

## 概述

---

小程序 sdk 提供一套教室内的组件，供用户灵活组合自己的业务逻辑。

说明：由于微信会对小程序做不定期规则调整，为保证小程序业务正常，请关注与更新小程序 SDK 版本。

### SDK 下载地址

### 小程序插件使用方式

## 使用步骤

---

### 1) 准备工作

SDK 代码地址: <http://git2.baijiashilian.com/open-frontend/live-weixin>

- 1) sdk 放到项目根目录下(不要改文件名字和路径)
- 2) 使用组件的页面先注册组件
- 3) 开通小程序的 live-player 与 live-pusher 组件的使用权限。  
(重要)

[详情戳这里](#)

- 4) 页面 onload 以后调用 b jy.init()



5) 页面 onUnload 以后或者退出直播间请调用 `bjy.exit()` 销毁连接

6) 小程序后台配置 request 合法域名、socket 合法域名

微信小程序需要支持 https 和 wss 协议，需要配置的如下：

### request 合法域名

1. <https://www.baijiayun.com>
2. 启用专属域名的客户请配置为专属域名。
3. (专属域名从百家云账号中心获取。如：  
<https://demo.at.baijiayun.com>)

### socket 合法域名

根据对接账号所处环境，配置对应的 socket 合法域名，新注册账号为 latest 环境。

pro	pro2(ms03)	
pro-signal.baijiayun.com	pro-video-ms03.baijiayun.com	late cs.ba
pro-video-ms.baijiayun.com	pro-signal-bj01.baijiayun.com	late ms.ba
pro-video-cs.baijiayun.com	pro-video-cs01.baijiayun.com	late rs.ba
pro-video-rs.baijiayun.com	pro-video-rs-bj05.baijiayun.com	signal.l
pro-video-rs-bj02.baijiayun.com	pro-video-rs-bj06.baijiayun.com	
pro-video-rs-bj03.baijiayun.com	pro-video-rs-bj07.baijiayun.com	

## 2) bgy.init 方法

init 方法会初始化教室相关的服务器连接，封装信令层，初始化事件，并提供一些接口。

代码如下

```
1.
2. bgy.init({
3.   env: "", // 环境，可选值 test/beta/"，对于客户来
   说，不传默认线上环境
4.   sign: "", //签名
5.   noChat: false, // 当不需要聊天功能时，可以通过
   noChat: true不连接百家云的聊天服务器，少一条
   websocket连接,默认值为false
6.   hasDefaultTip: true, // 支持配置不要直播间的默认
   提示hasDefaultTip: false，默认值为true
7.   apiOrigin: "", //专属域名，专属域名从百家云账号中
   心获取。如： https://demo.at.baijiayun.com
8.   class: {
9.     id: "", //教室 id，需要在百家云后台开课，开完课
   后，把课程 id 传入，
10.    // 如果需要在后续用到教室相关的信息，可像这样
   写在下面
11.    // 如果不用，可省略
12.    name: '课程名称',
13.    startTime: '开始时间，精确到毫秒的时间戳',
14.    endTime: '结束时间，精确到毫秒的时间戳'
15.  },
16.  // 当前进入教室的用户的信息
17.  // 如果是学生进教室，传学生的信息
```

```

18. // number 必须确保是唯一的，如果不唯一，后进的
    用户会踢掉先进的用户
19.   user: {
20.     number: '用户 number',
21.     avatar: '头像地址',
22.     name: '用户昵称',
23.     type: '用户类型 0-学生,
24.     group: 0 // 学生分组信息，不传时候，默认为0，
        不分组
25.   },
26.   // 如果知道当前教室的老师信息，可传入，否则可省
        略
27.   teacher: {
28.     number: '老师 number',
29.     avatar: '头像地址',
30.     name: '老师昵称',
31.     type: 1
32.   }
33. });

```

同时支持参加码进入的方式

```

1. bjj.init({
2.   env: "", // 环境，可选值 test/beta/"，对于客户来
        说，不传默认线上环境
3.   apiOrigin: "", // 专属域名，专属域名从百家云账号中
        心获取。如： https://demo.at.baijiayun.com
4.   hasDefaultTip: true, // 支持配置不要直播间的默认
        提示hasDefaultTip: false，默认值为true
5.   class: {
6.     id: "", // 教室 id，需要在百家云后台开课， 开完课
        后，把课程 id 传入，
7.   },
8.   user: {
9.     number: '用户 number',

