

互联网应用开发技术

*Web Application Development*

---

# 第15课 微信小程序 – BOOKSTORE

Episode Fifteen  
**WeChat App Sample**

陈昊鹏  
[chen-hp@sjtu.edu.cn](mailto:chen-hp@sjtu.edu.cn)



# 小程序的启动



REliable, INtelligent & Scalable Systems

- App.json中配置的第一个页面是首页
- 小程序启动之后，回调onLaunch函数

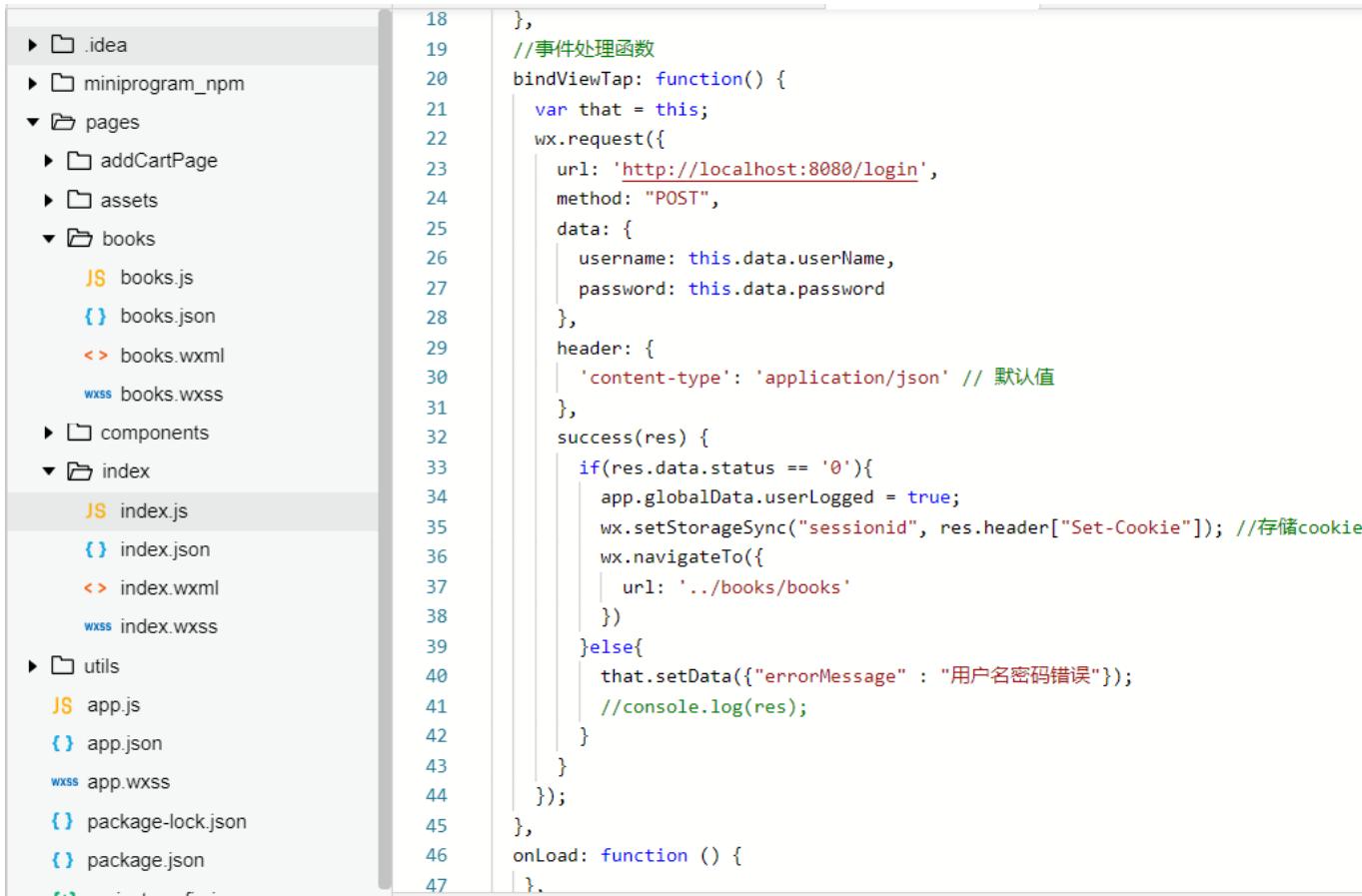
```
{  
  "pages": [  
    "pages/index/index",  
    "pages/logs/logs"  
  ]  
}
```

```
App({  
  onLaunch: function () {  
    // 小程序启动之后 触发  
  }  
})
```

- 现根据json文件生成页面
- 装载WXML结构和WXSS样式
- 装载JS

```
Page({  
    data: { // 参与页面渲染的数据  
        logs: []  
    },  
    onLoad: function () {  
        // 页面渲染后 执行  
    }  
})
```

- wx.request
- wx.navigateTo



The image shows a code editor interface with a sidebar on the left displaying a file tree. The tree includes .idea, miniprogram\_npm, pages (with addCartPage, assets, books), components, index (with index.js, index.json, index.wxml, index.wxss), and utils (with app.js, app.json, app.wxss, package-lock.json, package.json). The index.js file is currently selected.

The main pane displays the following JavaScript code:

```
18 },
19 //事件处理函数
20 bindViewTap: function() {
21     var that = this;
22     wx.request({
23         url: 'http://localhost:8080/login',
24         method: "POST",
25         data: {
26             username: this.data.userName,
27             password: this.data.password
28         },
29         header: {
30             'content-type': 'application/json' // 默认值
31         },
32         success(res) {
33             if(res.data.status == '0'){
34                 app.globalData.userLogged = true;
35                 wx.setStorageSync("sessionid", res.header["Set-Cookie"]); //存储cookie
36                 wx.navigateTo({
37                     url: '../books/books'
38                 })
39             }else{
40                 that.setData({"errorMessage" : "用户名密码错误"});
41                 //console.log(res);
42             }
43         }
44     });
45 },
46 onLoad: function () {
47 }
```

