Video.js 是一个通用的在网页上嵌入视频播放器的 JS 库，Video.js 自动检测浏览器对 HTML5 的支持情况，如果不支持 HTML5 则自动使用 Flash 播放器。

### 使用方法

在页面中引用video-js.cs样式文件和video.js

|  |  |
| --- | --- |
| 1  2 | <link href="video-js.css" rel="stylesheet" type="text/css">  <script src="video.js"></script> |

设置flash路径，Video.js会在不支持html5的浏览中使用flash播放视频文件

|  |  |
| --- | --- |
| 1  2  3 | <script>      videojs.options.flash.swf = "video-js.swf";  </script> |

****html****

poster="\*\*"播放初始图。可使用三种视频格式，根据所需要格式选择对应的。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <video id="example\_video\_1" class="video-js vjs-default-skin" controls preload="none" width="640" height="264"        poster="<http://video-js.zencoder.com/oceans-clip.png>"        data-setup="{}">      <source src="[http://](http://www.jq22.com/http:)视频地址格式1.mp4" type='video/mp4' />      <source src="[http://](http://www.jq22.com/http:)视频地址格式2.webm" type='video/webm' />      <source src="[http://](http://www.jq22.com/http:)视频地址格式3.ogv" type='video/ogg' />      <track kind="captions" src="demo.captions.vtt" srclang="en" label="English"></track><!-- Tracks need an ending tag thanks to IE9 -->      <track kind="subtitles" src="demo.captions.vtt" srclang="en" label="English"></track><!-- Tracks need an ending tag thanks to IE9 -->  </video> |

设置自动播放将下面代码加到html中代码后面

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <script type="text/javascript">      var myPlayer = videojs('example\_video\_1');      videojs("example\_video\_1").ready(function(){          var myPlayer = this;          myPlayer.play();      });  </script> |

默认情况下，大的播放按钮是被定为在左上角的，这样就不会覆盖视频内容。如果你想让这个播放按钮居中，你可以给你的 video 标签添加额外的 vjs-big-play-centered 样式，比如：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <video id="example\_video\_1" class="video-js vjs-default-skin vjs-big-play-centered"    controls preload="auto" width="640" height="264"    poster="<http://video-js.zencoder.com/oceans-clip.png>"    data-setup='{"example\_option":true}'>    ...  </video> |

如果你还对播放按钮样式不满意可重新定义.video-js .vjs-big-play-button{/\*这里的样式重写\*/}。

#### 为动态加载的 HTML 元素设置 Video.js

如果你的 web 页面或者应用是动态加载 video 标签的（ajax，appendChild，等等）,这样在页面加载后这个元素是不存在的，那么你会想要手动设置播放器而不是依靠 data-setup 属性。要做到这一点，首先将 data-setup属性从 video 标签中移除掉，这样在播放器初始化的时候就不会混乱了。接下来，运行下面的 javascript ，有时在 Video.js 加载后，有时是在 video 标签被加载进 DOM 后，

|  |  |
| --- | --- |
| 1  2  3 | videojs("example\_video\_1", {}, function(){    // Player (this) is initialized and ready.  }); |

videojs 方法中的第一个参数是你的 video 标签的 ID，用你自己的代替。

第二个参数是一个选项对象。它允许你像设置 data-setup 属性一样设置额外的选项。

第三个参数是一个 'ready' 回调。一旦 Video.js 初始化完成后，就会触发这个回调。

你也可以传入一个元素本身的引用来代替元素ID：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | videojs(document.getElementById('example\_video\_1'), {}, function() {    // This is functionally the same as the previous example.  });  videojs(document.getElementsByClassName('awesome\_video\_class')[0], {}, function() {    // You can grab an element by class if you'd like, just make sure    // if it's an array that you pick one (here we chose the first).  }); |

如果您无法播放内容，您得确保使用了 正确的格式，你的 HTTP 服务器可能无法提供正确的 MIME类型 的内容

var videoObj = videojs(“videoId”);

****ready:****

|  |  |
| --- | --- |
| 1  2  3  4 | myPlayer.ready(function(){      //在回调函数中，this代表当前播放器，      //可以调用方法，也可以绑定事件。  }) |

****播放：****

|  |  |
| --- | --- |
| 1 | myPlayer.play(); |

****暂停：****

|  |  |
| --- | --- |
| 1 | myPlayer.pause(); |

****获取播放进度：****

|  |  |
| --- | --- |
| 1 | var whereYouAt = myPlayer.currentTime(); |

****设置播放进度：****

|  |  |
| --- | --- |
| 1 | myPlayer.currentTime(120); |

****视频持续时间****，加载完成视频才可以知道视频时长，且在flash情况下无效

|  |  |
| --- | --- |
| 1 | var howLongIsThis = myPlayer.duration(); |

****缓冲****，就是返回下载了多少

|  |  |
| --- | --- |
| 1 | var whatHasBeenBuffered = myPlayer.buffered(); |

****百分比的缓冲****

|  |  |
| --- | --- |
| 1 | var howMuchIsDownloaded = myPlayer.bufferedPercent(); |

****声音大小****（0-1之间）

|  |  |
| --- | --- |
| 1 | var howLoudIsIt = myPlayer.volume(); |

****设置声音大小****

|  |  |
| --- | --- |
| 1 | myPlayer.volume(0.5); |

****取得视频的宽度****

|  |  |
| --- | --- |
| 1 | var howWideIsIt = myPlayer.width(); |

****设置宽度：****

|  |  |  |
| --- | --- | --- |
| 1 | myPlayer.width(640); |  |

****获取高度****

|  |  |
| --- | --- |
| 1 | var howTallIsIt = myPlayer.height(); |

****设置高度：****

|  |  |
| --- | --- |
| 1 | myPlayer.height(480); |

****一步到位的设置大小：****

|  |  |
| --- | --- |
| 1 | myPlayer.size(640,480); |

****全屏****

|  |  |
| --- | --- |
| 1 | myPlayer.enterFullScreen(); |

****离开全屏****

|  |  |
| --- | --- |
| 1 | myPlayer.enterFullScreen(); |

****添加事件****

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | durationchange  ended //播放结束  firstplay  fullscreenchange  loadedalldata  loadeddata  loadedmetadata  loadstart  pause //暂停  play  //播放  progress  seeked  seeking  timeupdate  volumechange  waiting  resize inherited    var myFunc = function(){      // Do something when the event is fired  }; |

****事件绑定****

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | myPlayer.on("ended", function(){      console.log("end", this.currentTime());  });  myPlayer.on("pause", function(){      console.log("pause")  }); |

****删除事件****

|  |  |
| --- | --- |
| 1 | myPlayer.removeEvent(“eventName”, myFunc); |

****一个页面中有多个视频****，需要点击后触发bootstrap 的模态窗，再弹出视频

我个人的方法是：

****html:****

|  |  |
| --- | --- |
| 1 | <a videohref="[http://xxx.mp4](http://xxx.mp4/)" class="video\_link"><img  src="../images/xxx.jpg"/></a> |

****JS:****

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | $(".video\_link").click(function() {      var myPlayer = videojs('my-video');      var videoUrl = $(this).attr("videohref");      videojs("my-video", {}, function() {          window.myPlayer = this;          $("#mymoda .video-con #my-video source").attr("src", videoUrl);          myPlayer.src(videoUrl);          myPlayer.load(videoUrl);          myPlayer.play();      });      $(".click-modal").click();  });  // 模态窗消失时，关闭视频  $('#mymoda').on('hidden.bs.modal', function() {      myPlayer.pause();  }); |