```

```
10.     avatar: '头像地址',
11.     name: '用户昵称',
12.     type: '用户类型 0-学生,
13.     group: 0 // 学生分组信息, 不传时候, 默认为0,
        不分组
14.   }
15. });
```

退出直播间请调用 **bjy.ext()** 销毁直播连接

### 3) 调用组件, 实现功能

组件具体使用如下:

## loading 组件

可用于进入教室的服务器连接加载等待

### 参数

参数	类型	必填	默认	说明
options	Object	否	{}	配置信息
roomLoadEnded	Event	否	空	进入教室成

options 默认值

```
1. {
2.   showInfo: true, // 是否显示加载百分比
3.   strokeWidth: 6, // 进度条的高度
4.   activeColor: '#1694ff', // 已经加载部分颜色
5.   backgroundColor: '#d8d9d8' // 进度条背景颜色
6. }
```

## 示例

1. `<loading class="loading-wrapper" options="{options}"`
2. `bind:loadEnded="roomLoadEnded">`
3. `</loading>`

## teacherPlayer 组件(模拟器不能播放流, 查看效果请用真机! 用真机! )

- 1) 老师视频播放器, 专门用于老师视频的播放, 相对于 player 组件, teacherPlayer 组件有老师音视频状态变化提示等区分学生的行为。
- 2) 组件宽高, 自适应容器宽高, 建议宽高比为 4:3

## 参数

参数	类型	必填	默认	说明
optTriggerType	string	否	tap	触发操作 (tap none)
stopPlay	Boolean	否	false	停止播放
orientation	string	否	vertical	方向, playe
objectFit	string	否	contain	适应相 信live
showFullScreen	boolean	否	true	是否显

## 事件

## teacherPlayerTap

用户操作触发

参数 event

event.detail.user 当前 player 的 user 信息

event.detail.support 当前流是否可以支持播放

## AVStatusChange

音频，视频状态变化触发

参数

event.detail.support 当前流是否在小程序可播放

event.detail.user 用户信息

event.detail.changeInfo={audioTo: true,videoTo: false}

1. changeInfo有audioTo字段，代表音频状态有变化  
audioTo==true,表示音频从关闭到打开，  
audioTo==false,表示音频从打开到关闭
2. changeInfo有videoTo字段，代表视频状态有变化  
videoTo==true,表示视频从关闭到打开，  
videoTo==false,表示视频从打开到关闭

## isSupportedChanged

小程序是否可以播放的状态变化，当推流方用网页时候，小程序不支持播放

参数

event.detail.user 用户信息

event.detail.support 是否可以播放

## 示例

```
1. <teacherPlayer
2.
3.     bind:playerTap="onPlayerOptTap"
4.
5.     bind:isSupportedChanged="onTeacherPlayerSupport
6.
7.     bind:AVStatusChange='onTeacherPlayerAVStatusCh
8. >
9. </teacherPlayer>
```

## 拉流事件监听

```
1. // 通过事件监听获取比例 bjs.init 之前监听
2. // 用户开关摄像头会触发下面的事件
3. eventEmitter.on(eventEmitter.MEDIA_PUBLISH,
4.     function (event, data) {
5.         if (data.videoOn || data.audioOn) {
6.             // 正在进行媒体播放
7.             console.log('宽高比例' + (data.width /
8.                 data.height))
9.         } else {
10.            // 老师关闭摄像头
11.        }
12.    })
13. // 屏幕共享和媒体播放会触发下面的事件
14. eventEmitter.on(eventEmitter.ASSIST_MEDIA_PUBLI
15.     function (event, data) {
16.         if (data.videoOn || data.audioOn) {
17.             // 正在进行媒体播放
18.             console.log('宽高比例' + (data.width /
19.                 data.height))
20.         } else {
21.            // 停止媒体播放 还原播放器原始比例
```

```
19.   }  
20. })
```

## userPlayer 组件(模拟器不能播放流，查看效果请用真机！用真机！)

学生发言视频播放。

- 1) 学生发言人数不确定。
- 2) 由于小程序的live-player是原生组件，不能放到scroll-view里面滚动，所以在学生发言人数过多，指定容器（styleInfo.wrapWidth）放不下时候，会将视频的默认宽度变小，保持高度不变。当宽度缩小为1/2时，会将高度也折半，视频会变成两行显示。