# Book Store Sample

REin

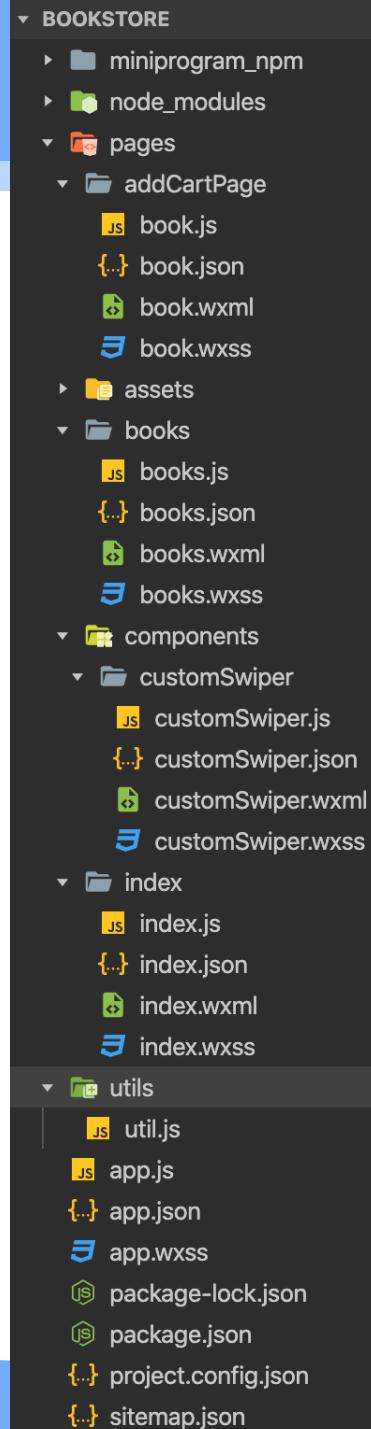
REliable, INtelligent & Scalable Systems

- 将网页端改为微信小程序
- 包含登录，浏览图书，加购物车等功能



# Book Store Sample

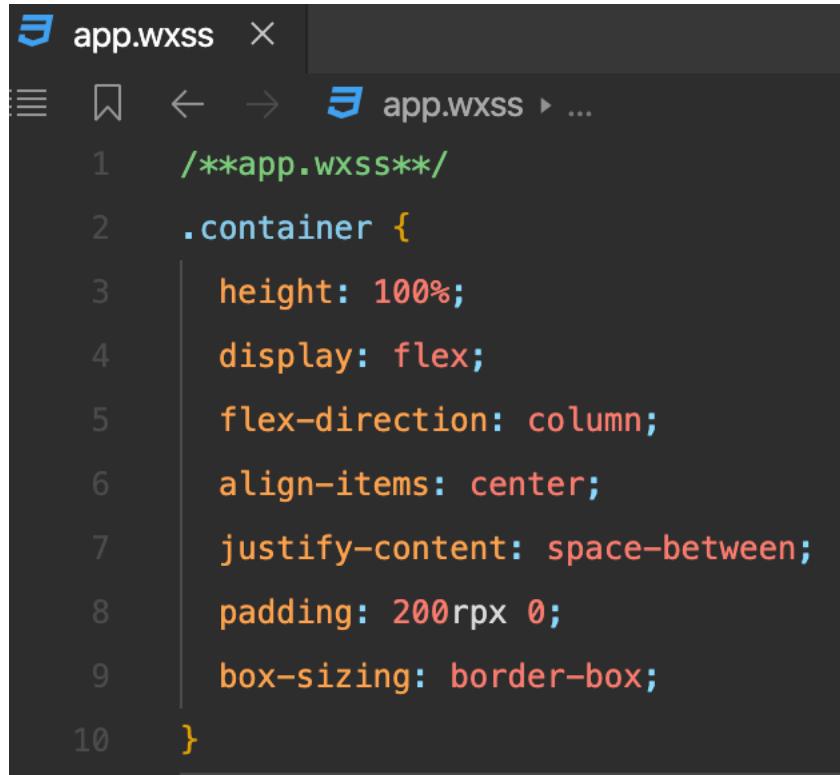
- 代码结构
- 每个页面包含四个文件
  - .json: 页面配置
  - .js: 页面逻辑
  - .wxss: 页面样式
  - .wxml: 页面界面
- app.json: 全局配置
- app.js: 程序入口



- app.json

```
← →  {..} app.json ▶ ...
{
  "pages": [
    "pages/index/index",
    "pages/books/books",
    "pages/addCartPage/book"
  ],
  "window": {
    "backgroundTextStyle": "light",
    "navigationBarBackgroundColor": "#fff",
    "navigationBarTitleText": "Book Store",
    "navigationBarTextStyle": "black"
  },
  "sitemapLocation": "sitemap.json",
  "usingComponents": {
    "van-image": "@vant/weapp/image",
    "van-search": "@vant/weapp/search",
    "van-goods-action": "@vant/weapp/goods-action",
    "van-goods-action-icon": "@vant/weapp/goods-action-icon",
    "van-goods-action-button": "@vant/weapp/goods-action-button",
    "van-overlay": "@vant/weapp/overlay",
    "van-tabbar": "@vant/weapp/tabbar",
    "van-tabbar-item": "@vant/weapp/tabbar-item"
  }
}
```

- app.wxss



```
app.wxss  X
1  /**app.wxss*/
2  .container {
3      height: 100%;
4      display: flex;
5      flex-direction: column;
6      align-items: center;
7      justify-content: space-between;
8      padding: 200rpx 0;
9      box-sizing: border-box;
10 }
```

# Book Store Sample



REliable, INtelligent & Scalable Systems

- app.wxss

The screenshot shows a code editor with two tabs: 'app.js' and 'app.wxss'. The 'app.js' tab is currently active, displaying the following code:

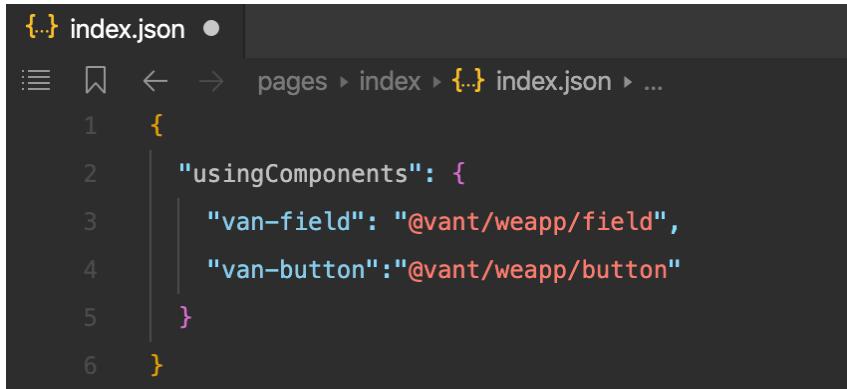
```
//app.js
App({
  onLaunch: function () {
    // 展示本地存储能力
    var logs = wx.getStorageSync('logs') || []
    logs.unshift(Date.now())
    wx.setStorageSync('logs', logs)

    // 登录
    wx.login({
      success: res => {
        // 发送 res.code 到后台换取 openId, sessionKey, unionId
      }
    })
  }
})
```

The 'app.wxss' tab is visible in the background. The code for 'app.wxss' is as follows:

```
15   // 获取用户信息
16   wx.getSetting({
17     success: res => {
18       if (res.authSetting['scope.userInfo']) {
19         // 已经授权，可以直接调用 getUserInfo 获取头像昵称，不会弹框
20         wx.getUserInfo({
21           success: res => {
22             // 可以将 res 发送给后台解码出 unionId
23             this.globalData.userInfo = res.userInfo
24
25             // 由于 getUserInfo 是网络请求，可能会在 Page.onLoad 之后才返回
26             // 所以此处加入 callback 以防止这种情况
27             if (this.userInfoReadyCallback) {
28               this.userInfoReadyCallback(res)
29             }
30           }
31         })
32       }
33     }
34   },
35   globalData: {
36     userLogged: false,
37   }
})
```

- index.json



```
{..} index.json ●
pages ▶ index ▶ {..} index.json ▶ ...
1  {
2    "usingComponents": {
3      "van-field": "@vant/weapp/field",
4      "van-button": "@vant/weapp/button"
5    }
6 }
```

- Index.wxss



```
index.wxss ×
pages ▶ index ▶ index.wxss ▶ ...
1  /**index.wxss**/
2  .userinfo {
3    display: flex;
4    flex-direction: column;
5    align-items: center;
6  }
7
8  .userinfo-nickname {
9    color: #aaa;
10 }
```

- index.wxml

```
index.wxml ×
pages › index › index.wxml

1  <!--index.wxml-->
2  <view class="container">
3    <view class="userinfo">
4      <van-cell-group>
5        <van-field bindblur="bindNameInput" label="用户名" placeholder="请输入用户名"/>
6        <van-field bindblur="bindPasswordInput" password="true" label="密码" placeholder="请输入密码"
error-message="{{errorMessage}}" border= true />
7      </van-cell-group>
8    </view>
9    <view style="width:15rem; height:6rem; margin: 5rem 0 " >
10      <van-button bindtap="bindViewTap" type="primary" block>登录</van-button>
11    </view>
12  </view>
```

# Book Store - Index



REliable, INtelligent & Scalable Systems

- index.js

```
index.js  x  index.js  x

1 //index.js
2 //获取应用实例
3 const app = getApp()
4
5 Page({
6   data: {
7     userName: '',
8     password: '',
9     errorMessage: '',
10    },
11   //用户名输入
12   bindNameInput: function(event){
13     this.setData({userName : event.detail.value})
14   },
15   //密码输入
16   bindPasswordInput: function(event){
17     this.setData({ password: event.detail.value })
18   },
19   //事件处理函数
20   bindViewTap: function() {
21     var that = this;
22     wx.request({
23       url: 'http://localhost:8080/login',
24       method: "POST",
25       data: {
26         username: this.data.userName,
27         password: this.data.password
28       },
29     }
30     header: {
31       'content-type': 'application/json' // 默认值
32     },
33     success(res) {
34       if(res.data.status == '0'){
35         app.globalData.userLogged = true;
36         wx.setStorageSync("sessionid", res.header["Set-Cookie"]); //存储cookie
37         wx.navigateTo({
38           url: '../books/books'
39         })
40       }else{
41         that.setData({"errorMessage" : "用户名密码错误"});
42         //console.log(res);
43       }
44     });
45   },
46   onLoad: function () {
47   },
48 }


