件默认实现了到新浪、微信、朋友圈、QQ 以及 QQ空间的分享能力，以便您直接快速参考:

所属平台	源码参考	参考文档
iOS 平台	在“小直播”的终端源码包中搜索 "WechatTimeLine" 即可。	友盟分享组件
Android 平台	在“小直播”的终端源码包中搜索 "ShareAction" 即可。	友盟分享组件

具体分享的URL是什么呢？我们会在 页面做了什么 中向您详细介绍。

Step2: 用户打开链接

这一步是用户操作，我们不需要做什么，但是有几个点我们需要给您同步：

- 安全拦截

  不是所有的视频地址都能在微信里播放的，更精确的说，是大部分 URL 都不行，必须要去微信申请安

全许可。否则，即使网页可以打开，里面的视频也是加载不出来，因为微信在流媒体加载前就已经给你拦截掉了。

- X5 内核
  由于都是隶属腾讯旗下的产品，Android 版微信的内嵌网页，大概率是使用 X5 浏览器内核（QQ浏览器内核）打开页面，但也有可能是使用系统自带的浏览器打开。X5 浏览器内核对常见流媒体视频协议的支持还是不错的，尤其是 HLS(m3u8) 和 MP4。

- Safari 内核
  在 iOS 平台并没有其它浏览器内核出来表演的份，只有 Safari 内核可以使用，但是微信有能力操控它内嵌的 WebView 控件实现一些 iPhone 自带的 Safari 浏览器默认不开启的功能，比如 Video 的自动播放。（然而，这个功能指开放给了部分域名）

Step3: Web页面制作

这部分工作可以交给贵团队的 Web 前端工程师处理。当然，从零开始制作一个支持在线直播观看，同时又具备聊天室功能的 Web 页面，还是需要一定的经验储备的，所以我们想要尽可能地为您提供一些快速上手的参考：

小直播源码集中包含 Web 分享页面的 DEMO源码 ，实现了直播观看、文本消息以及点赞等功能。

文件名	所属文件夹	功能说明
mobile.css	css	示例网页的CSS样式表文件，负责控制页面的外观和形态，可以自由修改和替换。
pictures	img	示例网页所用的示例图片，您可以根据需要自行替换和调整。
json2.js	sdk	IM 云通讯的基础JS库，主要用于提供Web端的聊天室功能相关API。
webim.js	sdk	IM 云通讯的基础JS库，主要用于提供Web端的聊天室功能相关API。
base.js	js	IM 云通讯的基础JS库，主要用于提供Web端的聊天室功能相关API。

文件名	所属文件夹	功能说明
lib.js	js	本页面依赖的一些基本的 javascript 公共脚本文件
config.js	js	配置中心，比如后台服务器地址就是在这里进行配置
xzb.js	js	核心 javascript 文件，直播观看和 IM 聊天室的实现逻辑，都位于此 js 中。
xiaozhibo.html	根目录	唯一的 html 页面，其中的 PlayerContainer 为视频渲染区域，其上紧贴着的 div 是聊天区域。

3.1 配置config.js

配置项	含义	参考文档
SERVER	为该网页提供视频播放信息的业务服务器。	DOC
accountMode	IM SDK 的账号集成模式（配置要跟小直播中保持一致）。	DOC
sdkAppID	IM 服务开通后分配的一个id，如果分享URL的参数中携带可以不配置。	DOC
accountType	IM 服务开通后分配的一个type，如果分享URL的参数中携带可以不配置。	DOC

3.2 调试&部署页面

- 如果您要调试，要注意直接 在 Windows 下用浏览器打开 xiaozhibo.html 是不行的，需要将其上传到一台可访问的服务器上进行调试，如果您没有自己的服务器，可以参照源码包里的 readme.pdf 部署一台。

- 静态网页的部署推荐使用CDN 内容分发网络，CDN 的优势就是能极大的缩减用户打开页面的速度，从而提升用户体验。

Step4: Web后台搭建

这部分工作可以交给贵团队的后端工程师处理，主要工作就是在您的业务服务器上提供一个 信息查询接口：。

因为 Step3 中的 Web 页面是静态的，但是每个直播的信息都不一样，比如主播是谁？主播的头像是什么？房间的标题是什么？甚至，有没有录播的回放录像？

这些信息就需要 Web 页面中的 xzb.js 通过 ajax
异步请求到您的业务后台服务器查询。我们在小直播中，将这一查询协议定义为 GetUserInfo。

小直播源码集中包含 PHP 后台服务器 DEMO源码 ，其中 GetUserInfo.php 中提供了对 GetUserInfo
协议的实现。

Step5: 页面做了什么

完成 Step4 之后基本已经搞定了本文档所言功能的对接，但到这里您的感觉可能是：

> “按照你的步骤做完了，但好像整个系统就是个黑盒子，它里面原理是什么？”

