

腾讯云点播

播放器 SDK 手册

产品文档



腾讯云

【版权声明】

©2013-2017 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

文档声明.....	2
播放器 SDK 手册.....	4
点播播放综述.....	4
超级播放器 Android.....	5
超级播放器 iOS.....	10
超级播放器 Web.....	14
Web 视频播放器综述.....	14
使用文档.....	16
开发文档.....	31
播放器 SDK 下载.....	37

播放器 SDK 手册

点播播放综述

概述

腾讯云点播提供了移动端和 Web 端播放器。开发者可以快速集成，以极为简单的方式播放点播中的视频。

特点

- 仅需要视频在点播中的 fileId 即可播放，简单快速。
- 如果视频经过转码，播放器在播放视频时会自动展示出多种清晰度供用户选择（流畅、标清、高清、超清等）。
- 如果视频开启了视频播放控制（视频加密、Referer 防盗链和 Key 防盗链），播放器可以正常支持。

场景

点播播放器仅用于播放发布在腾讯云点播中的视频，不适用于播放其他平台的视频资源。

移动端播放器

提供 Android 和 iOS 平台的 SDK，适合在移动端 APP 中集成点播视频的播放功能。

Web 端播放器

集成在 Web 服务器，用于客户端（PC 和移动端）通过浏览器观看点播视频。（Web 播放器尚未发布，敬请期待）

超级播放器 Android

功能介绍

超级播放器是基于

TXVodPlayer

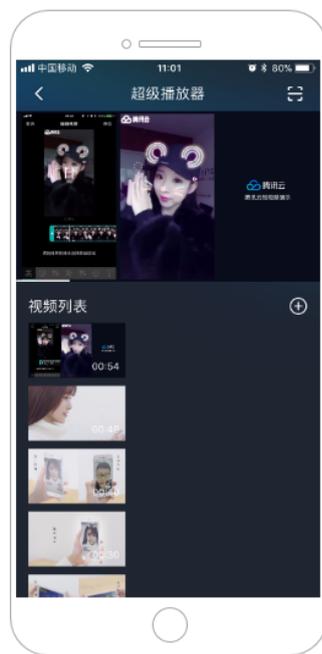
实现的集视频信息拉取、

横竖屏切换、清晰度选择、弹幕等功能于一体的解

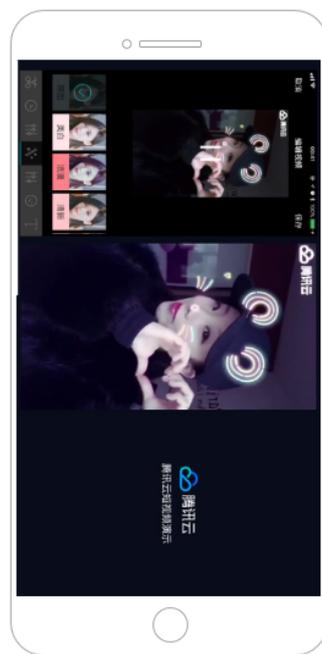
决方案，且完全开源。帮助您在短时间内，打造一个媲美市面上各种流行视频App的播放体验。



体验入口



竖屏模式



横屏模式



接入准备

1. 下载 SDK + Demo 开发包，下载地址为([Android](#))。

2. 播放器的 UI 部分代码开源，开源代码位于

`app/src/main/java/com/tencent/liteav/demo/play/`

文件夹中，图片资源位于

```
app/src/main/res/drawable-xxhdpi/
```

文件夹中，您需要先将这两部分拷贝到您的App工程中。

3. Demo的弹幕集成了第三方开源库

DanmakuFlameMaster

，可以自行在github获取，也可如Demo在build.gradle示例配置

```
compile 'com.github.ctiao:DanmakuFlameMaster:0.5.3'
```

创建播放器

超级播放器主类为

SuperVideoPlayer

，您需求先创建它。

```
mSuperVideoPlayer = (SuperVideoPlayer) findViewById(R.id.video_player_item_1);  
mSuperVideoPlayer.setVideoPlayCallback(mVideoPlayCallback);
```

视频信息获取

与播放普通url地址不同，获取视频信息需要通过fileId方式。

```
TXPlayerAuthParams *p = [TXPlayerAuthParams new];  
p.appId = 1252463788;
```

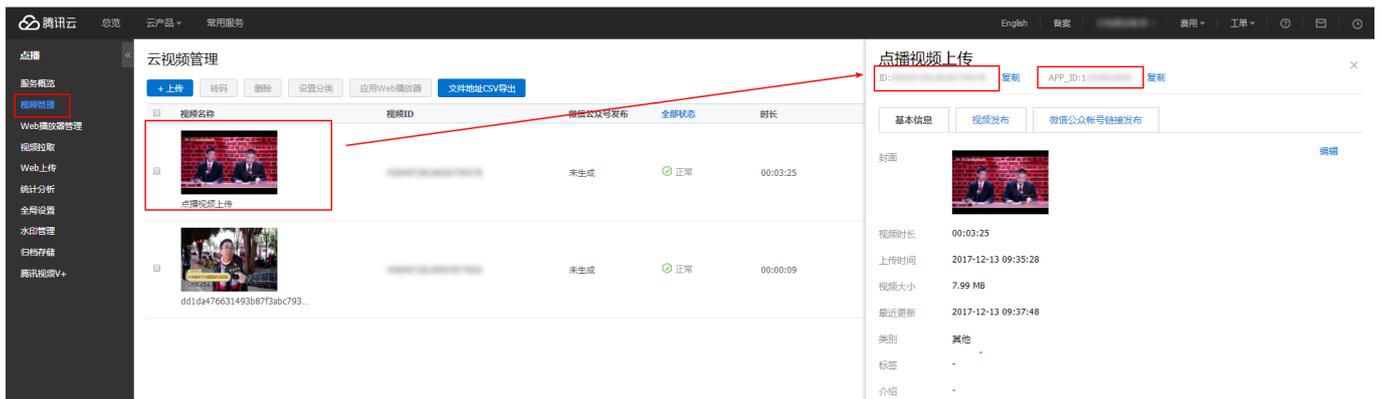
```
p.fileId = @"4564972819220421305";
```

```
TXPlayerAuthBuilder authBuilder = new TXPlayerAuthBuilder();
try {
    authBuilder.setAppId(Integer.parseInt(playerAuthParam.appId));
    authBuilder.setFileId(playerAuthParam.fileId);
    mTXPlayerGetInfo.startPlay(authBuilder);
} catch (NumberFormatException e) {
    Toast.makeText(mContext, "请输入正确的AppId", 0).show();
}
```