```

# Book Store – Books

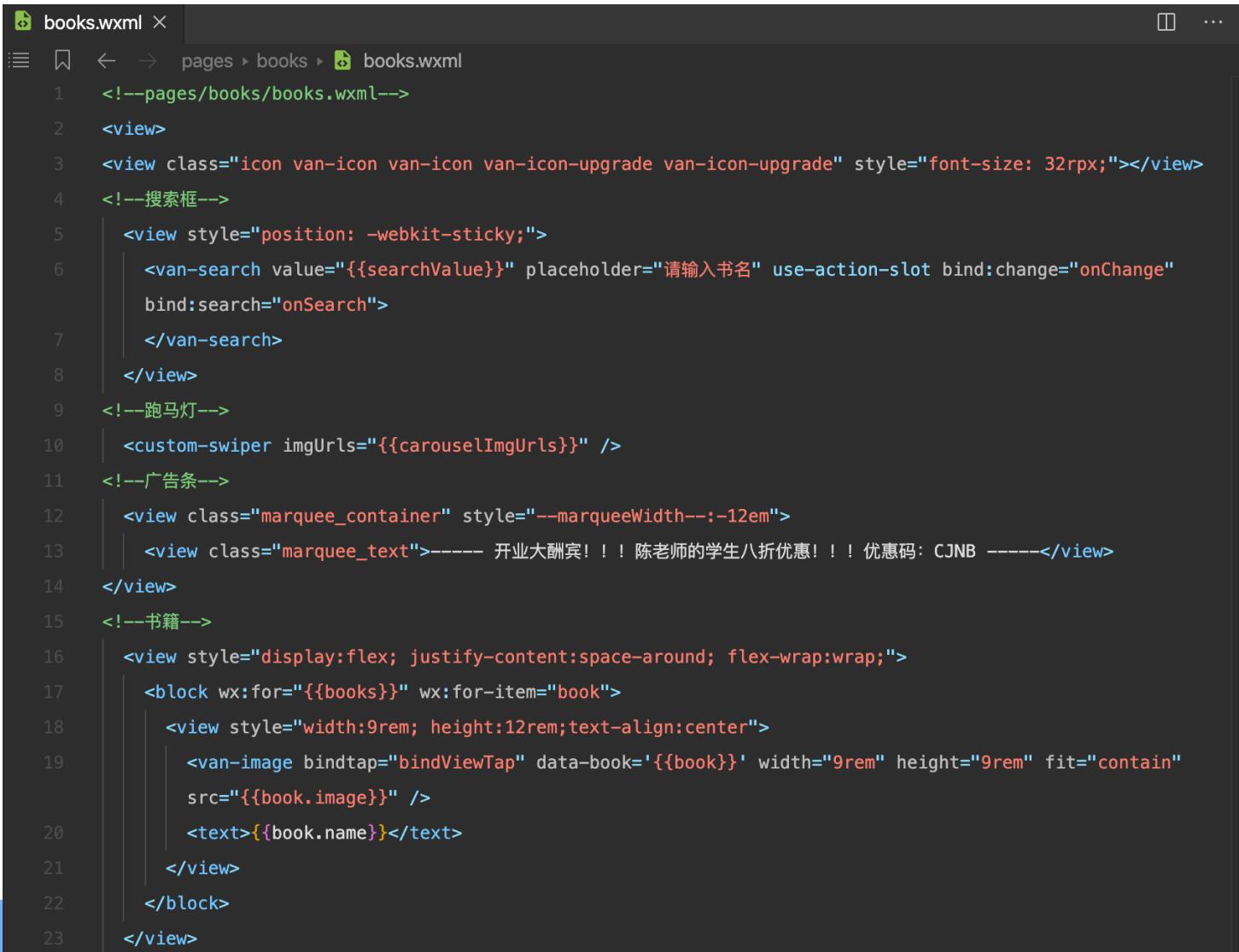
- Books.json

```
{..} books.json ●  
pages > books > {..} books.json > ...  
1 {  
2   "usingComponents": {  
3     "custom-swiper": "../components/customSwiper/customSwiper"  
4   }  
5 }
```

- Books.wxss

```
books.wxss X  
pages > books > books.wxss > ...  
1 /* pages/books/books.wxss */  
2 @keyframes around {  
3   from {  
4     margin-left: 100%;  
5   }  
6   to {  
7     /* var接受传入的变量 */  
8     margin-left: var(--marqueeWidth--);  
9   }  
10 }  
11  
12 .marquee_container{  
13   background-color: #fe4655;  
14   height: 50rpx;  
15   line-height: 44rpx;  
16   position: relative;  
17   width: 100%;  
18   margin-top: 0rpx;  
19 }  
20 .marquee_container:hover{  
21   /* 不起作用 */  
22   animation-play-state: paused;  
23 }  
24 .marquee_text{  
25   color: #fff;  
26   font-size: 28rpx;  
27   display: inline-block;  
28   white-space: nowrap;  
29   animation-name: around;  
30   animation-duration: 10s; /*过渡时间*/  
31   animation-iteration-count: infinite;  
32   animation-timing-function: linear;  
33 }
```