### 参数

参数	类型	必填	默认	说明
excludeUsers	Array	否	[]	不显示的用户数组值为用户的id，如 [122,345]
styleInfo	Object	否	详情	样式信息

styleInfo.wrapWidth 容器宽度，默认值 300

styleInfo.playerWidth 单个播放器的宽度，默认值 100

styleInfo.playerHeight 单个播放器的高度，默认值 75

styleInfo.fontSize 字体大小，默认值 12

### 事件

**onUserPlayerTap**

用户操作触发

参数 event

event.detail.user 当前 player 的 user 信息

event.detail.support 当前流是否可以支持播放

## AVStatusChange

音频，视频状态变化触发

参数

event.detail.support 当前流是否在小程序可播放

event.detail.user 用户信息

event.detail.changeInfo={audioTo: true,videoTo: false}

1. changeInfo有audioTo字段，代表音频状态有变化  
audioTo==true,表示音频从关闭到打开，  
audioTo==false,表示音频从打开到关闭
2. changeInfo有videoTo字段，代表视频状态有变化  
videoTo==true,表示视频从关闭到打开，  
videoTo==false,表示视频从打开到关闭

## isSupportedChanged

小程序是否可以播放的状态变化，当推流方用网页时候，小程序不支持播放

参数

event.detail.user 用户信息

event.detail.support 是否可以播放

## setUserList

设置用户信息时触发，可用于获取当前发言用户

event.detail.userList 当前显示的用户信息列表

event.detail.allUserList 所有发言用户的信息，包含通过 excludeUser 属性设置不显示的

## 示例

```
1. <userPlayer
2.   styleInfo="{{styleInfo}}"
3.   excludeUsers="{{excludeUsers}}"
4.   bind:setUserList="onSetUserList"
5.   bind:userPlayerTap="onUserPlayerTap"
6.
7.   bind:AVStatusChange="onAVStatusChange"
8.
9.   bind:isSupportedChanged="onIsSupportedChanged"
10. >
11. </userPlayer>
```

## whiteboard

课件组件,用户教室内的课件显示

### 参数

参数	类型	必填	默认	说明
hidePageCount	Boolean	否	true	是否显示
size	Object	否	屏幕宽, 高信息	宽度, size.w size.h
styleInfo	Object	否	{}	样式
drawing	Boolean	否	false	是否显示

## styleInfo 默认值

```
1.  
2. {  
3.   backgroundColor: 'black',  
4.   pageCountColor: 'white', // 页码信息字体颜色  
5.   pageCountBackground: '#C2C0C1' // 页码信息背  
   景色  
6. }
```

## 事件

### whiteboardTap

用户操作触发

参数 event

## 示例

```
1. <whiteboard id="whiteboard" class="  
   {{whiteboardClass}}"  
2.   hidePageCount="{{hidePageCount}}"  
3.   styleInfo="{{styleInfo}}"  
4.   width="{{width}}"  
5.   height="{{height}}"  
6.   drawing="{{drawing}}"  
7.   bind:whiteboardTap="onWhiteboardTap"  
8. >  
9. </whiteboard>
```

## messageList

聊天消息列表

用户教室内的聊天消息的显示

自适应容器宽高显示

## 参数

参数	类型	必填	默认	说明
maxCount	Number	否	500	消失的最大消息 消息超出会将最
historySize	Number	否	10	进教室请求的历
styleInfo	Object	否		自定义样式

styleInfo 默认值

```
1.  
2. {  
3.   messageBackground: 'white',  
4.   fromColor: '#9E9DA0',  
5.   contentColor: 'black'  
6. }
```

## messageSticker

聊天置顶组件，可参考 [pages/ui/messageList](#)

```
1. <view>  
2.   <messageSticker></messageSticker>  
3. </view>
```

## 事件

### addMessage

有新消息触发

参数 event.detail.message 消息信息

## 示例

```
1. <messageList  
   bind:addMessage="onAddMessage"  
2.   historySize="{{historySize}}"  
3.   maxCount="{{maxCount}}"  
4.   styleInfo="{{styleInfo}}">  
5. </messageList>
```

