

点播上传参考 demo

推荐参考百家云提供的上传 API 自行实现上传：[上传视频/音频](#)

如需使用分块上传，可使用百家云 sdk 组件

1. `<!-- 引入 jquery -->`
2. `<script src="https://code.jquery.com/jquery-3.1.1.js"></script>`
3. `<!-- 引入百家云上传 sdk -->`
4. `<script src="https://live-cdn.baijiayun.com/js-sdk/2.25.0/uploader/VideoUploader.js"></script>`

或者使用vue版本的分块上传组件

npm地址：<https://www.npmjs.com/package/bjy-v-upload>

参考页面：[上传demo](#)

注意事项

- demo 中获取上传（getUploadUrl）和续传（getResumeUploadUrl）接口需要替换为客户调用百家云 openapi 的接口
<https://dev.baijiayun.com/wiki/detail/4#-h5-7>
<https://dev.baijiayun.com/wiki/detail/4#-h5-9>
- 确保 new BJY.VideoUploader 的时候 mainElement 对应的 input file dom 存在

- `uploader.currentFiles` 对应选择的文件列表，可以遍历 `files`，然后使用 `uploader.autoUpload(file)` 上传列表的每一个文件

demo 源码

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>上传视频</title>
5.   <meta charset="utf-8" />
6. </head>
7. <body>
8.
9.   <script src="https://code.jquery.com/jquery-3.1.1.js"></script>
10.  <script src="https://live-cdn.baijiayun.com/js-sdk/2.25.1/uploader/VideoUploader.js"></script>
11.
12.  <div class="bjy-demo">
13.    <div class="button upload-wrapper add-file-button">
14.      <input type="file">
15.      上传新视频
16.    </div>
17.
18.    <div class="button btn-stop">
19.      停止上传
20.    </div>
21.
22.    <div class="button btn-continue">
23.      继续上传
24.    </div>
25.    <div class="item-list"></div>
```

```
26. </div>
27.
28. <script>
29.     var uploader = new BJY.VideoUploader({
30.         mainElement: $('input[type="file"]'),
31.         ignoreError: true,
32.         name: "",
33.         multiple: true,
34.         useChunk: true,
35.         needSaveChunkInfoToLocalStorage: true,
36.         getUploadUrl: function (data) {
37.             // 此上传地址为 demo 调用，客户调用请使用
38.             // 用 openapi getUploadUrl 接口
39.             // https://dev.baijiayun.com/wiki/detail/4#-h5-7
40.             return $.ajax({
41.                 url:
42.                 'https://www.baijiayun.com/default/video_demo/get
43.                 type: 'post',
44.                 data: {
45.                     file_name: data.videoName,
46.                     definition: 1
47.                 }
48.             })
49.             .then(function (response) {
50.                 var result = response.data;
51.                 return {
52.                     id: result.video_id,
53.                     url: result.upload_url
54.                 };
55.             });
56.         },
57.         getResumeUploadUrl: function (data) {
58.             // useChunk true 时必须实现此方法获取断
59.             // 点续传的上传地址
60.             // 此上传地址为 demo 调用，客户调用请使用
61.             // 用 openapi getResumeUploadUrl 接口
```

```
https://dev.baijiayun.com/wiki/detail/4#-h5-9
57.         return $.ajax({
58.             url:
59.             'https://www.baijiayun.com/default/video_demo/get
60.             type: 'post',
61.             data: {
62.                 video_id: data.uploadId
63.             }
64.         })
65.         .then(function (response) {
66.             var result = response.data;
67.             if (!result) {
68.                 alert(response.msg);
69.                 return;
70.             }
71.             return {
72.                 fid: result.video_id,
73.                 id: result.video_id,
74.                 url: result.upload_url,
75.                 uploaded: result.upload_size
76.             };
77.         });
78.         onFileChange: function (files) {
79.             // 获取文件列表 使用
80.             uploader.autoUpload(file) 上传列表的每一个文件
81.             var currentFiles =
82.             uploader.currentFiles;
83.             // 遍历 currentFiles, 可以通过
84.             currentFiles[index].fileItem 获取到文件的信息
85.             // fileItem.status 0 未上传 1 上传中 2 上传
86.             完成 3 上传失败
87.             if (!currentFiles.length) {
88.                 return;
89.             }
90.         }
91.     }
92. }
```

```
87.         if (!uploader.validateFiles(currentFiles))
88.         {
89.             uploader.reset();
90.             return;
91.         }
92.         var itemHTML;
93.         $.each(
94.             currentFiles,
95.             function (index, file) {
96.                 // 上传完成的不继续上传
97.                 if (file.fileItem.status === 1 ||
98.                     file.fileItem.status === 2) {
99.                     return
100.                 }
101.                 itemHTML = '<div class="item">'
102.                     + '<span class="item-text
103.                     item-name">'
104.                     + file.videoName
105.                     + '</span>'
106.                     + '<span class="item-text
107.                     item-status">等待上传</span>'
108.                     + '</div>';
109.                 uploader.autoUpload(file);
110.                 $('<div class="item">'
111.                     + '<span class="item-text
112.                     item-name">'
113.                     + file.videoName
114.                     + '</span>'
115.                     + '<span class="item-text
116.                     item-status">等待上传</span>'
117.                     + '</div>');
```