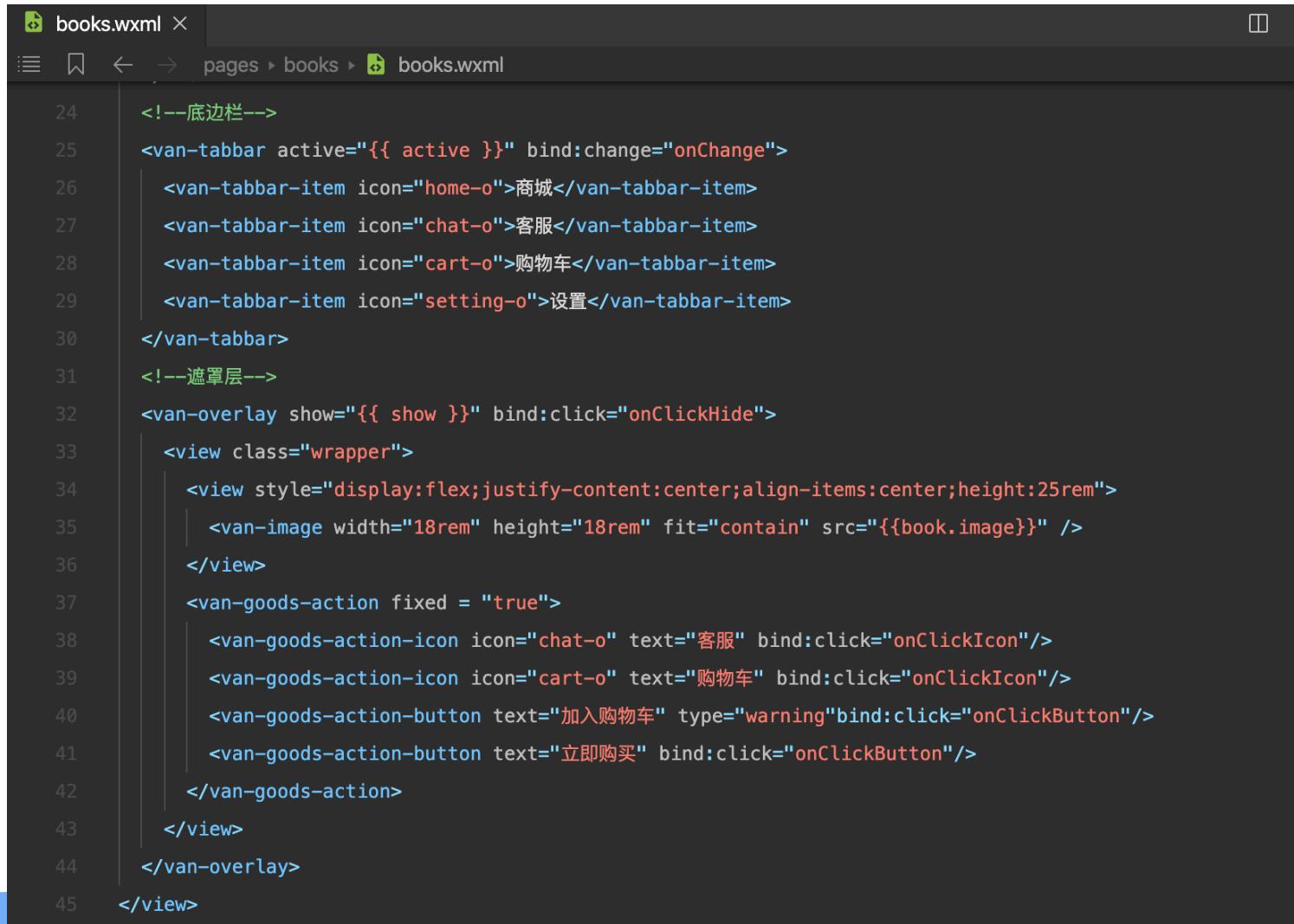
- Books.wxml



The screenshot shows a code editor window with the file "books.wxml" open. The file path is visible at the top: pages > books > books.wxml. The code itself is a WXML template for a book store page. It includes a search bar, a marquee text banner, and a grid of book cards. The code uses various components like `<view>`, `<van-search>`, `<custom-swiper>`, and `<van-image>`. It also includes some CSS-like styling with `style="..."` attributes.

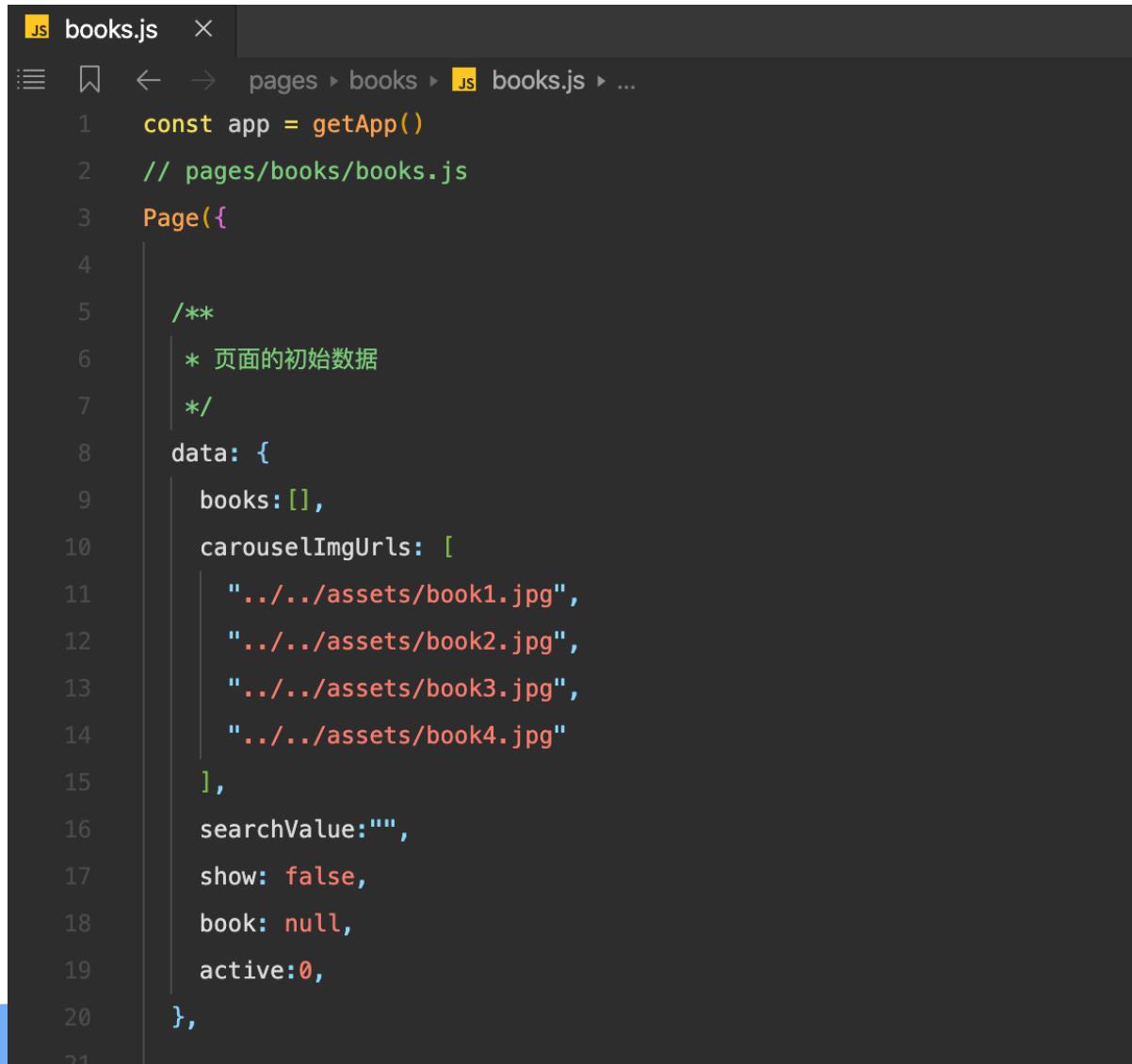
```
<!--pages/books/books.wxml-->
<view>
  <view class="icon van-icon van-icon-upgrade van-icon-upgrade" style="font-size: 32rpx;"></view>
  <!--搜索框-->
  <view style="position: -webkit-sticky;">
    <van-search value="{{searchValue}}" placeholder="请输入书名" use-action-slot bind:change="onChange" bind:search="onSearch">
    </van-search>
  </view>
  <!--跑马灯-->
  <custom-swiper imgUrls="{{carouselImgUrls}}"/>
  <!--广告条-->
  <view class="marquee_container" style="--marqueeWidth--:-12em">
    <view class="marquee_text">----- 开业大酬宾！！！陈老师的学生八折优惠！！！优惠码: CJNB -----</view>
  </view>
  <!--书籍-->
  <view style="display:flex; justify-content:space-around; flex-wrap:wrap;">
    <block wx:for="{{books}}" wx:for-item="book">
      <view style="width:9rem; height:12rem;text-align:center">
        <van-image bindtap="bindViewTap" data-book="{{book}}" width="9rem" height="9rem" fit="contain" src="{{book.image}}"/>
        <text>{{book.name}}</text>
      </view>
    </block>
  </view>
</view>
```