fileId在一般是在视频上传后，由服务器返回：

1. 客户端视频发布后，服务器会返回fileId到客户端
2. 服务端视频上传，在[确认上传](#)的通知中包含对应的fileId

如果文件已存在腾讯云，则可以进入 [点播视频管理](#)，找到对应的文件。点开后再右侧视频详情中，可以看到appId和fileId。



SDK在请求成功后，将视频信息将以事件的形式通知到上层

在Demo中SuperVideoPlayer中示例

```
mTXPlayerGetInfo = new TXVodPlayer(context);
mTxplayer.setVodListener(mPlayVodListener);
```

```
mTXPlayerGetInfo.setVodListener(mGetVodInfoListener);
```

```
/**
 * 获取fileId对应的视频信息
 */
private ITXVodPlayListener mGetVodInfoListener = new ITXVodPlayListener() {
    @Override
    public void onPlayEvent(TXVodPlayer player, int event, Bundle param) {
        String playEventLog = "receive event: " + event + ", " +
param.getString(TXLiveConstants.EVT_DESCRIPTION);
        Log.d(TAG, playEventLog);

        if (event == TXLiveConstants.PLAY_EVT_GET_PLAYINFO_SUCC) { // 获取点播文件信息成功
            VodRspData data = new VodRspData();
            data.cover = param.getString(TXLiveConstants.EVT_PLAY_COVER_URL);
            data.duration = param.getInt(TXLiveConstants.EVT_PLAY_DURATION);
            data.url = param.getString(TXLiveConstants.EVT_PLAY_URL);
            if (mVideoPlayCallback != null) {
                mVideoPlayCallback.onLoadVideoInfo(data);
            }
        }
    }
};
```

切换视频

播放器播放另一个视频，调用

```
setPlayUrl
```

即可

```
String url = "http://1252463788.vod2.myqcloud.com/xxx/yyy/v.f20.mp4";
```

```
if (mSuperVideoPlayer != null) {  
    mSuperVideoPlayer.updateUI("新播放的视频");  
    mSuperVideoPlayer.setPlayUrl(url);  
}
```

移除播放器

当不需要播放器时，调用onDestroy清理播放器内部状态，防止干扰下次播放。

```
if (mSuperVideoPlayer != null) {  
    mSuperVideoPlayer.onDestroy();  
}
```

超级播放器 iOS

功能介绍

超级播放器是基于

TXVodPlayer

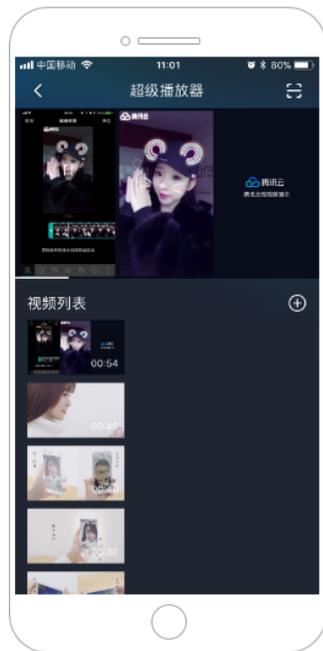
实现的集视频信息拉取、

横竖屏切换、清晰度选择、弹幕等功能于一体的解

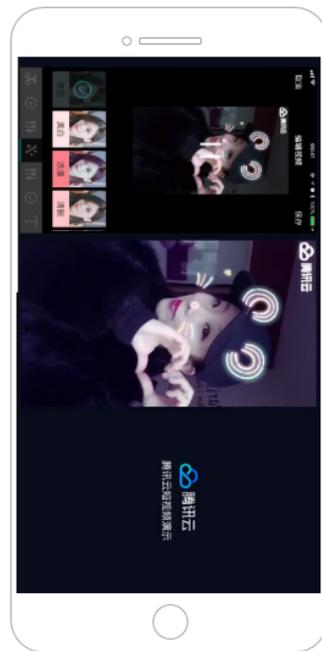
决方案，且完全开源。帮助您在短时间内，打造一个媲美市面上各种流行视频App的播放体验。



体验入口



竖屏模式



横屏模式



接入准备

1. 下载 SDK + Demo 开发包，下载地址为 ([iOS](#))。
2. 超级播放器的 UI 部分源码开源，开源代码位于

Player

文件夹，图片资源位于

Resource/Player

文件夹，您需要先将这两部分拷贝到您的App工程中。其它依赖的第三方库您可以自行 Pod 添加或在

Third

目录中获取

- Masonry
- SDWebImage

创建播放器

超级播放器主类为

ZFPlayerView

，您需求先创建它并添加的需要的父View中。

```
_playerView = [[ZFPlayerView alloc] init];
```

```
[_playerView playerControlView:nil playerModel:self.playerModel];
```

```
// 设置代理
```

```
_playerView.delegate = self;
```

```
// 加载成功后自动播放
```

```
[self.playerView autoPlayTheVideo];
```

