演播厅接口概览

一、快速入门

1、搭建好PHP环境

2、设计页面布局，分配好每块区域的作用，写好样式。

3、引入js

<script src="http://cdn.aodianyun.com/static/jquery/jquery-1.7.2.min.js" type="text/javascript"></script>

<script src="../shared/js/core-obj.js" type="text/javascript"></script>

<script src="../shared/js/core-ui.js" type="text/javascript"></script>

4、编写初始化演播室代码



5、若现有接口和模块不能满足新需求，再在此基础上，面向原始导播台开发，添加自己的模块。

二、演播室js接口说明

var studioApp = new StudioApp(); //一下是StudioApp类的接口说明

|  |  |  |  |
| --- | --- | --- | --- |
| 接口功能 | Function name | 参数 | 例子 |
| 演播室初始化 | Init | onStudioLoad：加载数据和初始化完成时回调的函数； | studioApp.Init({  onStudioLoad: function (info) {  //在这里初始化界面  //比如合成视频的预览框  studioApp.preVideoBox({  selector: "#idPgwVideo",  volume: 100, height: 220, width: '100%' });  }  }) |
|  |  |  |  |
| 初始化合成视频预览框 | preVideoBox | selector：jQuery选择器，可以是放视频的容器id或classname ；  volume:初始化时视频的音量大小(默认100)；  width:视频宽（默认360px）；  height: 视频高（默认240px）；  type:类型填写iframe时，视频以iframe的形式引入页面,认是直接填入页面的方式； | studioApp.preVideoBox({  selector: "#preVideoBox",  volume: 0  }); |
| 初始化合成视频播放音量条 | localVoiceBar | selector：放音量的容器id或classname；  direction:音量条的显示方式left2right为横条（默认是竖条）； | studioApp.localVoiceBar({  selector: ".voiceSliderPre",  direction: "left2right"  }); |
| 弹出合成视频的设置窗口 | showSetOutputDlg | 带参数时则代表固定该参数数据值,且不显示,户则无法改动  enableOutput：发布URL / 发布URL列表是否启用；  videoSize：视频大小；  videoBit：视频码率；  videoCBR：是否视频固定码率；  audioBit：音频码率； | <a onclick="studioApp.showSetOutputDlg{videoBit:’600k’}  )">设置输出流</a> |
| 弹出设置合成视频的节目输出延时的窗口 | showSetPgmDelayTimeDlg |  | <a onclick="studioApp.showSetPgmDelayTimeDlg()">设置节目输出延时</a> |
|  |  |  |  |
| 创建输入流视频 | smallVideoBox | selector：放视频的容器id或classname；  channel:填数字几就引入第几路视频流；  volume:初始化时视频的音量大小(默认100)；  width:视频宽（默认360px）；  height: 视频高（默认240px）；  type:类型填写iframe时，视频以iframe的形式引入页面,认是直接填入页面的方式； | studioApp.smallVideoBox({  selector: "#smallVideo\_2",  channel: 2,  volume: 100,  type: "iframe",  height: 360,  width: 640  }); |
| 初始化输入流视频的信号 | signalBox | selector：放信号的容器id或classname ；  channel：填数字几就引入第几路视频的信号 | studioApp.signalBox({ selector: ".signalBox1", channel: 0 }); |
| 初始化输入流视频的进度条 | progressBar | selector：放进度条的容器id或classname ；  channel：填数字几就引入第几路视频的进度条； | studioApp.progressBar({ selector: ".progressBar1", channel: 0 }); |
| 初始化输入流视频的播放音量 | localVoiceBar | selector：放音量的容器id或classname；  channel：填数字几就引入第几路视频的播放音量；  direction:音量条的显示方式left2right为横条（默认是竖条）； | studioApp.localVoiceBar({ selector: ".voiceSlider1", channel: 0 }); |
| 初始化输入流视频的发布音量 | pubVoiceBar | selector：音量的容器id或classname ；  channel：填数字几就引入第几路视频的发布音量；  direction:音量条的显示方式left2right为横条（默认是竖条）； | studioApp.pubVoiceBar({  selector: ".voiceSlider11",  channel: 0,  direction: "left2right"  }); |
| 弹出输入流视频的设置窗口 | showSetInputDlg | channel：填数字几就引入第几路视频的设置窗口； | <a onclick="studioApp.showSetInputDlg({ channel: 0 });">设置输入流0</a> |
| 弹出输入流视频抠像设置的窗口 | showSetKxDlg | channel：填数字几就引入第几路视频的抠像设置窗口； | <a onclick="studioApp.showSetKxDlg({channel:0})">视频1抠像设置</a> |
|  |  |  |  |
| 初始化导播台发布二维码框 | qrcodeImage | selector：放二维码的容器id或classname； | studioApp.qrcodeImage({ selector: '#qrcodeImg' }); |
| 初始化导播台的版本号、消耗的CPU等信息条 | osInfoBox | selector：放导播台信息的容器id或classname ； | studioApp.describeService({ selector: ".osInfoBox" }); |
