# 笔记05

## 回顾

### audio音频

### 表单组件

Button、checkbox/checkbox-group、radio/radio-group/ switch/ picker/input / from

### 导航

Navigator 页面链接

Api

路由传参/接参

## 模块化操作

可以将一些公共的代码抽离成为一个单独的 js 文件，作为一个模块。

模块化：针对的js，业务逻辑。

作用： 提高开发效率，方便后期维护

ES6/commonjs规范

### 应用层面

ES6： 前端代码，vue / react

Commonjs：后端，node

### 语法层面

#### ES6

暴露：

1. Export { 属性，属性.... }
2. Export default 变量

导入：

1）Imort { 变量,变量 } from “文件”

2）Import 变量 from “文件”

#### Commonjs

暴露：

Module.exports = {}

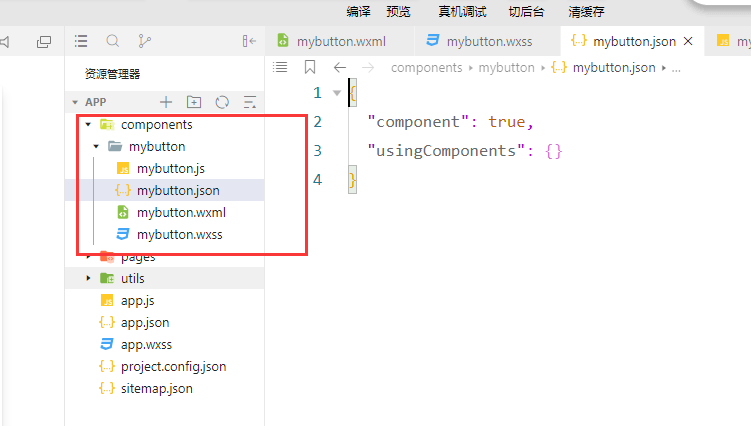
导入：

Var 变量 = require(‘路径’)