视频信息获取

与播放普通url地址不同，获取视频信息需要通过fileId方式。

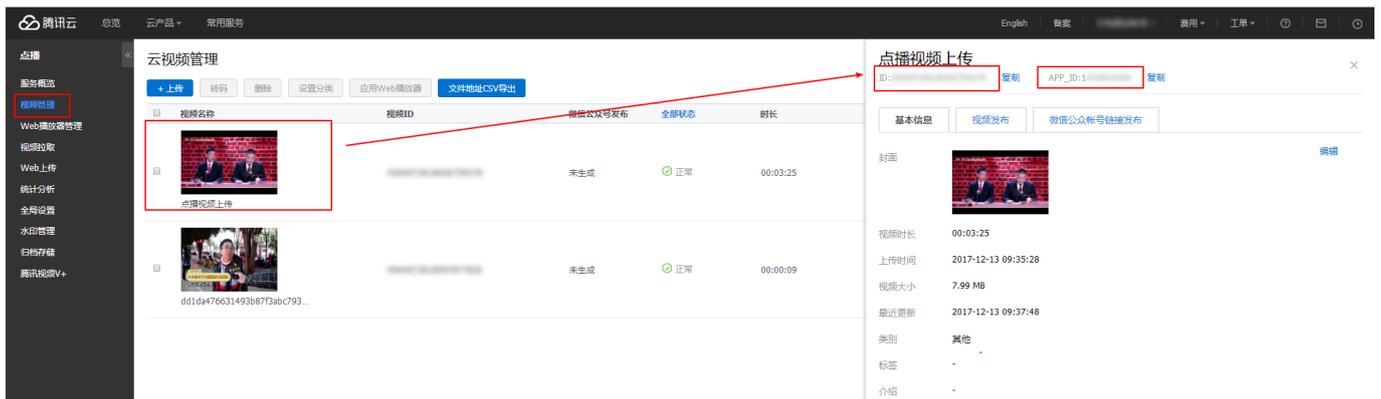
```
TXPlayerAuthParams *p = [TXPlayerAuthParams new];
p.appId = 1252463788;
p.fileId = @"4564972819220421305";
```

```
self.getInfoPlayer = [[TXVodPlayer alloc] init];
[self.getInfoPlayer setIsAutoPlay:NO];
self.getInfoPlayer.vodDelegate = self;
[self.getInfoPlayer startPlayWithParams:p];
```

fileId在一般是在视频上传后，由服务器返回：

1. 客户端视频发布后，服务器会返回fileId到客户端
2. 服务端视频上传，在[确认上传](#)的通知中包含对应的fileId

如果文件已存在腾讯云，则可以进入 [点播视频管理](#)，找到对应的文件。点开后再右侧视频详情中，可以看到appId和fileId。



SDK在请求成功后，视频信息将以事件的形式通知到上层

```
- (void)onPlayEvent:(TXVodPlayer *)player event:(int)EvtID withParam:(NSDictionary *)param
{
    if (EvtID == PLAY_EVT_GET_PLAYINFO_SUCC) {
```

```
ListVideoModel *model = [ListVideoModel new];
model.cover = param[EVT_PLAY_COVER_URL];
model.duration = [param[EVT_PLAY_DURATION] intValue];
model.url = param[EVT_PLAY_URL];
}
}
```

切换视频

播放另一个视频，您需要重新创建一个

playerModel

，调用resetToPlayNewVideo即可

```
_playerModel.title = [cell getSource].title;
_playerModel.videoURL = [NSURL URLWithString:[cell getSource].url];
_playerModel.placeholderImage = [UIImage imageWithData:[NSData dataWithContentsOfURL:
    [NSURL URLWithString:[cell getSource].cover]]];

[_playerView resetToPlayNewVideo:self.playerModel];
```

移除播放器

当不需要播放器时，调用resetPlayer清理播放器内部状态，防止干扰下次播放。

```
[self.playerView resetPlayer]; //非常重要
```

超级播放器 Web

Web 视频播放器综述

腾讯云 WEB 视频播放器介绍

腾讯云针对不同的业务场景和使用场景，提供了不同的 WEB 视频播放器，分别为：

[点播超级播放器](#)

[直播播放器1.0](#)

[TCPlayer Lite](#)

[点播播放器1.0](#)

下面将对这些播放器进行简要介绍

点播超级播放器

点播超级播放器是基于 video.js 框架并结合腾讯云点播业务而开发的视频播放器，采用以 HTML5

<video>

为主 Flash 为辅的播放方式，在浏览器不支持HTML5

<video>

的情况下采用 Flash 播放。播放器界面由 HTML CSS 实现，可以通过 CSS 定制界面。

该播放器只适用于使用腾讯云点播业务的 WEB 点播视频播放场景，并且会持续进行功能迭代更新。

具体介绍请看[使用文档](#)、[开发文档](#)

直播播放器1.0

直播播放器1.0是结合腾讯云直播业务而开发的视频播放器，以 Flash 为主要播放模式，需要浏览器支持并启用 Flash 插件。

该播放器只适用于使用腾讯云直播业务的 WEB 直播视频播放场景。

具体介绍请看[使用文档](#)

TCPlayer Lite

独立播放器 TCPlayer Lite 实现了基本的视频播放器功能，采用 HTML5 和 Flash 相结合的播放模式，支持播放 hls、mp4 格式的点播视频和 RTMP、http-flv、hls 协议的直播视频，支持主要的桌面和移动端浏览器。

该播放器仅支持传入地址播放，不关联业务，适用于轻量化 WEB 视频播放场景，支持点播和直播，支持通过 CSS 定制化界面。

具体介绍请看[使用文档](#)

