

第3章 构建UI界面

第5节 媒体组件(1)

谢涛



内容提要

- audio 音频
- image 图片
- video 视频



1、audio音频

- audio音频组件需要有唯一的id
- 根据id使用`wx.createAudioContext('myaudio')`创建音频播放环境
- 注意：1.6.0版本开始，该组件不再维护。建议使用能力更强的 `wx.createInnerAudioContext` 接口
- audio音频属性如下表



属性	类型	说明
id	string	audio 组件的唯一标识符
src	string	要播放音频的资源地址
loop	boolean	是否循环播放
controls	boolean	是否显示默认控件
poster	string	默认控件上的音频封面的图片资源地址，如果 controls 属性值为 false 则设置 poster 无效
name	string	默认控件上的音频名字，如果 controls 属性值为 false 则设置 name 无效
author	string	默认控件上的作者名字，如果 controls 属性值为 false 则设置 author 无效
binderror	eventhandle	当发生错误时触发error事件，detail = {errMsg:MediaError.code}
bindplay	eventhandle	当开始/继续播放时触发play事件
bindpause	eventhandle	当暂停播放时触发 pause 事件
bindtimeupdate	eventhandle	当播放进度改变时触发 timeupdate 事件，detail = {currentTime, duration}
bindended	eventhandle	当播放到末尾时触发 ended 事件

audio音频组件返回错误码

返回错误码	描述
1	获取资源被用户禁止
2	网络错误
3	解码错误
4	不合适资源

示例代码

➤ index.wxml

```
<!--index.wxml-->
```

```
<audio poster="{{poster}}" name="{{name}}" author="{{author}}"  
src="{{src}}" id="myAudio" controls loop></audio>
```

```
<button type="primary" bindtap="audioPlay">播放</button>
```

```
<button type="primary" bindtap="audioPause">暂停</button>
```

```
<button type="primary" bindtap="audio14">设置当前播放时间为  
14秒</button>
```

```
<button type="primary" bindtap="audioStart">回到开头  
</button>
```




```
//index.js
const app = getApp()
```

```
Page({
  onReady: function (e) {
    // 使用 wx.createAudioContext 获取
    // audio 上下文 context
    this.audioCtx = wx.createAudioContext('myAudio')
  },
  data: {
    poster: '../bye.jpg',
    name: '再见',
    author: '张震岳',
    src: '../bye.mp3',
  },
})
```



```
audioPlay: function () {
  this.audioCtx.play()
},
audioPause: function () {
  this.audioCtx.pause()
},
audio14: function () {
  this.audioCtx.seek(14)
},
audioStart: function () {
  this.audioCtx.seek(0)
}
```

演示效果



2、image图片

- 支持JPG、PNG、SVG格式
- 2.3.0起支持云文件ID
- image图片组件有两类展现模式：
 - ✓ 缩放模式
 - 该模式包含4种方式
 - ✓ 裁剪模式
 - 该模式包含9种方式

image图片属性

属性	类型	默认值	说明
src	string		图片资源地址
mode	string	scaleToFill	图片裁剪、缩放的模式
lazy-load	boolean	false	图片懒加载，在即将进入一定范围（上下三屏）时才开始加载
show-menu-by-longpress	boolean	false	开启长按图片显示识别小程序码菜单
binderror	eventhandle		当错误发生时触发， event.detail={errMsg}
bindload	eventhandle		当图片载入完毕时触发， event.detail={height, width}

mode合法値

值	说明
scaleToFill	缩放模式，不保持纵横比缩放图片，使图片的宽高完全拉伸至填满 image 元素
aspectFit	缩放模式，保持纵横比缩放图片，使图片的长边能完全显示出来。也就是说，可以完整地将图片显示出来。
aspectFill	缩放模式，保持纵横比缩放图片，只保证图片的短边能完全显示出来。也就是说，图片通常只在水平或垂直方向是完整的，另一个方向将会发生截取。
widthFix	缩放模式，宽度不变，高度自动变化，保持原图宽高比不变
top	裁剪模式，不缩放图片，只显示图片的顶部区域
bottom	裁剪模式，不缩放图片，只显示图片的底部区域
center	裁剪模式，不缩放图片，只显示图片的中间区域
left	裁剪模式，不缩放图片，只显示图片的左边区域
right	裁剪模式，不缩放图片，只显示图片的右边区域
top left	裁剪模式，不缩放图片，只显示图片的左上边区域
top right	裁剪模式，不缩放图片，只显示图片的右上边区域
bottom left	裁剪模式，不缩放图片，只显示图片的左下边区域
bottom right	裁剪模式，不缩放图片，只显示图片的右下边区域

