

1. 插件配置

原 <https://ext.dcloud.net.cn/plugin?id=10485> 和 <https://ext.dcloud.net.cn/plugin?id=10483> 已废弃，请勿使用。

本插件地址 <https://ext.dcloud.net.cn/plugin?id=11759>

HBuilderX 云端依赖插件后，需要配置百家云个性域名



2. Android 配置

2.1. 在 manifest.json 下配置

```
1. "buildFeatures" : {  
2.     "dataBinding" : true,  
3.     "viewBinding" : true  
4. }
```

2.2. 如遇上 so 库冲突，在 manifest 下配置

```
1. "packagingOptions": [           //可选，字符串数组类型，Android平台云端打包时build.gradle的packagingOptions配置项
2.     "pickFirst 'libs/*/xxx.so'",
3.     "merge '**/LICENSE.txt'"
4. ],
```

2.3. 插件依赖冲突

如遇到插件之间或者插件与 HBuilderX 默认依赖冲突，请参考社区的这个帖子解决。<https://ask.dcloud.net.cn/article/39280>

3. Vue 调用

3.1. 直播

```
1. // 引用直播模块
2. const live = uni.requireNativePlugin("BJY-SDK-LiveUI")
3.
4.
5. 签名字段说明：
6. name: 用户昵称
7. number: 用户唯一标识
8. type: 0:学生 1:老师 2:助教
9. avatar: 用户图像 url 地址
10. groupId: 分组号，默认0不分组，只有分组直播才用到，不分组则不需要传此参数
11. // 进入标准三分屏模板界面（参加码方式）
12. startLiveTripleTemplateWithCode() {
```

```
13.     live.startLiveTripleTemplateWithCode("2g8hk8",
14.     /** 签名进教室
15.
16.     live.startLiveTripleTemplateWithSign("roomId",
17.     "sign", {
18.         "name": "yjm",
19.         "number": "12345",
20.         "type": 0,
21.         "avatar": "",
22.         "groupid", 0
23.     })
24.     /**/
25. }
26. // 进入企业竖屏模板界面
27. startLiveEETemplateWithCode() {
28.     live.startLiveEETemplateWithCode("epfjsf",
29.     "yjm")
30.     /** 签名进教室
31.
32.     live.startLiveEETemplateWithSign("roomId",
33.     "sign", {
34.         "name": "yjm",
35.         "number": "12345",
36.         "type": 0,
37.         "avatar": "",
38.         "groupid", 0
39.     })
40.     /**/
41. },
42.
43. // 进入直播带货模板界面
44. startLiveSellTemplateWithCode() {
```

```

41.     live.startLiveSellTemplateWithCode("epfjsf",
    "yjm")
42.     /** 签名进教室
43.
    live.startLiveSellTemplateWithSign("roomId",
    "sign", {
44.         "name": "yjm",
45.         "number": "12345",
46.         "type": 0,
47.         "avatar": "",
48.         "groupId", 0
49.     })
50.     **/
51. },
52.
53. // 进入专业小班课模板界面
54. startProfessionalLiveTemplateWithCode() {
55.
    live.startProfessionalLiveTemplateWithCode("epfjsf"
    "yjm")
56.     /** 签名进教室
57.
    live.startProfessionalLiveTemplateWithSign("roomId",
    "sign", {
58.         "name": "yjm",
59.         "number": "12345",
60.         "type": 0,
61.         "avatar": "",
62.         "groupId", 0
63.     })
64.     **/
65. }

```

3.2. 在线点播回放

```
1. // 引用点播回放模块
2. const playback = uni.requireNativePlugin("BJY-
   SDK-PlaybackUI")
3.
4. // 进入点播界面
5. startVideo() {
6.     // VideoConfig 参数按需配置，如不需要设置，请传空对象 '{}'
7.     playback.startVideo("160491264",
   "W7zE72DM4167ltDCxPWSwbXbP8gWUnGsUHbgqb
   {
8.         "userId": "853145204",
9.         "userName": "yongjiaming",
10.        "supportLooping": true,
11.        "supportBreakPointPlay": true,
12.        "isLandscape": true
13.    })
14. }
15.
16. // 进入普通回放
17. startPlayback() {
18.
19.     playback.startPlayback("22120569887615",
   "FQj0pl0VChB5q0vfU4E4B08e8lOPKvV6OrdV7ckmS
   null) }
20.
21. // 进入长期课回放
22. startPlayback2() {
23.
24.     playback.startPlayback("22112282539916",
   "Q0XEwmsRDdt5q0vfU4E4B0MFS1SOSH3P4kjMRfxl
   17zPPAbfyY4TqPVTaj4Kp0fXMnVKLQ", {
25.         "session_id": "202212080"
```

```

26.         })
27.     },
28.
29. // 进入长期课裁剪版本回放
30. startPlayback3() {
31.
32.     playback.startPlayback("22092788575540",
33.                             "XFrvfzrsijh5q0vfU4E4B5G-
34.                             Dj0S3FQS0TFaRSpvjWf9mEaeSMauHajPy2Z0h_U_5F
35.                             {
36.                             "session_id": "202212220",
37.                             "clipedVersion": 1
38.                         })
39.     }