```
118.         .eq(data.fileItem.index)
119.         .find('.item-status').html('上传' +
    data.percent + '>uploaded: ' + data.uploaded +
    '> total: ' + data.total + '> index : ' + (0 ||
    (data.fileItem && data.fileItem.chunk &&
    data.fileItem.chunk.index)));
120.     },
121.     onUploadSuccess: function (data) {
122.         // 上传成功回调
123.         console.log('onUploadSuccess',
    data.fileItem);
124.         $('<div class="item">
125.             .eq(data.fileItem.index)
126.             .find('.item-status').html('上传成功');
127.         },
128.         onChunkUploadError: function (data) {
129.             // 上传失败回调
130.             alert('上传错误, 请重新上传');
131.         },
132.         onUploadError: function (data) {
133.             // 上传出错回调
134.             alert('上传错误, 请重新上传');
135.             $('<div class="item">
136.                 .eq(data.fileItem.index)
137.                 .find('.item-status').html('上传失败');
138.             },
139.             onUploadComplete: function (data) {
140.                 // 上传完成回调
141.             },
142.             onError: function (data) {
143.                 alert(data.content);
144.             }
145.         });
146.         $('<div class="item">
147.             // uploader.stopFile 传入上传的 index 可以停
    止上传
```

```
148.     uploader.stopFile(0);
149.   });
150.   $('.btn-continue').click(function () {
151.     // 继续上传，开启分块上传后才能调用
152.     var data = uploader.currentFiles[0];
153.     uploader.resumeUpload(data);
154.   });
155.   // 删除文件，从 uploader.currentFiles 中选择需
   要删除的文件的 index 传入
156.   // uploader.removeFile(index);
157. </script>
158.
159. <div class="bjy-desc">
160.   <h3> VideoUploader 的相关配置: </h3>
161.
162.   <p> {jQuery} mainElement 主元素，即文件输
   入框元素</p>
163.   <p> {string} name 传给后端的 name</p>
164.   <p> {Function} getUploadUrl promise 返回
   { video_id: "", upload_url: "" }</p>
165.   <p> {Function} getResumeUploadUrl
   promise 返回 { uploaded: number, id: "", url: "" }
   </p>
166.   <p> {Function} onError 错误处理</p>
167.   <p> {boolean?} useChunk 是否分片上传
   </p>
168.   <p> {boolean?}
   needSaveChunkInfoToLocalStorage 是否需要将分片
   信息保存在本地（useChunk 为 true时建议开启）
   </p>
169.   <p> {number?} maxSize 上传文件的最大尺寸
   </p>
170.   <p> {string?} maxSizeError 超过上传文件的
   最大尺寸的错误信息</p>
171.   <p> {number?} resumeExpireTime 本地记录
   的分片信息的过期时间，默认是一天之后过期</p>
```

```
172.     <p> {number?} resumeUploadMaxCount 获取续传数据的最大尝试次数</p>
173.     <p> {number?} chunkUploadMaxCount 分片上传的最大尝试次数</p>
174.     <p> {Function?} validateVideo 校验视频，比如同时上传的最大数量，或者是否重复上传</p>
175.     <p> {Function?} isChunkUploadSuccess 判断分片是否上传成功，这属于业务逻辑</p>
176.     <p> {Function?} onResumeUploadWait 拉取到服务器的续传数据前，可能需要等待一段时间</p>
177.     <p> {Function?} onResumeUploadCancel 拉取不到服务器的续传数据，则会自动取消</p>
178.     <p> {Function?} onFileChange 选择文件之后触发</p>
179.     <p> {Function?} onUploadStart 开始上传的时候触发</p>
180.     <p> {Function?} onUploadProgress 获取文件的上传进度</p>
181.     <p> {Function?} onUploadSuccess 文件上传成功的时候触发</p>
182.     <p> {Function?} onUploadError 文件上传失败</p>
183.     <p> {Function?} onChunkUploadSuccess 分片上传成功</p>
184.     <p> {Function?} onChunkUploadError 分片上传失败</p>
185.     <p> {Function?} onUploadComplete 文件上传完成（无论上传成功还是失败之后）触发</p>
186. </div>
187.
188. <style type="text/css">
189.     .bjy-desc {
190.         width: 600px;
191.         position: absolute;
192.         right: 0;
193.         border: 1px solid #ccc;
```



```
194.     padding: 10px;
195.   }
196.   .bjy-demo {
197.     width: 500px;
198.     float: left;
199.   }
200.   .bjy-demo .button {
201.     background-color: #37A4F5;
202.     display: inline-block;
203.     color: #fff;
204.     padding: 10px 15px;
205.     vertical-align: top;
206.   }
207.   .bjy-demo .upload-wrapper {
208.     position: relative;
209.     overflow: hidden;
210.   }
211.   .bjy-demo .upload-wrapper object,
212.   .bjy-demo .upload-wrapper
213.     input[type="file"] {
214.     opacity: 0;
215.     filter: alpha(opacity=0);
216.     cursor: pointer;
217.     position: absolute;
218.     left: 0;
219.     top: 0;
220.     width: 100%;
221.     height: 100%;
222.     z-index: 2;
223.   }
224.   .item-list {
225.     font-size: 14px;
226.     color: #2e5580;
227.     padding: 10px;
228.     display: block;
229.     float: left;
```

```
229.     width: 100%;
230. }
231. .item-name {
232.     display: inline-block;
233.     margin-right: 20px;
234. }
235. </style>
236. </body>
237. </html>
```



下载为pdf格式