点播播放器1.0

点播播放器1.0是结合腾讯云点播业务而开发的视频播放器，以 Flash 为主要播放模式，需要浏览器支持并启用 Flash 插件。该播放器主要面向比较老旧的浏览器，比如IE8、9、10、11等。

由于 Flash 逐步被现代浏览器禁止运行，并且 Flash 将于2020年停止开发和更新，腾讯云点播推出了以 HTML5 播放为主的点播超级播放器方案，逐步替换点播播放器1.0。

点播播放器1.0具体介绍请看[使用文档](#)

如何选择播放器

通过以上对各个播放器的介绍，了解每种播放器的特点以及适用场景，建议结合所使用的腾讯云业务，以及自身所具备的开发能力来选择合适的 WEB 播放器。

如遇到播放问题，请查看[常见问题文档](#)

使用文档

前言

本文档是介绍腾讯云视频点播服务的 Web 播放器，它可以帮助腾讯云客户通过灵活的接口，快速与自有 Web 应用集成，实现视频播放功能，本文档适合有一定 Javascript 语言基础的开发人员阅读。

能力介绍

腾讯云视频点播 WEB 播放器是通过 HTML5

<video>

标签以及 Flash 实现视频播放。在浏览器不劫持视频播放的情况下，实现了视频播放效果的多平台统一体验，并结合腾讯云点播视频服务，提供防盗链、HLS 加密播放等功能。

视频格式支持

播放格式	PC 浏览器环境	移动端浏览器环境
HLS (m3u8)	支持 (IE11/10/9/8 需要开启 Flash)	支持 (Android 4+、iOS 原生支持)
MP4	支持 (IE8 需要开启 Flash)	支持 (Android 4.4+、iOS 原生支持)

平台支持

桌面端支持最新版本的现代浏览器 Chrome , Firefox , Safari , Edge , QQ 浏览器 , 以及非现代浏览器 IE11/10/9/8 (IE11/10/9/8 需要开启 Flash , 只支持 Win7 IE8)

移动端只要实现 HTML5

<video>

标准的浏览器都支持，比如 Android Chrome , iOS Safari , 微信 , 手机 QQ , 手机 QQ 浏览器等使用本播放器，同一段代码可以自动实现 PC

浏览器和手机浏览器的自适应切换，播放器内部会自动区分平台使用最优的播放方案。例如：在 IE11/10/9/8 浏览器中会使用 Flash 播放器以实现其不支持 HTML5 播放 HLS 的能力，在 Chrome 等现代浏览器中优先使用 HTML5 技术实现视频播放，而手机浏览器上会使用 HTML5 技术实现视频播放。

点播平台的转码服务

由于 MP4 和 HLS (m3u8) 是目前在 PC 浏览器和手机浏览器上支持程度最广泛的格式，所以腾讯云的点播平台最终会把上传的视频发布为 MP4 和 HLS (m3u8) 格式。

准备工作

step 1 : 开通服务

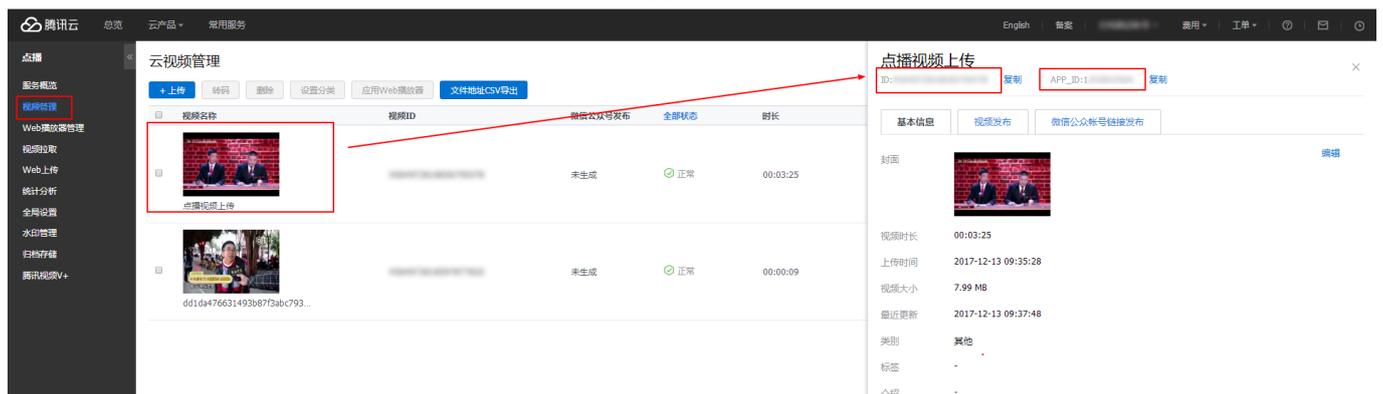
在 [腾讯云官网](#) 注册腾讯云帐号，然后开通点播服务。

step 2 : 上传文件

点播服务开通之后，进入 [点播视频管理](#) 就可以上传新的视频文件，如果您没有开通点播服务，则无法操作这一步骤。

step 3 : 获取 fileID 与 appID

上传完视频并转码之后，您就可以在视频管理页面查到文件的 fileID，这个是播放器播放视频的最基本信息，同时您的 appID 也可以在视频管理页面查看到。下图中的两个 ID，左边一个是视频文件的 fileID，另一个是您的 appID。



初始化 Web 播放器

在准备工作完成后，通过以下 3 个步骤，您就可以在您的网页上添加一个视频播放器。

step 1：在页面中引入文件

在合适的地方引入播放器样式文件与脚本文件

```
<link href="//imgcache.qq.com/open/qcloud/video/tcplayer/tcplayer.css" rel="stylesheet">
<script src="//imgcache.qq.com/open/qcloud/video/tcplayer/tcplayer.min.js"></script>
```