## 组件化

创建自定义组件，接受一个 Object 类型的参数。

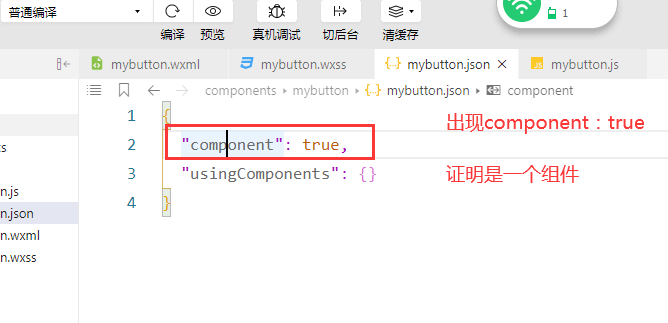
### 自定义组件的创建



Wxml： 与页面的wxml一直，都是页面结构，视图层

Wxss：与页面的wxss一直，都是页面样式

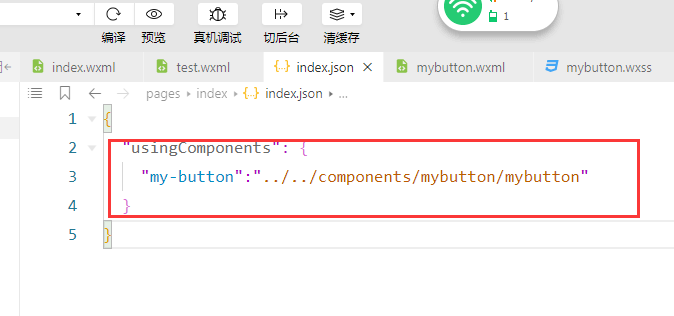
Json：就是页面的配置文件，



Js： 功能是js逻辑文件，但是与page的js有所不同

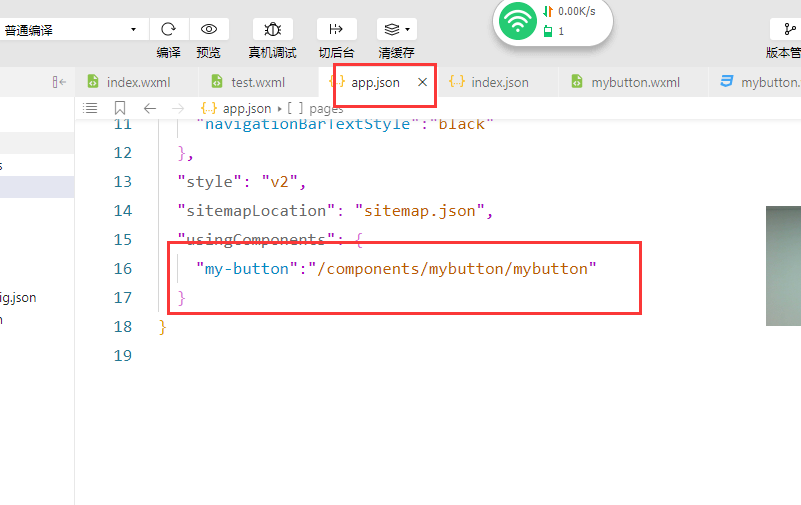
### 如何使用自定义组件

#### 局部导入

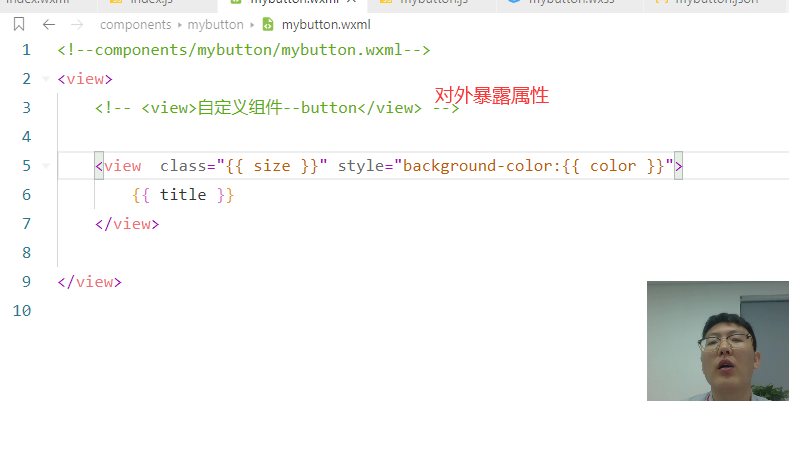


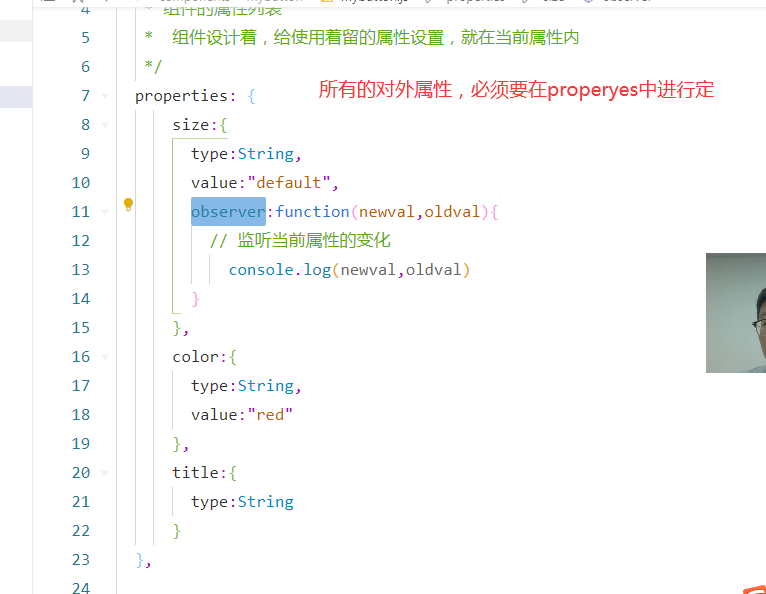


#### 全局导入



### 如何编辑封装自定义组件内容

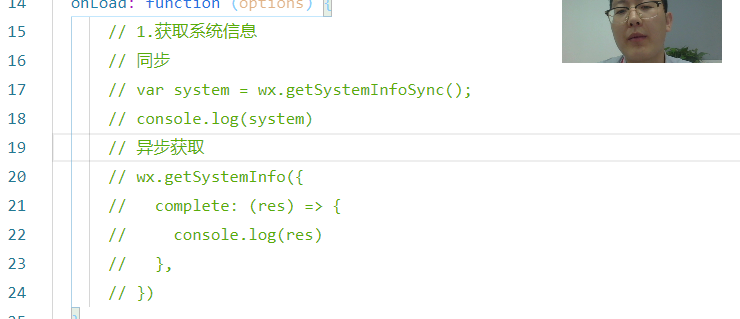






## Api

### 基础api



### 交互API

 // 2.交互

      // 显示消息提示框

      // wx.showToast({

      //   title: '数据加载中',

      //   icon:"loading",

      //   duration:5000,

      //   image:"../../icon/myFull.png",  //图片

      //   complete:(res)=>{

      //     console.log(res)