## messageSender

消息发送

## 参数

参数	类型	必填	默认	说
placeholder	String	否	''	输 的
messageMaxLength	Numnber	否	140	输
focus	Boolean	否	false	是

## 事件

### heightChange

组件高度变化，显示表情高度会发生变化

## 参数

event.detail.height 组件高度

event.detail.top 组件 top

event.detail.showEmotion 表情当前是否显示

sendMessage

发送消息事件

## 参数

event.detail.content 消息内容

## 示例

```
1. <messageSender placeholder="
   {{placeholder}}">
2.   messageMaxLength="
   {{messageMaxLength}}">
3.     heightChange="onHeightChange"
4.     sendMessage="onSendMessage"
5.     focus="{{focus}}">
6. </messageSender>
```

## notice

教室里通知显示

## 参数

参数	类型	必填	默认	说明
styleInfo	Object	否		自定义样式

styleInfo 默认值

```
1.
2. {
3.   titleColor: 'black',
4.   contentBackground: '#9E9DA0',
5.   contentColor: 'black'
6. }
```

## 示例

```
1. <notice styleInfo="{{styleInfo}}"></notice>
```

# questionAnswer

问答模块，仅支持提问

## 参数

参数	类型	必填	默认	说明
maxCount	Number	否	20	列表最多注： 一个问题作一条
inputMaxLength	Number	否	140	提问的字
limitPerMinute	Number	否	15	每分钟最

## 示例

```
1. <questionAnswer maxCount="{{maxCount}}">
```

```
2.     inputMaxLength="{{inputMaxLength}}"
3.     limitPerMinute="{{limitPerMinute}}">
4. </questionAnswer>
```

## linkSwitch

切换线路组件，如果配置了百家云多条线路，可以在直播中进行切换

### 示例

```
1. <linkSwitch></linkSwitch>
```

## 上麦功能

上麦（这里指能发布自己的音视频）分两种：

1 自动上麦，不需要主讲（老师）授权

使用 **publisher** 组件进行推流即可。摄像头、麦克风的开关控制可以通过自己的交互实现，只需要触发

**MEDIA\_SWITCH\_TRIGGER** 通知 **publisher** 组件即可。

关摄像头

```
1.     eventEmitter.trigger(eventEmitter.MEDIA_SWITCH_TRIGGER,
2.     {
3.         videoOn: false
4.     });
```

2 举手上麦，需要主讲（老师）同意以后才能发布上麦发布自己的音视频流。

- 实现方式一：举手的交互可以自己设计，触发 sdk 相应的事件即可。
- 实现方式二：用我们提供的 handup 组件

实现方式一的事件：

学生（观众）操作

1. 举手操作

2. [REDACTED]

3. [REDACTED]

```
eventEmitter.trigger(eventEmitter.SPEAK_START_TRIGGER)  
{
```

4. `timeout: 3000` // 举手后的等待响应时间，单位毫秒，该时间内老师未响应，视为超时，需要重新举手

5. `});`

1. [REDACTED]

2. 取消举手（在老师（主讲）处理举手之前，取消举手操作）

3. [REDACTED]

4. [REDACTED]

```
eventEmitter.trigger(eventEmitter.SPEAK_CANCEL_TRIGGER)
```

1. 终止上麦

2. [REDACTED]

3. [REDACTED]

```
eventEmitter.trigger(eventEmitter.SPEAK_STOP_TRIGGER)
```

4. [REDACTED]

5. // 如果用户处理上麦状态的时候，小程序后台了，此时需要终止发言状态，触发一下该事件

老师（主讲）处理结果，通过监听事件实现

```
1. 老师（主讲）同意上麦请求
2.
3. emitter.on(
4.
5. emitter.SPEAK_APPLY_RESULT_ACCEPT,
6. function () {
7. // 这里实现业务逻辑
8. }
9. )
```

```
1. 老师（主讲）拒绝上麦请求
2.
3. emitter.on(
4.
5. emitter.SPEAK_APPLY_RESULT_REJECT,
6. function () {
7. // 这里实现业务逻辑
8. }
9. )
```

```
1. 老师（主讲）超时未处理
2.
3. emitter.on(
4.
5. emitter.SPEAK_APPLY_RESULT_TIMEOUT,
6. function () {
7. // 这里实现业务逻辑
8. }
9. )
```