- Books.wxml

A screenshot of a code editor window titled "books.wxml". The window shows the file structure under "pages > books > books.wxml". The code itself is a WXML template for a bottom tab bar and a modal overlay. It includes components like , , , and . The code uses variables like {{ active }}, {{ book.image }}, and event handlers like "onChange", "onClickHide", and "onClickIcon".

```
<!--底边栏-->
<van-tabbar active="{{ active }}" bind:change="onChange">
  <van-tabbar-item icon="home-o">商城</van-tabbar-item>
  <van-tabbar-item icon="chat-o">客服</van-tabbar-item>
  <van-tabbar-item icon="cart-o">购物车</van-tabbar-item>
  <van-tabbar-item icon="setting-o">设置</van-tabbar-item>
</van-tabbar>
<!--遮罩层-->
<van-overlay show="{{ show }}" bind:click="onClickHide">
  <view class="wrapper">
    <view style="display:flex;justify-content:center;align-items:center;height:25rem">
      <van-image width="18rem" height="18rem" fit="contain" src="{{book.image}}"/>
    </view>
    <van-goods-action fixed = "true">
      <van-goods-action-icon icon="chat-o" text="客服" bind:click="onClickIcon"/>
      <van-goods-action-icon icon="cart-o" text="购物车" bind:click="onClickIcon"/>
      <van-goods-action-button text="加入购物车" type="warning" bind:click="onClickButton"/>
      <van-goods-action-button text="立即购买" bind:click="onClickButton"/>
    </van-goods-action>
  </view>
</van-overlay>
</view>
```

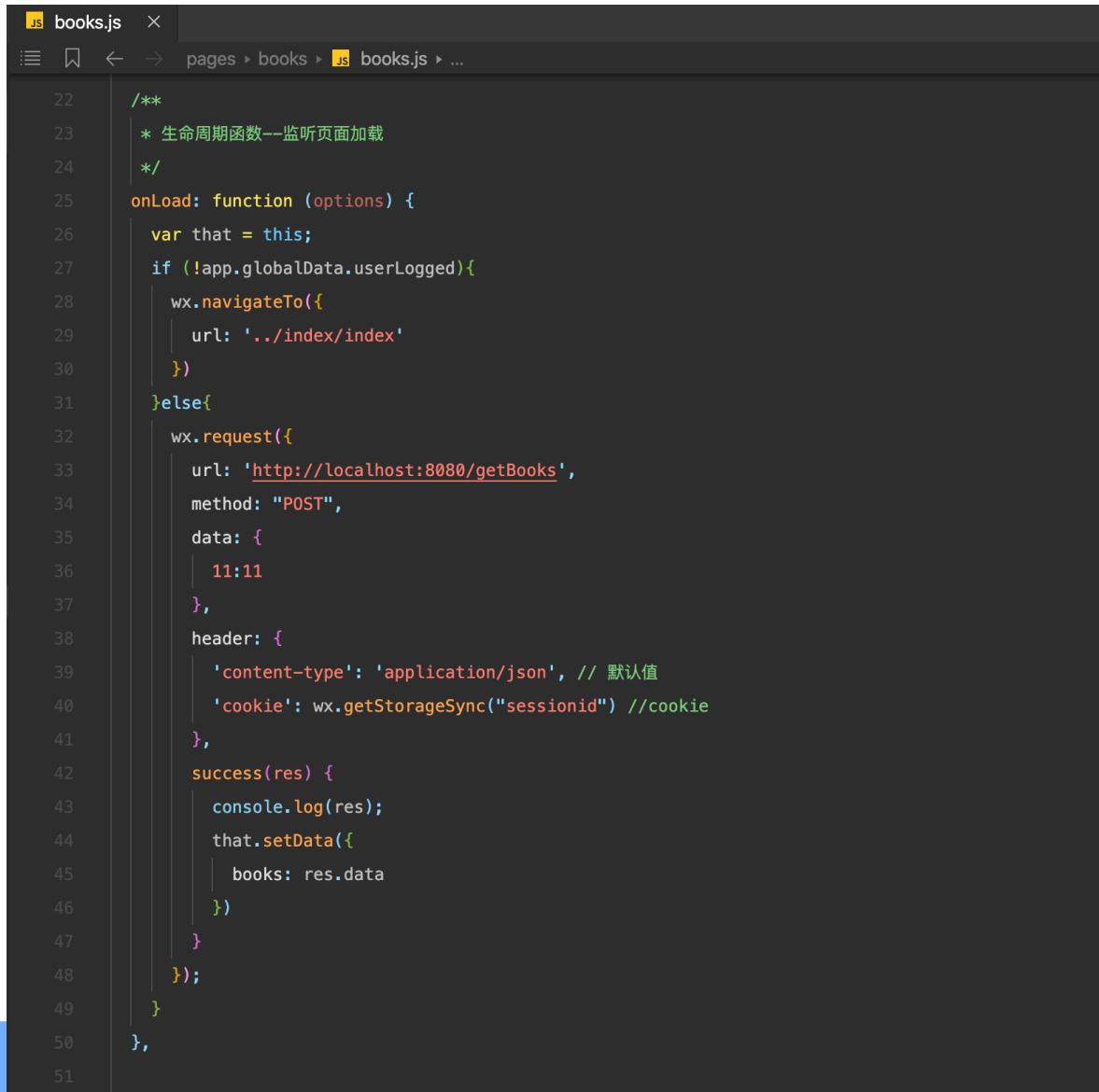
- Book.js



A screenshot of a code editor showing a file named "books.js". The file is a JavaScript file located in a directory structure: pages > books > books.js. The code defines a component named "Page" which uses the "app" object from "getApp()". It includes initial data for books and carousel image URLs, and handles search and active book state.

```
JS books.js  ×
pages › books › JS books.js › ...
1 const app = getApp()
2 // pages/books/books.js
3 Page({
4
5     /**
6      * 页面的初始数据
7      */
8     data: {
9         books:[],
10        carouselImgUrls: [
11            "../../assets/book1.jpg",
12            "../../assets/book2.jpg",
13            "../../assets/book3.jpg",
14            "../../assets/book4.jpg"
15        ],
16        searchValue:"",
17        show: false,
18        book: null,
19        active:0,
20    },
21}
```