注意事项：

- 如果需要在 Chrome Firefox 等现代浏览器中通过 HTML5 播放 hls，需要在 tcplayer.min.js 之前引入 hls.js。

```
<script
src="//imgcache.qq.com/open/qcloud/video/tcplayer/lib/hls.min.0.8.8.js"></script>
```

step 2：放置播放器容器

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（容器 ID 以及宽高都可以自定义）。

```
<video id="player-container-id" width="414" height="270" preload="auto" playsinline webkit-
playinline x5-playinline>
</video>
```

注意事项：

- 播放器容器必须为

```
<video>
```

标签。

- 示例中的 `player-container-id` 为播放器容器的ID，可自行设置。
- 播放器容器区域的尺寸，建议通过 CSS 进行设置，通过 CSS 设置比属性设置更灵活，可以实现例如铺满全屏、容器自适应等效果。
- 示例中的 `preload` 属性规定是否在页面加载后载入视频，通常为了更快的播放视频，会设置为 `auto`，其他可选值：`meta`（当页面加载后只载入元数据），`none`（当页面加载后不载入视频），移动端由于系统限制不会自动加载视频。
- `playsinline` `webkit-playinline` `x5-playinline` 这几个属性是为了在标准移动端浏览器不劫持视频播放的情况下实现行内播放，此处仅作示例，请按需使用。

step 3：初始化代码

在页面初始化的代码中加入以下初始化脚本，传入在准备工作中获取到的 `fileID` 与 `appID`。

```
var player = TCPlayer('player-container-id', { // player-container-id 为播放器容器ID，必须与html中一致
  fileID: '4564972818956091133', // 请传入需要播放的视频fileID 必须
  appID: '1253668508' // 请传入点播账号的appID 必须
});
```

注意事项：

- 需要播放的视频需经过腾讯云转码，原始视频无法保证在浏览器中正常播放。

完整的示例页面：

[示例代码链接](#)

功能使用说明

下面将对播放器的部分功能进行详细说明，包括最佳实践与注意事项。

播放器尺寸设置

这里介绍几种给播放器设置尺寸的方法

- 可以给

```
<video>
```

标签设置 width height 属性，width height 的属性值是以像素计量的（比如 width = "100px" 或 width= 100），不能设置百分比。

- 可以通过 CSS 设置尺寸，支持像素、百分比等类型的值（比如：width:"100px" 或 width:"100%"）。

如果不设置宽高，播放器在获取到视频的分辨率后，将会以视频的分辨率设置播放器的显示尺寸，如果浏览器的可视区域的尺寸小于视频分辨率，会造成播放器区域超出浏览器的可视区域，所以通常不建议这样做。最佳实践为通过 CSS 设置播放器的尺寸。

熟练运用 CSS 可以实现铺满全屏、容器自适应等效果。

示例：

[CSS 设置尺寸](#)

[铺满网页可视区域](#)

[等比率自适应](#)

播放多种清晰度

1、首先需要在控制台设置转码多种清晰度，如下图



2、
视频转码后，将会生成多种清晰度的文件，在【控制台】>【视频管理】视频列表中单击视频将会看到如下图

源文件URL列表

名称	解码率	
原始	1495kbps	预览播放 显示源地址
mp4 标清	562kbps	预览播放 显示源地址
mp4 高清	1545kbps	预览播放 显示源地址
mp4 超高清	2734kbps	预览播放 显示源地址

经过以上两个步骤，该视频已具备多种清晰度，使用 fileID appID 在腾讯云点播播放器中播放即可。
清晰度选择效果如下图：



注意事项：

- 在浏览器劫持视频播放的情况下，该功能无法使用。通常在移动浏览器中，浏览器会劫持视频播放，用浏览器自带的播放器替代。

示例：

[多种清晰度](#)