      //   }

      // })

      // 网络请求，下载，或上传

      // 隐藏消息提示框

      // wx.hideToast({

      //   complete: (res) => {},

      // })

      //

      // wx.showLoading({

      //   title: '数据加载中',

      // })

      // 当请求成功之后，隐藏

      // setTimeout(()=>{

      //   wx.hideLoading({

      //     complete: (res) => {},

      //   })

      // },3000)

  del(e){

    let id = e.target.dataset.id;

    // 提示用户是否删除

    wx.showModal({

       title:"删除提示",

       content:`您是否要删除id为${id}的商品？`,

       cancelText:"back",

       cancelColor:"red",

       success:(res)=>{

          console.log(res)

          // 取消：  cancel（取消）  true    confirm（确定）  false

          if(res.confirm){

              //确定删除了

              console.log('删除')

          }else{

            //不删除

              console.log('不删除')

          }

       }

    })

  }

 changePhoto(){

    let \_this = this;

    // 选择照片  【手机相册，拍照】

     wx.showActionSheet({

        itemList: ['手机相册', '拍照'],

        success (res) {

          // console.log(res.tapIndex)

          if(res.tapIndex ==  1){

            \_this.fn('camera')

          }else if(res.tapIndex ==  0){

            \_this.fn('album')

          }

        },

        fail (res) {

          console.log(res.errMsg)

        }

      })

  }



### 导航栏

 onLoad: function (options) {

      let {id,name} = options; // 获取参数

    // 动态设置导航标题

      wx.setNavigationBarTitle({

        title:name

      })

      // 在顶部window，设置loading样式

      wx.showNavigationBarLoading({

        complete: (res) => {},

      })

      setTimeout(()=>{

        wx.hideNavigationBarLoading({

          complete: (res) => {},

        })

      },3000)

      // 隐藏 返回首页的按钮（图标）

      wx.hideHomeButton({

        complete: (res) => {},

      })

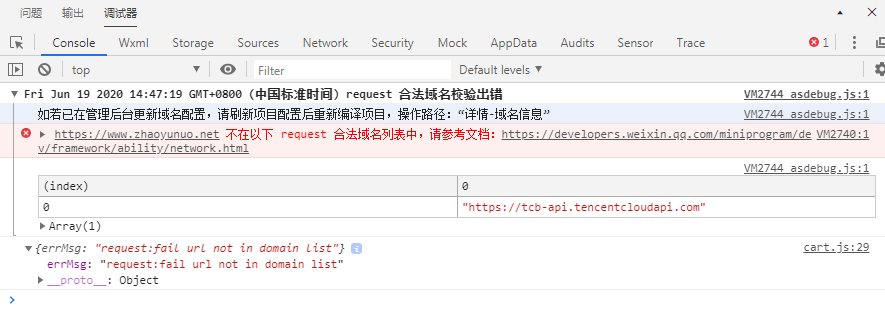
  },

### 网络请求

注意：

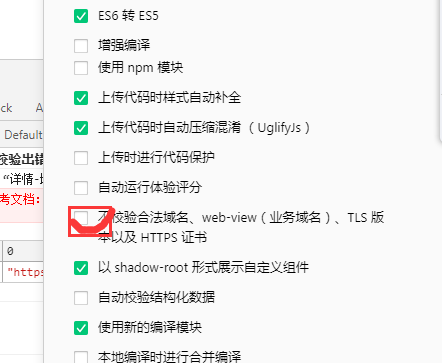
1. 小程序中所有的请求的地址必须是https开头的
2. 请求的地址，必须要在小程序的后台进行配置





解决方法：

如果没有进行域名的配置，那么可以在微信开发者工具中进行处理：但是上线发布，必须要进行配置。



 getData(){

       // 网络请求

       wx.request({

         url: 'http://www.zhaoyunuo.net/data.json',

         data:{},

         header:{

           "content-type":"application/json"