## publisher 组件(模拟器不能预览, 查看效果请用真机! 用真机! )

- 1) 本地视频采集推流。
- 2) 组件宽高自适应容器宽高
- 3) 由于小程序的限制, 只能推出 16: 9 的视频, 容器高度应该设置成 16: 9 比例, 否则本地预览可能不全

### 参数

参数	类型	必填	默认	说明
beauty	Number	否	0	美颜
whiteness	Number	否	0	美白
autoFocus	Boolean	否	true	自动对焦
backgroundMute	Boolean	否	false	背景静音
orientation	Boolean	否	vertical	竖屏
minBitrate	Number	否	200	最低码率
maxBitrate	Number	否	1000	最高码率
zoom	Boolean	否	false	缩放
styleInfo	Object	否	{fontSize: 12,color: 'black'}	样式

### 事件

**statusChange**

音视频状态变化 (老师 (主讲) 端有“结束发言”的操作, 也会导致用户音视频状态变化)

参数 event

event.detail.newValue 新的状态

event.detail.oldValue 旧状态

### **publisherTap**

tap 事件

参数 event

event.detail.event 事件源

event.detail.user user 信息

### **stateChange**

状态变化事件

同小程序的 publisher 组件的 bindstatechange

### **netStatusInfo**

网络状态

同小程序的 publisher 组件的 bindnetstatus

### **error**

渲染错误事件

同小程序的 publisher 组件的 binderror

## **交互**

**开关麦克风, 摄像头通过触发事件**

```
1.
   emitter.trigger(emitter.MEDIA_SWITCH_
   {
2.   videoOn: true,
3.   audioOn: true
4.   });
```

切换前后摄像头

```
1.
   emitter.trigger(emitter.SWITCH_CAMERA
```

示例

```
1. <publisher
2.   bind:netStatusInfo="onNetStatusInfo"
3.   bind:statusChange="onStatusChange"
4.   bind:publisherTap="onPublisherTap"
5.   beauty="{{beauty}}"
6.   whiteness="{{beauty}}"
7.
8. >
9. </publisher>
```

## handup 组件

用于实现举手功能

```
1. <handup></handup>
```

事件

**speakApplyAccept**

老师（主讲）同意举手触发

参数 `event.user` 当前用户

### **speakApplyReject**

老师（主讲）拒绝举手触发

参数 `event.user` 当前用户

### **注意**

1. 页面后台时候，推拉流会异常，此时需要触发 `eventEmitter.PAGE_HIDE` 事件，组件收到事件后会关闭音视频

1. `eventEmitter.trigger(eventEmitter.PAGE_HIDE);`

## **点名组件**

用于实现学生响应点名功能

### **示例**

1. `<rollCall></rollCall>`

### **事件**

无

## **答题器组件**

用于实现学生响应答题器功能

**示例（可自行控制答题器显示位置）**

1. `<answer style="position: fixed; bottom: 15px; left: 15px; right: 20px;" /><answer>`

## 事件

无

## 测验组件

用于实现学生响应测验功能（暂不支持学生提交答案后关闭页面，后续版本更新）

## 示例

1. `<quiz></quiz>`

## 课前问卷

用于实现收集学生课前问卷

## 示例

1. `<survey></survey>`

## 注意事项

需要将 `sdk/config.js` 中 `exports.QUESTIONNAIRE_PAGE = 'https://www.baijiayun.com/default/questionnaire/index?miniprogram=1'`；前缀域名替换为客户的自定义域名 如 `exports.QUESTIONNAIRE_PAGE = 'https://demo.at.baijiayun.com/default/questionnaire/index?miniprogram=1'`；并将 `https://demo.at.baijiayun.com` 加入到小程序业务域名

## 课后评价

下课时，根据管理后台配置，收集学生课后评价

### 示例

```
1. <evaluation></evaluation>
```

### 注意事项

需要将 sdk/config.js 中 exports.EVALUATION\_PAGE = 'https://www.baijiayun.com//m/evaluation/index?miniprogram=1&'; 前缀域名替换为客户的自定义域名 如 exports.EVALUATION\_PAGE = 'https://demo.at.baijiayun.com/m/evaluation/index?miniprogram=1&'; 并将https://demo.at.baijiayun.com 加入到小程序业务域名