| 重启导播台服务 | restartService |  | <a onclick="studioApp.restartService()">重启服务</a> |
| 重置导播台配置 | resetService |  | <a onclick="studioApp.resetService()">重置配置</a> |
| 修改导播台密码 | showSetLcpsPassword |  | <a onclick="studioApp.showSetLcpsPassword()">修改密码</a> |
| 初始化导播台配置 | expectLcps | inputCount：设置导播台源个数；  sceneWidth：设置导视频宽；  sceneHeight：设置导视频高；  success：设置成功后的回调函数；  failure:设置失败后的回调函数； | studioApp.expectLcps({  inputCount: 5,  sceneWidth: 1280,  sceneHeight: 720,  success: function () { }  }); |
| 返回导播台的信息 | getLcpsConfigInfo | 见附表一； | studioApp.getLcpsConfigInfo() |
|  |  |  |  |
| 加载场景并根据场景设置导播台的页面场景 | LoadSceneWithInitLcps | onGetSceneListSuccess：获取场景列表成功后调用的函数；  onInitSceneItem：将场景显示在页面上（生成的html中作为参数存储的属性名切勿随便更改。按右图命名）；  onSelectSceneItem：处理已被选中的场景；  onInitSceneCompleted：初始化场景完成后调用的函数； |  |
| 设置场景 | Lcps.SelectWebPage | para：需要修改的参数；  success：设置成功后的回调函数；  failure:设置失败后的回调函数  webPageInfo的参数说明：  index：第几个场景；  zIndex：场景显示的位置，数字越大就显示最前  enable：场景是否显示；  beforeVideo：场景是否在视频前显示；  url：场景的样式的地址；  SceneList[index].coverUrl：场景封面图的地址；  SceneList[index].mgrUrl：场景管理页面的地址；  Title：场景的描述；  Reload：是否重新加载；  "$$hashKey": "object:240"固定写法 | studioApp.Lcps.SelectWebPage({  para: webPageInfo,  failure: function (error) {  alert("设置场景失败");  }  });  var webPageInfo = {  "index": index,  "zIndex": $(self).attr("sortVal"),  "enable": self.checked,  "beforeVideo": parseInt($(self).attr("PosOfVideo")) == 2,  "url": $(self).attr("scene"),  "coverUrl": SceneList[index].coverUrl,  "mgrUrl": SceneList[index].mgrUrl,  "title": $(self).parent().attr("data-title"),  "reload": false,  "$$hashKey": "object:240"  }; |
|  |  |  |  |
| 设置演播厅信息 | studioApp.SetStudioConfig | config：可以存储任何信息在这里，但是数据大小不要超过4k；  success：设置成功后的回调函数；  failure:设置失败后的回调函数； | studioApp.SetStudioConfig({  config: data  }); |
| 设置输入流信息 | Lcps.ChangeInput | params：要改变得输入流参数；  success：设置成功后的回调函数；  failure:设置失败后的回调函数；  params请求参数说明：  channel ：通道，必选，数字几就修改第几路视频；  enableChromaKey：是否允许抠像；  playURL源地址，必选；  inputType 源类型，必选，值为"拉流", "推流", " 直接发布","白板"；  maxBufferTime 源最大缓冲时间，单位秒，可选；  wisId 当inputType为白板时使用，白板id，可 选； | studioApp.Lcps.ChangeInput({  params: {  playURL: 'kx.mp4',  inputType: "拉流",  channel: 0,  enableChromaKey: true,  maxBufferTime: 0  },  failure: function () {  alert("设置导播台源失败");  }  }); |
| 切换输入流场景 | Lcps.Switch | Params：要改变得输入流场景的参数；  success：设置成功后的回调函数；  failure:设置失败后的回调函数  Params请求参数说明：  数字几就修改第几路视频；  w0视频0宽，可选，值为number类型的值；  h0视频0高，可选，值为number类型的值；  x0视频0水平位置，可选，值为number类型的值；  y0视频0垂直位置，可选，值为number类型的值；  v0视频0音量，可选，值为number类型的值，范 围 0-100；  z0视频0叠加顺序，可选，值为number类型的值， 范围 0-(inputCount-1)； | //两个视频叠加  //效果最终在合成视频中显示  studioApp.Lcps.Switch({  paraArray: paraArray,  failure: function (error) {  alert("设置导播台视频失败");  }  });  Var paraArray=  [[{"w":640,"h":360,"x":0,"y":0,"v":100,"z":0},{"w":320,"h":180,"x":320,"y":180,"v":100,"z":1},{"w":320,"h":180,"x":640,"y":360,"v":0,"z":2},{"w":320,"h":180,"x":640,"y":360,"v":0,"z":3}](http://ip:3000/switch?paraArray=%5b%7B%22w%22:640,%22h%22:360,%22x%22:0,%22y%22:0,%22v%22:100,%22z%22:0%7D,%7B%22w%22:320,%22h%22:180,%22x%22:320,%22y%22:180,%22v%22:100,%22z%22:1%7D,%7B%22w%22:320,%22h%22:180,%22x%22:640,%22y%22:360,%22v%22:0,%22z%22:2%7D,%7B%22w%22:320,%22h%22:180,%22x%22:640,%22y%22:360,%22v%22:0,%22z%22:3%7D)] |