指定播放清晰度

这里分为两种情况：指定播放某个清晰度和让播放器默认播放某个清晰度

指定播放某个清晰度

通过播放器的 definition 参数指定播放某个清晰度

可选值	说明
10	MP4 手机
20	MP4 标清

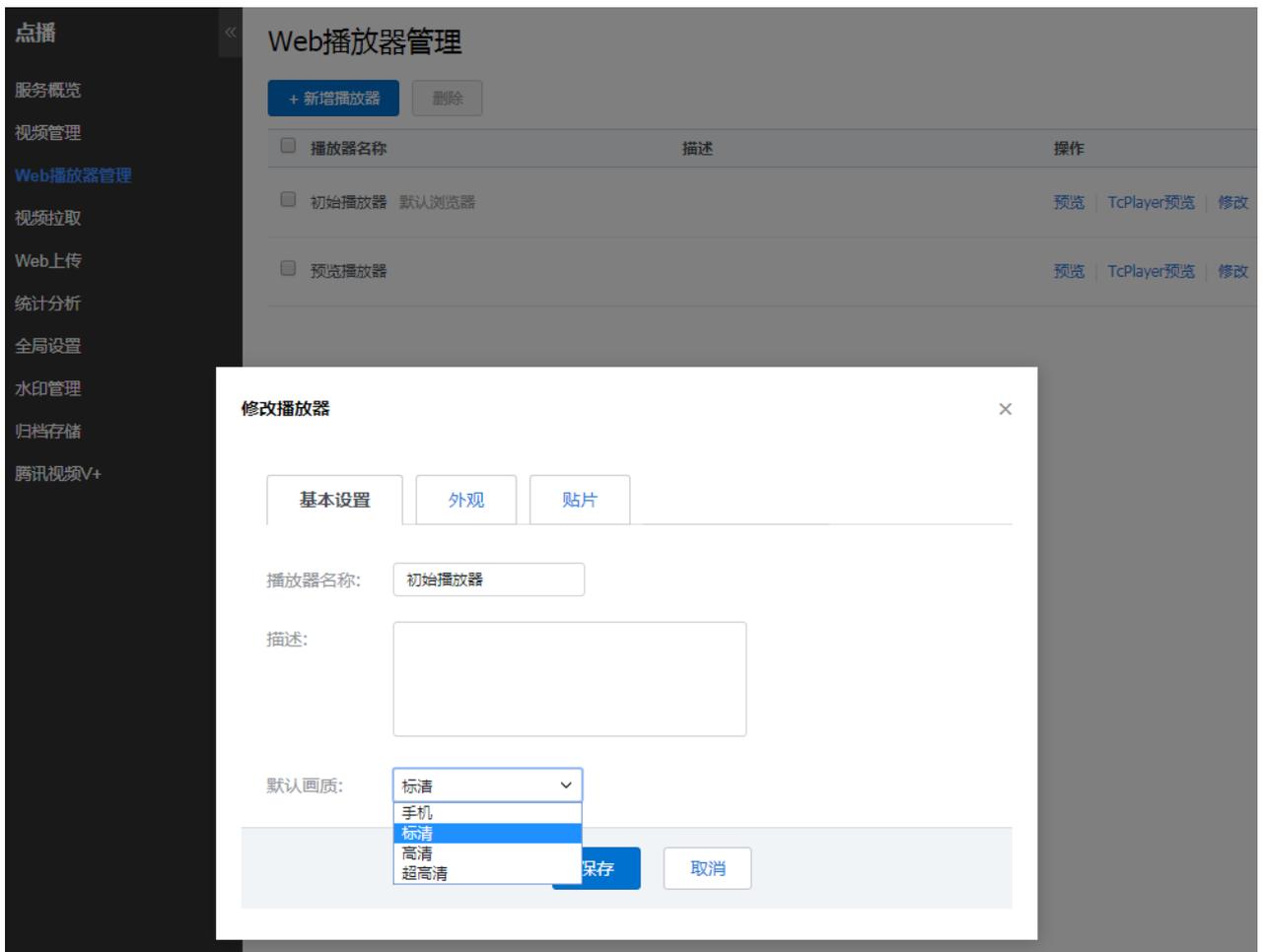
可选值	说明
30	MP4 高清
40	MP4 超清
210	HLS 手机
220	HLS 标清
230	HLS 高清
240	HLS 超清

在下面的示例中，将指定播放 MP4 手机清晰度视频：

[指定播放清晰度](#)

让播放器默认播放某个清晰度

- 在“控制台-Web 播放器管理”选定某个播放器配置进行设置默认画质



- 在“控制台-视频管理”将视频与某个播放器配置进行关联，在使用腾讯云播放器播放该视频时，将会

使用关联的播放器配置。



注意事项：

- 如果默认清晰度的视频不存在，则获取该视频的清晰度列表中第一个文件进行播放。比如播放器配置默认播放超高清，但是视频只有标清和高清，这时会播放标清视频。
- 控制台播放器配置在设置后，大概需要10分钟使所有 CDN 节点生效该配置。
- 在浏览器劫持视频播放的情况下，该功能无法使用。

续播功能

开启续播功能的前提，必须通过 fileID 播放。有了唯一的 fileID，播放器才能记录该视频的播放时长，当播放未结束时关闭页面，在同一个浏览器中再次打开播放页面，可从上次观看的时间点继续播放。

通过以下参数开启续播功能：

```
var player = TCPlayer('player-container-id', {
  fileID: "", // 请传入需要播放的视频 fileID 必须
  appID: "", // 请传入点播账号的 appID 必须
  plugins:{
    ContinuePlay: { // 开启续播功能
      // auto: true, //[可选] 是否在视频播放后自动续播
      // text:'上次播放至', //[可选] 提示文案
    }
  }
});
```

```
// btnText: '恢复播放' // [可选] 按钮文案
},
}
});
```

开启成功后将会看到的效果如下图：



示例：

[续播](#)

注意事项：

- 必须通过 fileID appID 播放经过腾讯云转码后的视频，才能使用该功能。
- 该功能通过通过 localStorage 存储播放时间点，浏览器需支持该特性。
- 在浏览器劫持视频播放的情况下，该功能无法使用。
- 该功能不是多端多浏览器互通的，比如在 PC 浏览器上没看完，不能在移动端浏览器上续播或者在 PC 上另一个浏览器续播，需额外的接口，可以自行开发。

设置播放器 logo

腾讯云点播播放器支持配置播放器 logo，可以在【控制台】>【Web 播放器管理】选定某个播放器配置，单击外观栏目进行设置 logo 信息。设置 logo 信息后，使用该播放器配置播放视频时，将会在指定位置显示 logo。

示例：

[显示 Logo](#)

注意事项：

- 控制台播放器配置在设置后，大概需要10分钟使所有 CDN 节点生效该配置。
- 在浏览器劫持视频播放的情况下，设置的 logo 将无法显示。

图片贴片功能

腾讯云点播播放器支持配置片头、片中暂停、片尾显示图片贴片,并且可以添加超链接。可以在【控制台】>【Web 播放器管理】选定某个播放器配置，单击贴片栏目进行设置贴片信息。

默认的贴片显示样式为水平垂直居中显示，如果图片的尺寸大于播放器显示区域，将按播放器的宽度等比缩放图片，水平居中显示图片，图片超出播放器区域部分将无法显示。

可以通过 CSS 自定义贴片的显示样式。

```
.tcp-image-patch-start{} /* 片头贴片样式Class */  
.tcp-image-patch-pause{} /* 片中贴片样式Class */  
.tcp-image-patch-ended{} /* 片尾贴片样式Class */
```