## 直播第三方弹窗信令&自定义信令接收

```
1. // 等待服务器连接完成后监听第三方弹窗事件&自定义广
   播信令回调
2.   eventEmitter
3.   .on(
4.     eventEmitter.ROOM_SERVER_LOGIN_SUCCESS,
5.     function () {
6.       // 第三方弹窗打开url信令
7.       // 监听打开弹窗事件
8.       bgy.broadcast.messageTypes['custom_webpage']
9.       = function (fromId, data, fromCache) {
10.        if (data.url) {
11.          console.log('打开弹窗: ' + data.url);
12.        } else {
```

```
12.     console.log('关闭弹窗');
13.     }
14. }
15. // 获取打开弹窗缓存
16. bgy.broadcast.get('custom_page');
17.
18. // 自定义广播信令
19. bgy.api.init();
20. // 接收自定义广播信令
21.
22. bgy.api.onReceiveBroadcast('ht_custom_key1',
23.     function (fromId, data, cache) {
24.         console.log(data);
25.     })
26. // 获取自定义广播信令缓存
27.
28. bgy.api.getCachedBroadcast('ht_custom_key1');
29. }
```

## 注意事项

需要将 `sdk/config.js` 中 `exports.QUIZ_PATH =`  
`'https://www.baijiayun.com/m/quiz/student'`; 前缀域名替换  
为客户的自定义域名 如 `exports.QUIZ_PATH =`  
`'https://demo.at.baijiayun.com/m/quiz/student'`; 并将  
`https://demo.at.baijiayun.com` 加入到小程序业务域名

## 老师窗口全屏

因为微信限制动态生成视频窗口无法支持全屏操作，所以特殊处理为如果只有老师发言的情况下支持视频全屏，可以使用 `teacher-fullscreen` 分支的代码集成

## change log

---

- tag 1.0.0

初始版本 基础 sdk，组件主讲视频、白板、聊天、公告、加载、上麦用户视频列表显示

- tag 1.0.1

升级 BJY.init 方法，支持 noChat 配置。

当不需要聊天功能时，可以通过 noChat=true 不连接百家云的聊天服务器，少一条 websocket 连接

- tag 1.0.2

升级 BJY.init 方法，支持 hasDefaultTip 配置。支持设置不要直播间的默认提示

- tag 1.2.0

新增 publisher，handup 等组件和 API，支持发言，举手  
白板支持涂鸦

- tag 1.3.0

新增问答的组件

- tag 1.4.0

支持请求的个性域名

- tag 1.4.1

修复聊天服务器无法连接问题

- tag 1.5.0

添加点名组件

- tag 1.6.0

支持 webrtc 课程

- tag 1.7.0

支持分组和大小班切换

- tag 1.8.0

支持答题器&测验&多白板&APP 开课画笔优化

- tag 1.8.1

支持测验提交后关闭

- tag 1.9.0

部分代码重构优化

- tag 2.0.0

增加自定义广播信令接收

- tag 2.0.1

兼容微信 socket loading 时间过长问题

- tag 2.1.0

bv 底层支持连麦

## webview 嵌套百家云直播页面

---

1. 提供合法域名校验文件放置于百家云服务器
2. 创建教室时关闭微信授权



系统自动分配昵称:  启用  不启用

微信用户获取昵称:  启用  不启用

1. 复制百家云后台直播观看链接如

```
1. https://zhanminsi.at.baijiayun.com/m/room/prep
room_id=19111366632400&code=4e9qhu&url'
```

2. 链接后拼接上以下参数

```
1. &enterH5=true&dsp=1&disable_ppt_animate=
```

3. 小程序中赋值 webview src

```
1. <web-view
src="https://zhanminsi.at.baijiayun.com/m/room
room_id=19111366632400&code=4e9qhu&url'
</web-view>
```