- Book.js



The screenshot shows a code editor window with the file "book.js" open. The file path is "pages > books > book.js". The code is written in JavaScript and uses the wx API for WeChat Mini Program development. It includes an onLoad lifecycle function that checks if the user is logged in. If not, it navigates to the index page. If yes, it performs a POST request to "http://localhost:8080/getBooks" to fetch books data, logs the response, and sets it as the page's data.

```
22  /**
23   * 生命周期函数--监听页面加载
24   */
25  onLoad: function (options) {
26      var that = this;
27      if (!app.globalData.userLogged){
28          wx.navigateTo({
29              url: '../index/index'
30          })
31      }else{
32          wx.request({
33              url: 'http://localhost:8080/getBooks',
34              method: "POST",
35              data: {
36                  11:11
37              },
38              header: {
39                  'content-type': 'application/json', // 默认值
40                  'cookie': wx.getStorageSync("sessionid") //cookie
41              },
42              success(res) {
43                  console.log(res);
44                  that.setData({
45                      books: res.data
46                  })
47              }
48          });
49      }
50  },
```

- Book.js



A screenshot of a code editor showing the `book.js` file. The file contains JavaScript code with various comments and lifecycle methods. The code is color-coded, with comments in green and other text in white. The file path in the top bar is `pages > books > book.js`.

```
books.js
51
52    /**
53     * 生命周期函数--监听页面初次渲染完成
54     */
55    onReady: function () { },
56
57    /**
58     * 生命周期函数--监听页面显示
59     */
60    onShow: function () { },
61
62    /**
63     * 生命周期函数--监听页面隐藏
64     */
65    onHide: function () { },
66
67    /**
68     * 生命周期函数--监听页面卸载
69     */
70    onUnload: function () { },
71
72    /**
73     * 页面相关事件处理函数--监听用户下拉动作
74     */
75    onPullDownRefresh: function () { },
76
77    /**
78     * 页面上拉触底事件的处理函数
79     */
80    onReachBottom: function () { },
```