示例：

[图片贴片](#)

注意事项：

- 贴片建议使用体积不超过50KB且尺寸不超过播放器显示区域的图片，避免因图片过大影响视频初始化速度。
- 控制台播放器配置在设置后，大概需要10分钟使所有 CDN 节点生效该配置。
- 在浏览器劫持视频播放的情况下，设置的贴片将无法显示。

Referer 防盗链

开启流程请看 [Referer 防盗链说明文档](#)

播放器初始化需增加参数如下：

```
var player = TCPlayer('player-container-id', {
  fileID: "", // 请传入需要播放的视频fileID 必须
  appID: "", // 请传入点播账号的appID 必须
  flash:{
    swf: '[[腾讯云隔离域名]/vod-player/[appID]/[fileID]/tcplayer/player.swf' //swf 文件地址 必须
  }
});
```

需传入 swf url，如果浏览器使用 Flash 播放，将会去这个地址获取 Flash 播放器。Flash 播放器发起视频请求时，请求的 Referer 会带上该 url 或者带上页面的 url。

注意事项：

- 播放器在 Flash 模式下发起视频请求的 Referer 在 IE、Firefox 浏览器中会带上 swf url，与 Chrome 浏览器会带上页面的 url 的情况不同。
- 您也可以将 player.swf 文件下载后，存放到您的 CDN 服务器中，swf 参数传入指向您的 CDN 服务器路径。
- 腾讯云提供的隔离域名是每个用户独有的域名，一个 appID 对应一个域名，通常格式为 [appID].vod2.myqcloud.com。

- 需要将播放器 swf url 的域名添加到白名单内，开启了Referer防盗链的视频才能在 Flash 模式下播放。
- 播放器的 Flash swf 文件默认存放在 imgcache.qq.com 域名下。

Key 防盗链

开启流程请看 [Key 防盗链说明文档](#)

播放器初始化需增加参数如下：

```
var player = TCPlayer('player-container-id', {  
  fileID: "", // 请传入需要播放的视频 filID 必须  
  appID: "", // 请传入点播账号的 appID 必须  
  t: "",  
  us: "",  
  sign:""  
});
```

参数 t、us、sign的具体含义请查看 [Key 防盗链说明文档](#)

注意事项：

- sign 的计算方法为： $sign = md5(KEY+appId+fileId+t+us)$ ，与 [Key 防盗链说明文档](#) 中的计算方式不同，其余参数一致。
- 如果同时开启了 Referer 防盗链，在 Referer 防盗链配置的示例代码基础上增加参数即可。

试看功能

使用试看功能需要先开启 Key 防盗链，开启流程请看 [Key 防盗链说明文档](#)

播放器初始化需增加参数如下：

```
var player = TCPlayer('player-container-id', {
  fileID: "", // 请传入需要播放的视频 fileID 必须
  appID: "", // 请传入点播账号的 appID 必须
  t: "",
  us: "",
  sign:"",
  exper:"
});
```

参数 t、us、sign、exper 的具体含义请查看 [Key 防盗链说明文档](#)

注意事项：

- 带试看的 sign 计算方法为： $\text{sign} = \text{md5}(\text{KEY} + \text{appId} + \text{fileId} + \text{t} + \text{exper} + \text{us})$ ，与 [Key 防盗链说明文档](#) 中的计算方式不同，其余参数一致。
- 播放器播放的视频时长是 exper 参数指定的长度，与已往在播放端控制播放时长的试看功能不同，播放器不会获取完整的视频。

HLS 加密播放

开启流程请看[视频加密文档](#)

注意事项：

- 如果播放页面或者Flash swf url 与解密密钥服务器域名不一致，Key 服务器需要部署 corssdomain.xml 和 CORS（"跨域资源共享" Cross-origin resource sharing），允许 Flash

和 JavaScript 跨域获取解密密钥。

- crossdomain.xml 中配置的是 swf url 的域名，并且 xml 文件必须放置在 Key 服务器的根目录。
- 播放器的 Flash swf 文件默认存放在 imgcache.qq.com 域名下。
- 视频只能进行一次加密，不可多次加密，严格按照视频加密文档操作。
- 解密密钥正确长度为16字节，起始和末尾位置不能有空白字符。

开发文档

前言

本文档介绍腾讯云视频点播服务的 Web 播放器的相关参数以及 API，需结合 [《使用文档》](#) 使用。本文档适合有一定 Javascript 语言基础的开发人员阅读。

初始化参数

播放器初始化需要传入两个参数，第一个为播放器容器 ID，第二个为功能参数对象。

```
var player = TCPlayer('player-container-id', options);
```

options 参数列表

options 对象可配置的参数：

名称	类型	默认值	说明
appID	String	无	必选
fileID	String	无	必选
width	String / Number	无	播放器区域宽度，单位像素，按需设置，可通过 CSS 控制播放器尺寸。
height	String / Number	无	播放器区域高度，单位像素，按需设置，可通过 CSS 控制播放器尺寸。
controls	Boolean	true	是否显示播放器的控制栏
poster	String	无	设置封面图片完整地址
autoplay	Boolean	false	是否自动播放
playbackRates	Array	[0.5, 1, 1.25, 1.5, 2]	设置变速播放倍率选项，仅 H5 有效
loop	Boolean	false	是否循环播放
muted	Boolean	false	是否静音播放

