

集成文档

保利威云直播Web播放器同时支持Flash和HTML5两种播放模式，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、RTMP、HLS等主流视频格式和协议，并结合保利威云直播服务，提供视频直播流流畅播放等功能。

快速集成

在线运行

js demo

```
<div id="player"></div>
<script src="//player.polyv.net/resp/live-h5-player/latest/liveplayer.min.js"></script>
<script>
  var player = polyvLivePlayer({
    wrap: '#player',
    width: 800,
    height: 533,
    uid: 'uid',
    vid: 'vid'
  });
</script>
```

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ie10以下的播放器，可以使用下面代码

```
<script src='https://player.polyv.net/livescript/liveplayer.js'></script>
<div id='player'></div>
<script>
var player = polyvObject('#player').livePlayer({
  width: '800',
  height: '533',
  uid: 'uid',
  vid: 'vid'
});
</script>
```

vue demo

```
<template>
  <div id="player"></div>
</template>
<script>
export default {
  data() {
    return {
      playerJs: 'https://player.polyv.net/resp/live-h5-player/latest/liveplayer.min.js',
      uid: 'uid',
      vid: 'vid'
    };
  },

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

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

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

react demo

```
import React from 'react';

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

  componentDidMount() {
    if(!window.polyvLivePlayer){
      this.loadScript('https://player.polyv.net/resp/live-h5-player/latest/liveplayer.min.js')
        .then(() =>{
          this.loadPlayer();
        });
    }
  }

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

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

  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;
```