

集成文档

保利威云点播Web播放器同时支持Flash和HTM5两种播放模式，并可以根据终端和浏览器环境自动选择最优播放模式，实现了视频播放效果的多平台统一体验。支持FLV、MP4、HLS等主流视频格式和协议，并结合保利威云点播服务，提供加密视频播放等功能。

适配情况

保利威云点播的转码服务，在默认情况下会把上传的视频转码出以下几种格式：

- 非加密视频：MP4（H5播放）、FLV（Flash播放）
- 加密视频：HLS（H5播放）、PTS（Flash播放）

视频格式与平台支持情况如下：

浏览器/播放器	Flash播放器（已停止更新）	H5播放器
Chrome	✓	✓
Firefox	✓	✓
Edge	✓	✓
IE	IE8+	IE11+ for Windows 8.1+
Safari	✓	✓
iOS Safari	✗	✓
iOS 微信、QQ	✗	✓
Android 微信、QQ	✗	✓
Android 其它	✗	✓

1. PC端IE11（Windows 8.1+）及以上浏览器支持HTML5播放，低版本IE将切换至FLASH播放器；移动端不支持Flash播放器。
2. 视频编码格式仅支持H.264编码。

3. 如果对视频设置了不转码的话，播放器会直接播放视频源文件，能否正常播放取决于观众浏览器支持的视频格式。

已知的H5适配问题

1. 设置自动播放无效，不会自动播放
 - 移动端浏览器中音频自动播放一直是禁止的，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），但不排除一些特性的浏览器和webview允许自动播放。
 - 桌面端Chrome 55+、Safari 11+版本的浏览器也都限制自动播放。
2. 移动端H5播放器自带的皮肤在部分情况下无法显示，导致无法使用播放器提供的部分控制功能（如画质/倍速切换等）。
 - iOS设备在全屏播放视频时，会强制使用系统的播放器，此时无法显示自带的播放器皮肤。
 - 由于Android系统有众多不同的定制版本，每个版本对H5 video标签的实现都有差别，而且大多数的国内浏览器厂商，都会在视频播放时挟持video标签，由浏览器内置播放器播放视频，所以在Android上播放视频的一致性相比iOS要差很多。在无法显示自带播放器皮肤情况下，可以通过播放器提供的API接口来进行播放控制。
3. 音量调节功能在iOS和Android浏览器上无效。
4. 倍速播放功能，部分移动端浏览器不支持设置倍速，功能会失效，比如android微信。

快速集成代码

不依赖于任何的前端js库，只需要在页面中引用如下js文件，就可以进行播放器的初始化。

js

```
<div id="player"></div>
<script src="//player.polyv.net/resp/vod-player/latest/player.js"></script>
<script>
  var player = polyvPlayer({
    wrap: '#player',
    width: 800,
    height: 533,
    vid: '88083abbf5bcf1356e05d39666be527a_8' ,
  });
</script>
```

在线运行

默认使用H5播放器，在ie及其他不支持的浏览器下自动切换flash

需要兼容ie9以下的播放器，可以使用下面代码（不再推荐，原因详见：[关于Flash Player即将停止服务的通知](#)）

```
<script src='../player.polyv.net/script/polyvplayer.min.js'></script>
<div id='player'></div>
<script>
var player = polyvObject('#player').videoPlayer({
  'width':'600',
  'height':'450',
  'vid' : '88083abbf5bcf1356e05d39666be527a_8' ,
});
</script>
```

vue

```
<template>
  <div id="player"></div>
</template>
<script>
export default {
  data() {
    return {
      vodPlayerJs: 'https://player.polyv.net/resp/vod-player/latest/player.js',
      vid: '88083abbf5bcf1356e05d39666be527a_8',
    };
  },

  mounted(){
    this.loadPlayerScript(this.loadPlayer);
  },

  methods: {
    loadPlayerScript(callback) {
      if (!window.polyvPlayer) {
        const myScript = document.createElement('script');
        myScript.setAttribute('src', this.vodPlayerJs);
        myScript.onload = callback;
        document.body.appendChild(myScript);
      } else {
        callback();
      }
    },

    loadPlayer() {
      const polyvPlayer = window.polyvPlayer;
      this.player = polyvPlayer({
        wrap: '#player',
        width: 800,
        height: 533,
        vid: this.vid ,
      });
    }
  },
  destroyed() {
    if (this.player) {
      this.player.destroy();
    }
  }
};
</script>
```

react

```
import React from 'react';

class Player extends React.Component {
  constructor(props) {
    super(props);
  }

  componentDidMount() {
    if(!window.polyvPlayer){
      this.loadScript('https://player.polyv.net/resp/vod-player/latest/player.js')
        .then(() =>{
          this.loadPlayer();
        });
    }
  }

  componentWillUnmount() {
    if(this.player){
      this.player.destroy();
    }
  }

  loadPlayer() {
    this.player = window.polyvPlayer({
      wrap: '.player',
      width: '100%',
      height: '100%',
      vid: '88083abbf5bcf1356e05d39666be527a_8',
    });
  }

  loadScript(src) {
    const headElement = document.head || document.getElementsByTagName('head')[0];
    const _importedScript = {};

    return new Promise((resolve, reject) => {
      if (src in _importedScript) {
        resolve();
        return;
      }
      const script = document.createElement('script');
      script.type = 'text/javascript';
      script.onerror = err => {
        headElement.removeChild(script);
        reject(new URIError(`The Script ${src} is no accessible.`));
      }
      script.onload = () => {
        _importedScript[src] = true;
        resolve();
      }
      headElement.appendChild(script);
      script.src = src;
    })
  }
}
```

```
    }

    render() {
      return (
        <div className="wrap">
          <div className="player"></div>
        </div>
      )
    }
  }
}

export default Player;
```

参数

参数	类型	默认值	说明
wrap	string / HTMLElement	-	页面上存在需要载入播放器的DOM元素或css选择器
width	number / string	100%	播放器的宽度，支持像素值和百分比两种类型
height	number / string	auto	播放器的高度，支持像素值和百分比两种类型
vid	string	-	云点播平台的视频唯一id

更多参数请查看 [属性和接口说明](#)。

其它分支版本

除了线上通用版本的播放器外，我们还提供一些其他分支版本的播放器。这些分支版本存在的原因可能有：

- 当前分支版本处于公测阶段，公测期过后会合并或替换通用版本。
- 分支版本具有的某些特性或采用的技术实现方案无法合并到通用版本。
- 分支版本针对某类场景进行了功能定制，只适用于部分客户。

目前可用的Web播放器分支版本有：

移动端H5播放器1.0版本

- 移动端H5播放器1.0版本目前已不再维护，建议使用2.0版本。

- 目前示例所用的均为2.0版本移动端H5播放器。
- 若需降级到移动端H5播放器1.0版本，可咨询保利威售后工程师。

VRM13版本播放器

- VRM13版本播放器采用了新的技术实现方案，大大提升了PC端H5播放器的安全性，[点此了解详情](#)。
- 不再兼容IE浏览器
- 播放器支持的参数、接口、事件与线上通用版本播放器保持一致，只需替换js文件的地址即可使用：由 `<script src="//player.polyv.net/resp/vod-player/latest/player.js"></script>` 替换为 `<script src="//player.polyv.net/resp/vod-player-drm/canary/player.js"></script>`。