名称	类型	默认值	说明
preload	String	auto	是否需要预加载，有 3 个属性"auto"，"meta"，"none"，移动端由于系统限制，设置 auto 无效。
swf	String	无	Flash 播放器 swf 文件的 URL
posterImage	Boolean	true	是否显示封面
bigPlayButton	Boolean	true	是否显示居中的播放按钮（浏览器劫持嵌入的播放按钮无法去除）
language	String	"zh-CN"	设置语言
languages	Object	无	设置多语言词典
controlBar	Object	无	设置控制栏属性的参数组合，后面有详细介绍
plugins	Object	无	设置插件功能属性的参数组合，后面有详细介绍

controlBar 参数列表

controlBar 参数可以配置播放器控制栏的功能，支持的属性有：

名称	类型	默认值	说明
playToggle	Boolean	true	是否显示播放、暂停切换按钮
progressControl	Boolean	true	是否显示播放进度条
volumePanel	Boolean	true	是否显示音量控制
currentTimeDisplay	Boolean	true	是否显示视频当前时间
durationDisplay	Boolean	true	是否显示视频时长
timeDivider	Boolean	true	是否显示时间分割符
playbackRateMenuButton	Boolean	true	是否显示播放速率选择按钮
fullscreenToggle	Boolean	true	是否显示全屏按钮
QualitySwitcherMenuButton	Boolean	true	是否显示清晰度切换菜单

plugins 插件参数列表

plugins 参数可以配置播放器插件的功能，支持的属性有：

名称	类型	默认值	说明
ContinuePlay	Object	无	控制续播功能，支持的属性如下 auto: false, // 是否在播放时自动续播 text: '上次看到', // 提示文案 btnText: '恢复播放' // 按钮文案

对象方法

初始化播放器返回对象的方法列表：

名称	参数及类型	返回值及类型	说明
ready(function)	(Function)	无	设置播放器初始化完成后的回调
play()	无	无	播放以及恢复播放
pause()	无	无	暂停播放
currentTime(seconds)	(Number)	(Number)	获取当前播放时间点，或者设置播放时间点，该时间点不能超过视频时长
duration()	无	(Number)	获取视频时长
volume(percent)	(Number)[0,1][可选]	(Number) / 设置时无返回	获取或设置播放器音量
poster(src)	(String)	(String) / 设置时无返回	获取或设置播放器封面
requestFullscreen()	无	无	进入全屏模式
exitFullscreen()	无	无	退出全屏模式
isFullscreen()	无	Boolean	返回是否进入了全屏模式
on(type, listener)	(String, Function)	无	监听事件

名称	参数及类型	返回值及类型	说明
one(type , listener)	(String, Function)	无	监听事件，事件处理函数最多只执行 1 次
off(type , listener)	(String, Function)	无	解绑事件监听
buffered()	无	TimeRanges	返回视频缓冲区间
bufferedPercent()	无	值范围[0,1]	返回缓冲长度占视频时长的百分比
width()	(Number)[可选]	(Number) / 设置时无返回	获取或设置播放器区域宽度，如果通过 CSS 设置播放器尺寸，该方法将无效
height()	(Number)[可选]	(Number) / 设置时无返回	获取或设置播放器区域高度，如果通过 CSS 设置播放器尺寸，该方法将无效
videoWidth()	无	(Number)	获取视频分辨率的宽度
videoHeight()	无	(Number)	获取视频分辨率的高度
dispose()	无	无	销毁播放器

注意事项：

- 部分方法不能同步调用

事件

播放器可以通过初始化返回的对象进行事件监听，示例：

```
var player = TCPlayer('player-container-id', options);
player.on(type, function);
```

其中 type 为事件类型，支持的事件有：

名称	介绍
play	已经开始播放，调用 play() 方法或者设置了 autoplay 为 true 且生效时触发，这时 paused 属性为 false
playing	因缓冲而暂停或停止后恢复播放时触发，paused 属性为 false 。通常用这个事件来标记视频真正播放，play 事件只是开始播放，画面并没有开始渲染
loadstart	开始加载数据时触发
durationchange	视频的时长数据发生变化时触发
loadedmetadata	已加载视频的 metadata
loadeddata	当前帧的数据已加载，但没有足够的数据来播放视频的下一帧时，触发该事件
progress	在获取到媒体数据时触发
canplay	当播放器能够开始播放视频时触发
canplaythrough	当播放器预计能够在不停下来进行缓冲的情况下持续播放指定的视频时触发
error	视频播放出现错误时触发
pause	暂停时触发
ratechange	播放速率变更时触发
seeked	搜寻指定播放位置结束时触发
seeking	搜寻指定播放位置开始时触发
timeupdate	当前播放位置有变更，可以理解为 currentTime 有变更
volumechange	设置音量或者 muted 属性值变更时触发
waiting	播放停止，下一帧内容不可用时触发
ended	视频播放已结束时触发。此时 currentTime 值等于媒体资源最大值
resolutionswitching	清晰度切换进行中
resolutionswitched	清晰度切换完毕

错误码

当播放器触发 error 事件时，监听函数会返回错误码。错误码列表：

名称	介绍
-1	没有视频地址
-2	获取视频数据超时
1	视频加载播放被中断
2	由于网络问题造成加载视频失败
3	解码时发生错误
4	视频因格式不支持或者服务器或网络的问题无法加载
5	解密时发生错误
10	点播服务接口请求超时
11	点播服务接口没有响应
12	点播服务接口返回异常数据
13	点播视频没有转码，需在点播控制台进行转码

播放器 SDK 下载

概述

目前，腾讯云点播为客户端提供了Android，iOS 和 Web 三种客户端 SDK。

Android SDK

[视频播放](#)

[SDK 下载地址](#)

iOS SDK

[视频播放](#)

[SDK 下载地址](#)

Web SDK

[视频播放](#)

Web SDK不需要下载，直接在页面中引入js即可，详情请见以上链接