         },

         method:"get",

         success:(res)=>{

            console.log(res)

         },

         fail:(err)=>{

            console.log(err)

         }

       })

  }

封装请求api：

// 用来封装请求api的

const urlApi = (url,data={},method="get") => {

  return new Promise((resolve, reject) => {

    wx.request({

      url,

      data,

      header: {

        "content-type": "application/json"

      },

      method,

      success: (res) => {

        resolve(res,'请求成功')

      },

      fail: (err) => {

        reject(err,'请求失败')

      }

    })

  })

}

export  default   urlApi;

### 数据缓存

将数据存储在本地缓存中指定的 key 中。会覆盖掉原来该 key 对应的内容。除非用户主动删除或因存储空间原因被系统清理，否则数据都一直可用。单个 key 允许存储的最大数据长度为 1MB，所有数据存储上限为 10MB。

#### 存储



#### (2) 获取

 getstroage(){

      // 异步获取

      // wx.getStorage({

      //   key: 'classid',

      //   success(res){

      //     console.log(res)

      //   }

      // })

      // 同步  try  catch

      try{

        var info =  wx.getStorageSync('carts')

        // console.log(info)

        info.map((item,index)=>{

          console.log(item)

        })

      }catch(err){

        // 代码走到这里，加逻辑处理

          console.log(err.message)

      }

  },

#### 删除

 removestroage(){

    wx.removeStorage({

      key: 'cart',

      success(res){

        console.log(res)

      }

    })

  }

## 开发接口

### 用户信息

#### api的方式

· wx.authorize({scope: "scope.userInfo"})，不会弹出授权窗口，请使用 [<button open-type="getUserInfo"/>](https://developers.weixin.qq.com/miniprogram/dev/component/button.html)

思路：

如果第一次进入小程序，需要获取用户信息，授权必须使用button形式进行，（第一次获取用户信息，必须使用button），授权之后，就可以使用api的形式获取用户信息了！！！

#### 按钮button

   <view wx:if="{{ !onOff }}">

          <image src="{{ photo }}" class="photo"></image>

          <button  type="warn" size="mini" bindgetuserinfo="getuserinfo" open-type="getUserInfo">未登录</button>

      </view>

### 小程序登录

Wx.login()

商品分类、商品信息、商品详细、购物车、订单、用户信息、卡券、代金券......

都是后台提供接口。（php python java node）

前台：

电商小程序，

首页： 商品分类、商品信息

详情页： 商品详细

以上不是私密的；

购物车： 用户登录了，才能够获取当前购物车信息/订单信息

登录状态 （存活时间，60）

每次请求： data 登录状态

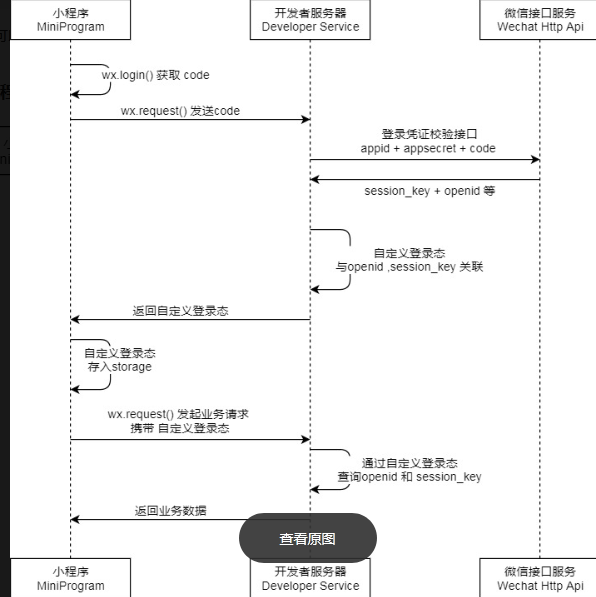
订单列表：

电话号码登录，获取电话号码 （常用的登录操作）（个人主体做不了）

输入手机号，获取验证码登录 （表单形式）

输入用户名，密码进行登录 (表单)

**Wx.login() 小程序自带的； 原理和流程图**



Mpvue

<https://www.showapi.com/apiGateway/view?apiCode=1700> 项目的接口地址