三、原始导播台配置参数说明

在帮助中心<http://www.aodianyun.com/aodianyun_doc/125> 开放的原始导播台api均已封装。未罗列出来的可以找到该API的方法名

然后这样调用 studioApp.Lcps.Switch

|  |
| --- |
| 附表一 |
| {  //string类型， code >=0 成功，否则失败  "code": "0",  // string类型，对应的提示消息字符串  "message": "success", // object类型，对应配置  "config": {  //array类型，导播台4个源地址  "sourceArray": [  {  // string类型，源地址  "playURL": "http://zb.v.qq.com:1863/?progid=100002100",  // string类型，源类型。可以是"拉流", "推流", "直接发布"  "inputType": "拉流"  },  {  "playURL": "http://zb.v.qq.com:1863/?progid=100104700",  "inputType": "拉流"  },  {  "playURL": "http://zb.v.qq.com:1863/?progid=100105500",  "inputType": "拉流"  },  {  "playURL": "http://zb.v.qq.com:1863/?progid=100104500",  "inputType": "拉流"  }  ],  // string类型，当前选择的通道。  "pgwCurrentChannel": 0, // array类型，导播台输出地址，目前只支持第0路输出  "pgwArray": [  { // bool类型，是否启用  "enable": true, //string类型，输出地址  "publishURL": "rtmp://127.0.0.1/3001/0\_pgw",  //string类型，输出视频大小。例如640x480, 640x?, ?x480，有问号的表示等比缩放  "videoSize": "?x480", //string类型，视频码率  "videoBitrate": "800k", //string类型，音频码率  "audioBitrate": "48k"  }, //此路禁用  {  "enable": false,  "publishURL": "rtmp://127.0.0.1/3001/1\_pgw",  "videoSize": "?x320",  "videoBitrate": "512k",  "audioBitrate": "48k"  }, //此路禁用  {  "enable": false,  "publishURL": "rtmp://127.0.0.1/3001/2\_pgw",  "videoSize": "?x240",  "videoBitrate": "256k",  "audioBitrate": "48k"  }, //此路作为pgw预览画面用  {  "enable": true,  "publishURL": "rtmp://127.0.0.1/3001/3\_pgw",  "videoSize": "?x240",  "videoBitrate": "256k",  "audioBitrate": "48k"  }  ], //number类型，场景宽  "sceneWidth": 640, //number类型，场景高  "sceneHeight": 480, //number类型，场景最大宽  "sceneWidthMax": 1280,  //number类型，场景最大高  "sceneHeightMax ": 1280,  //bool类型，是否启用logo  "logoEnable": true,  //string类型，logo地址  "logoPath": "adylogo.png", //number类型，logo宽  "logoWidth": 188, //number类型，logo高  "logoHeight": 56, //number类型，场景水平位置  " logoX ": 1020, //number类型，场景垂直位置  " logoY ": 44, //number类型，源个数  " inputCount ": 4,  //number类型，最大源个数  " inputCountMax": 8, //array类型，每个视频对应的叠加参数  "streamParaArray": [  { //number类型，视频水平位置  "x": 640, //number类型，视频垂直位置  "y": 480, //number类型，视频宽  "w": 640, //number类型，视频高  "h": 480, //number类型，音量  "v": 0, //number类型，叠加顺序  "z": 0  },  {  "x": 0,  "y": 0,  "w": 640,  "h": 480,  "v": 100,  "z": 1  },  {  "x": 640,  "y": 480,  "w": 640,  "h": 480,  "v": 0,  "z": 2  },  {  "x": 640,  "y": 480,  "w": 640,  "h": 480,  "v": 0,  "z": 3  }  ]  //object类型，保存的场景  "scene": {  //key值为对应的场景名字，value为场景参数值  "保存的场景名字": [  {  "w": 1280,  "h": 720,  "x": 1280,  "y": 0,  "v": 0,  "z": 0  },  {  "w": 1280,  "h": 720,  "x": 1280,  "y": 0,  "v": 0,  "z": 1  },  {  "w": 1280,  "h": 720,  "x": 1280,  "y": 0,  "v": 0,  "z": 2  },  {  "w": 1280,  "h": 720,  "x": 0,  "y": 0,  "v": 100,  "z": 3  }  ]  }  }  } |