## uniapp 小程序集成帮助

- 新建 wx 小程序目录 wxcomponents，将百家云 sdk 放入目录
- pages.json 声明需要使用的组件

```
1. {
```

```
2. "pages": [ //pages数组中第一项表示应用启动页，  
参考: https://uniapp.dcloud.io/collocation/pages  
3.   {  
4.     "path": "pages/index/index",  
5.     "style": {  
6.       "usingComponents": {  
7.         // b jy-live 组件为uni-app直播入口文件，  
初始化直播间以及事件监听建议都在该文件进行  
8.         "b jy-live":  
"/wxcomponents/sdk/b jyLive/b jyLive",  
9.         "loading":  
"/wxcomponents/sdk/component/loading/loading",  
10.        "teacher-player":  
"/wxcomponents/sdk/component/teacherPlayer/tea  
11.        },  
12.        "navigationBarTitleText": "uni-app"  
13.      }  
14.    }  
15.  ],  
16.  "globalStyle": {  
17.    "navigationBarTextStyle": "black",  
18.    "navigationBarTitleText": "uni-app",  
19.    "navigationBarBackgroundColor":  
"#F8F8F8",  
20.    "backgroundColor": "#F8F8F8"  
21.  },  
22.  "uniIdRouter": {}  
23. }
```

- page 页面使用组件

```
1. <template>  
2.   <view>  
3.     <b jy-live :options="options"></b jy-live>  
4.   </view>
```

```
5.     <loading class="loading-wrapper"
      @loadEnded="roomLoadEnded">
6.         </loading>
7.     </view>
8.     <view class="player" v-
      show="roomLoaded">
9.         <teacher-player></teacher-player>
10.    </view>
11. </view>
12. </template>
13.
14. <script>
15.   export default {
16.     data() {
17.       return {
18.         title: 'Hello',
19.         roomLoaded: false,
20.         // 初始化直播间需要的参数
21.         options: {
22.           env: "",
23.           apiOrigin:
24.             "https://zhanminsi.at.baijiayun.com/",
25.           sign:
26.             "53b28a4cf17c044761ed25b1e82a00c2",
27.           hasDefaultTip: true,
28.           class: {
29.             id: 23030188820642,
30.             name: '测试小程序',
31.             startTime: 1517969460000,
32.             endTime: 1833502260000
33.           },
34.           user: {
35.             number: '123',
36.             avatar: "",
37.             name: '123',
38.             type: 0
```

```
37.     }
38.   }
39. }
40. },
41.   mounted() {},
42.   methods: {
43.     roomLoadEnded: function() {
44.       this.roomLoaded = true
45.     }
46.   }
47. }
48. </script>
49.
50. <style>
51.   .player {
52.     width: 100vw;
53.     height: 56.25vw;
54.   }
55. </style>
```

- bjoyLive 组件为 uni-app 进直播的组件，进出直播间以及事件监听请在该组件进行，另外请不要在 wxcomponents 外做事件监听和调用 bjoy 函数

## uni-app 组件传参写法注意事项

- 小程序组件需要放在项目特殊文件夹 wxcomponents（或 mycomponents、swancomponents）。HBuilderX 建立的工程 wxcomponents 文件夹在 项目根目录下。vue-cli 建立的工程 wxcomponents 文件夹在 src 目录下。可以在 vue.config.js 中自定义其他目录
- 小程序组件的性能，不如 vue 组件。使用小程序组件，需要自己手动 setData，很难自动管理增量数据更新。而使用 vue 组件会自动 diff 更新增量数据。所以如无必要，建议使用 vue 组件而不是小程序组件。比如某些小程序 ui 组件，完全可以用更高性能的 uni ui 替代。

- 当需要在 vue 组件中使用小程序组件时，注意在 pages.json 的 globalStyle 中配置 usingComponents，而不是页面级配置。
- 注意数据和事件绑定的差异，组件使用时应按照 vue 的数据和事件绑定方式
- 属性绑定从 attr=""，改为 :attr="a"；从 title="复选框" 改为 :title="'复选框' + item"
- 事件绑定从 bpage 页面使用组件 nd:click="toggleActionSheet1" 改为 @click="toggleActionSheet1"，目前支付宝小程序不支持 vue 的事件绑定方式，具体参考：支付宝小程序组件事件监听示例
- 阻止事件冒泡 从 catch:tap="xx" 改为 @tap.native.stop="xx"
- wx:if 改为 v-if
- wx:for="" wx:key="" 改为 v-for="(item,index) in list"
- 详细的小程序转 uni-app 语法差异可参考文档 <https://ask.dcloud.net.cn/article/35786>。

[uni-app 小程序参考 demo](#)



下载为pdf格式