接下来我们从分享URL的构成和网页的内部原理两个方面来介绍其内部的原理：

5.1 分享URL的构成

```
http://imgcache.qq.com/open/qcloud/video/share/xiaozhibo.html?sdkappid=1400012345&acctype=8888&u
st1234&type=0
```

这个 URL 的主体是一个叫做 xiaozhibo 的 html 页面，它被放置于 imgcache.qq.com
域名上，该域名是腾讯云的静态CDN 地址。

> html
> 的名字以及服务器的域名您都可以根据自己的情况部署，但我们很推荐您将网页部署到腾讯云的静态

CDN 上，因为它可以让您的用户不论是在北京还是在青海，都能从最近的服务器拉取到该文件。

接下来，xiaozhibo.html 后面跟了一组参数，这组参数用来告诉 Web 页面： 应该进哪个聊天室？
应该播哪个视频URL？主播叫什么？头像是什么样子？视频究竟是直播还是点播？

参数名	含义	备注
sdkappi	IM 服务开通后分配的一个id，用来进入聊天室的必备信息。	参考文档
acctype	IM 服务开通后分配的一个type，跟 sdkappi 配合使用。	参考文档
userid	主播的 userid	在小直播里，主播 id 即为房间号
type	视频类型	0 - 代表直播， 1 - 代表点播，也就是录像回放

5.2 网页的内部原理

xiaozhibo.html 中挂载的 xzb.js 是网页的主控 javascript
文件，也就是驱动整个页面的逻辑中枢，它以如下的步骤去实现整个页面的功能：

- 主函数
  init() 为全局主入口函数，它串联起整个页面的全部逻辑链条：initParams() -> initLogin() -> initPlayer() -> ...

- 解析URL中的参数
  initParams() 函数负责将 URL 尾部的 userid 等参数解析出来，initParams() 的最后一个动作是去 config.js 中 SERVER 配置项所指定的服务器地址上，用 userid 作为参数查询播放URL。

- 创建播放器
  initPlayer() 函数会根据当前是 PC 浏览器还是手机浏览器，选择相应的方式创建播放器。

- 视频播放
  loadVideo() 负责驱动网页中的播放器播放视频URL，需要您注意的是，大部分手机浏览器是限制视频

的自动播放的（可能设计者考虑流量的问题），所以如果您发现同样的页面在不同手机上的自动播放表现不一致，这并不是什么奇怪的事情。

- 登录到聊天室
  initLogin() 通过从 URL 中解析的 sdkappid 和 acctype 等参数（如果 URL 中不懈怠则直接使用 config.js 中配置的 sdkappid 和 acctype），登录到聊天室中（Web 页面不具备创建聊天室的能力，所以成功进入的前提是主播在小直播 App 端已经创建了聊天室）。

常见问题

1. 微信拦截

不管直播还是点播都会有一个播放URL，需要把播放地址的域名添加到安全域名的列表里，不添加会有可能被微信以"非微信官方网页"为由拦截视频地址，导致不能播放视频。

登录微信公众平台进入“公众号设置”的“功能设置”里添加“JS接口安全域名”。设置域名后，该域名的网页内容不会被重新排版或者被拦截，但依然要遵守微信平台运营规则，否则依然会受到相应处罚。

怎么去掉“非微信官方网页，将由微信转换为手机预览模式”的提示页面？

2. Flash 跨域

PC 浏览器要依靠 Flash 控件完成视频播放，但 Flash 本身有跨域问题，如果你的网页以及 Step4 中的后台服务器不是部署在腾讯云的，需要在服务器的根目录下添加跨域配置文件 crossdomain.xml ：

```
<?xml version="1.0"?>
<cross-domain-policy>
  <allow-access-from domain="*" secure="false"/>
</cross-domain-policy>
```