```

3.3. 下载

```

1. // 引入下载模块
2. const downloadManager =
3.     uni.requireNativePlugin("BJY-SDK-Download")
4.
5. // 初始化 DownloadManager
6. downloadManager.initModule()
7.
8. // 获取下载记录，返回对象为 JSONArray，每个
9. // JSONObject 对应一条下载记录
10.
11. /**
12.  * {
13.  *   // 视频分辨率，-1未知类型；0标清；1高清；2超
14.  *   // 清；3，720P；4，1080P；5 音频
15.  *   "definition": 2,
16.  *   // 房间号，仅回放有值，点播为空
17.  *   "roomId": "22112282539916",
18.  *   // 0 点播下载类型，1 回放下载类型
19.  * }
20.  */

```

```

15.     "type": 1,
16.     // 下载文件大小, 单位 byte
17.     "downloadedLength": 4016792,
18.     // 下载状态, 0初始状态; 1下载中; 2暂停; 3出
    错; 4完成; 5已取消
19.     "status": 4,
20.     // 回放长期课专有
21.     "sessionId": "202307050",
22.     // 视频 id
23.     "videoid": "190660972",
24.     // 总文件大小
25.     "totalLength": 4016792,
26.     // 视频时长, 单位秒
27.     "duration": 1233
28. }
29. */
30.
31. downloadManager.getAllDownloadInfo((jsonObj)
    => {
32. for (let i = 0; i < jsonObj.length; i++) {
33.     const element = jsonObj[i];
34.     console.log("第" + i + "个下载: " +
        JSON.stringify(element));
35. }
36. })
37.
38. /**
39.  * 下载点播视频
40.  * videoid: 视频 id
41.  * token: 视频 token
42.  * callback: 回调, 类型为 JsonObject
43.  * JsonObject 字段如下
44.  * {
45.     event: 回调事件类型(String类型), 分别为
    onProgress、onError、onPaused、onStarted、
    onFinish、onDeleted

```

```
46. 其它字段同下载记录中列举的
47.  *
48.  */
49.
50.
51. downloadManager.downloadVideo("189522041",
52. "gSXewR2YNDpjXPso7lvWEIXKsSLI9AhOwO3UNMRV
53. ", (jsonObj) => {
54.     this.videoDownloadTips =
55.     jsonObj.event
56.     if (jsonObj.event == "onProgress")
57.     {
58.         this.videoDownloadTips +=
59.         jsonObj.downloadedLength * 1.0 /
60.         jsonObj.totalLength
61.     }
62.     // 下载完成后自动播放
63.     if (jsonObj.event == "onFinish") {
64.         playback.startLocalVideo(jsonObj.videoId,
65.         jsonObj.videoPath, jsonObj.definition, {})
66.     }
67. })
68.
69. /**
70.  * 下载回放
71.  * roomId: 房间号
72.  * sessionId: 长期课id, 非长期课传 0
73.  * token: token
74.  * callback: 回调, 类型为 JsonObject
75.  * 字段同点播
76.  */
77. downloadManager.downloadPlayback("2211228253
78. "202307050",
```

```
71. "CQNfKDoGFqIjzZSuP5BQeK9UE5HaKY3CQbjRozbU
    (
72.         jsonObj) => {
73.             this.playbackDownloadTips =
                jsonObj.event
74.             if (jsonObj.event == "onProgress")
                {
75.                 this.playbackDownloadTips +=
                    jsonObj.downloadedLength * 1.0 /
                    jsonObj.totalLength
76.             }
77.             // 下载完成后自动播放
78.             if (jsonObj.event == "onFinish") {
79.
                playback.startLocalPlayback(jsonObj.roomId,
                jsonObj.sessionId, jsonObj.videoPath, jsonObj
80.                    .signalPath,
                jsonObj.definition, {})
81.            }
82.        })
83.
84.
85. // downloadVideo内部调了start，调了
    downloadVideo 或者 downloadPlayback 之后才可以
    调 pause()、start()、cancelAndDelete()
86. // 开始下载
87. downloadManager.start("videoid");
88. downloadManager.start("roomId", "sessionId");
89. // 暂停下载
90. downloadManager.pause("videoid");
91. downloadManager.pause("roomId", "sessionId");
92. // 取消并删除下载文件
93. downloadManager.cancelAndDelete("videoid");
94. downloadManager.cancelAndDelete("roomId",
    "sessionId");
```

3.4. 离线点播回放

vue 调用

请确保**videoid/RoomId** 有对应的下载记录，否则无法正常调起播放页面

```
1. // 播本地点播
2. playback.startLocalVideo(jsonObj.videoid, {})
3. // 播本地回放
4. playback.startLocalPlayback(jsonObj.roomId,
    jsonObj.sessionId, {})
```

插件代码

```
1. /**
2.  * 播放离线点播
3.  *
4.  * @param videoid 视频 id
5.  * @param videoConfig 视频配置项
6.  */
7. @UnijSMethod(uiThread = true)
8. public void startLocalVideo(String videoid,
9.    JSONObject videoConfig)
10.
11. /**
12.  * 播放离线回放
13.  * @param roomId 房间号
14.  * @param sessionId 长期课 sessionId
15.  * @param videoConfig 视频配置
16.  */
17. @UnijSMethod(uiThread = true)
```

```
18. public void startLocalPlayback(String roomId,  
String sessionId, String videoPath, String  
signalPath, int definition, JSONObject  
videoConfig)
```



下载为pdf格式