示例代码

```
<!--index.wxml-->
<view class="page">
  <view class="page__hd">
    <text class="page__title">image</text>
    <text class="page__desc">图片</text>
  </view>
  <view class="page__bd">
    <view class="section section_gap" wx:for="{{array}}" wx:for-item="item">
      <view class="section__title">{{item.text}}</view>
      <view class="section__ctn">
        <image style="width: 200px; height: 200px; background-color: #eeeeee;"
mode="{{item.mode}}" src="{{src}}"></image>
      </view>
    </view>
  </view>
</view>
```

```
//index.js
const app = getApp()

Page({
  data: {
    array: [{
      mode: 'scaleToFill',
      text: 'scaleToFill: 不保持纵横比缩放图片，使
      图片完全适应'
    }, {
      mode: 'aspectFit',
      text: 'aspectFit: 保持纵横比缩放图片，使图
      片的长边能完全显示出来'
    }, {
      mode: 'aspectFill',
      text: 'aspectFill: 保持纵横比缩放图片，只保
      证图片的短边能完全显示出来'
    }, {
      mode: 'top',
      text: 'top: 不缩放图片，只显示图片的顶部区
      域'
    }, {
      mode: 'bottom',
      text: 'bottom: 不缩放图片，只显示图片的底
      部区域'
    }, {
```

```
mode: 'center',
      text: 'center: 不缩放图片，只显示图片的中间
      区域'
    }, {
      mode: 'left',
      text: 'left: 不缩放图片，只显示图片的左边区
      域'
    }, {
      mode: 'right',
      text: 'right: 不缩放图片，只显示图片的右边
      边区域'
    }, {
      mode: 'top right',
      text: 'top right: 不缩放图片，只显示图片的右
      上边区域'
    }, {
      mode: 'bottom right',
      text: 'bottom right: 不缩放图片，只显示图片
      的右下边区域'
    }],
    src: 'cat.jpg'
  },
  imageError: function (e) {
    console.log('image3发生error事件，携带值为',
    e.detail.errMsg)
  }
})
```


演示效果



3、video视频

- 视频组件
- 根据id使用

`wx.createVideoContext('myvideo')`创建视频播放环境

- video视频属性如下表

属性	类型	默认值	说明
src	string		要播放视频的资源地址，支持云文件ID (2.3.0)
controls	boolean	true	是否显示默认播放控件（播放/暂停按钮、播放进度、时间）
danmu-list	Array.<object>		弹幕列表
danmu-btn	boolean	false	是否显示弹幕按钮，只在初始化时有效，不能动态变更
enable-danmu	boolean	false	是否展示弹幕，只在初始化时有效，不能动态变更
autoplay	boolean	false	是否自动播放
object-fit	string	contain	当视频大小与 video 容器大小不一致时，视频的表现形式 contain : 包含。 fill : 填充。 cover : 覆盖
bindplay	eventhandle		当开始/继续播放时触发play事件
bindpause	eventhandle		当暂停播放时触发 pause 事件
bindended	eventhandle		当播放到末尾时触发 ended 事件
bindtimeup date	eventhandle		播放进度变化时触发，event.detail = {currentTime, duration} 。触发频率 250ms 一次

```
<!--index.wxml-->
<view class="section tc">
<video src="{{src}}" controls >/video>
<view class="btn-area">
<button bindtap="bindButtonTap">获取视频</button>
</view>
</view>
<view class="section tc">
<video id="myVideo" src="http://example.com/test.mp4"
danmu-list="{{danmuList}}" enable-danmu danmu-btn
controls>/video>
<view class="btn-area">
<button bindtap="bindButtonTap">获取视频</button>
<input bindblur="bindInputBlur"/>
<button bindtap="bindSendDanmu">发送弹幕</button>
</view>
</view>
```

```
//index.js
const app = getApp()

function getRandomColor() {
  let rgb = []
  for (let i = 0; i < 3; ++i) {
    let color = Math.floor(Math.random() *
    256).toString(16)
    color = color.length == 1 ? '0' + color :
    color
    rgb.push(color)
  }
  return '#' + rgb.join('')
}
```

```
Page({
  onReady: function (res) {
    this.videoContext =
    wx.createVideoContext('myVideo')
  },
  inputValue: '',
  data: {
    src: '',
    danmuList: [
      {
        text: '第 1s 出现的弹幕',
        color: '#ff0000',
        time: 1
```

```
},
    {
      text: '第 3s 出现的弹幕',
      color: '#ff00ff',
      time: 3
    }
  ],
  bindInputBlur: function (e) {
    this.inputValue = e.detail.value
  },
  bindButtonTap: function () {
    var that = this
    wx.chooseVideo({
      sourceType: ['album', 'camera'],
      maxDuration: 60,
      camera: ['front', 'back'],
      success: function (res) {
        that.setData({
          src: res.tempFilePath
        })
      }
    })
  },
  bindSendDanmu: function () {
    this.videoContext.sendDanmu({
      text: this.inputValue,
      color: getRandomColor()
    })
  }
})
```

运行效果