# Book Store – Books

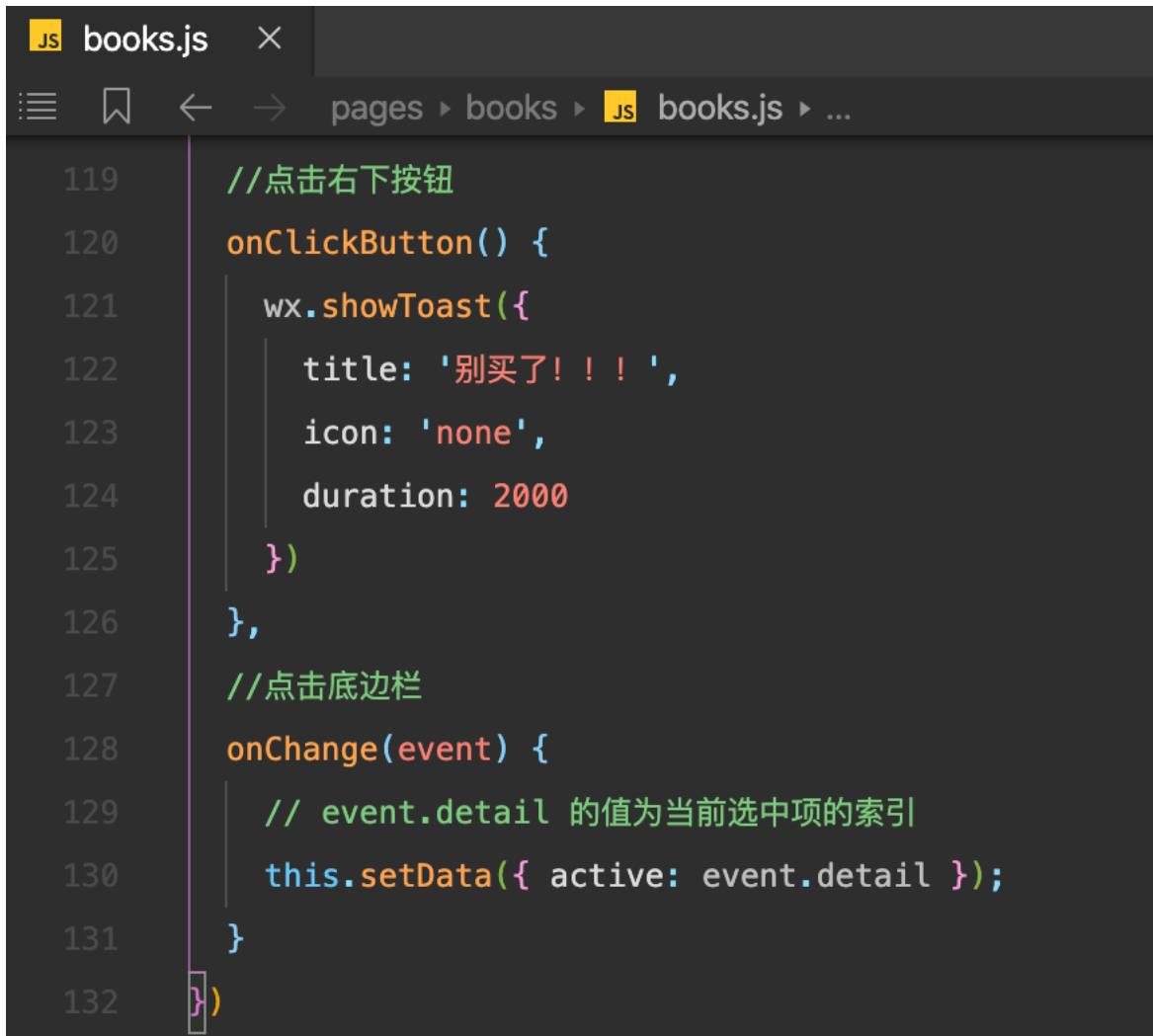


REliable, INtelligent & Scalable Systems

- Book.js

```
js books.js ×
pages › books › books.js › ...
88 //点击书籍
89 bindViewTap: function (event) {
90     this.setData({ show: true, book: event.currentTarget.dataset.book});
91     //console.log(event.currentTarget.dataset.book);
92 },
93 //搜索框
94 onChange(e) {
95     this.setData({
96         searchvalue: e.detail
97     });
98 },
99
100 onSearch() {
101     wx.showToast({
102         title: '搜索功能还没做，偷个懒！',
103         icon: 'none',
104         duration: 2000
105     })
106 },
107 //点击遮罩层
108 onClickHide() {
109     this.setData({ show: false });
110 },
111 //点击坐下图标
112 onClickIcon() {
113     wx.showToast({
114         title: '客服没钱请！购物车还没做！',
115         icon: 'none',
116         duration: 2000
117     })
118 },
```

- Book.js

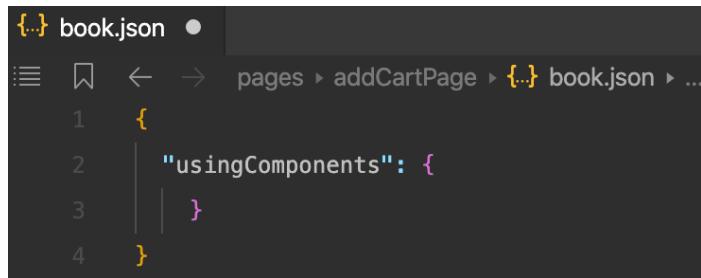


The screenshot shows a code editor window with a dark theme. The file is named "books.js" and is located in the "pages" directory under "books". The code is written in JavaScript and handles user interactions with books.

```
//点击右下按钮
onClickButton() {
  wx.showToast({
    title: '别买了!!!!',
    icon: 'none',
    duration: 2000
  })
},
//点击底边栏
onChange(event) {
  // event.detail 的值为当前选中项的索引
  this.setData({ active: event.detail });
}
})
```

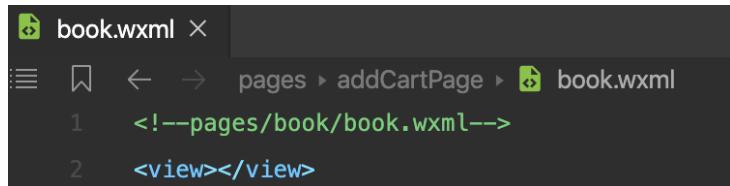
# Book Store – addCartsPage

- Book.json



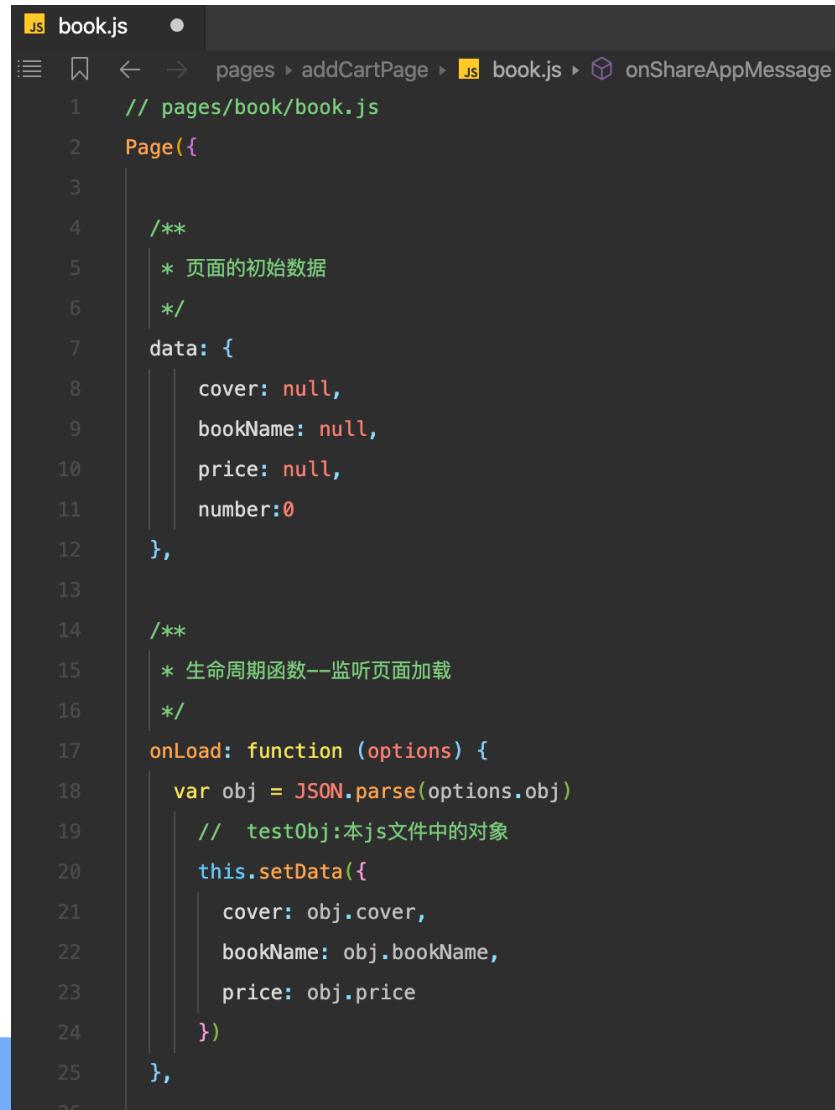
```
{-> book.json
{
  "usingComponents": {
    ...
  }
}}
```

- Book.wxml



```
book.wxml <->
<!--pages/book/book.wxml-->
<view></view>
```

- Book.js



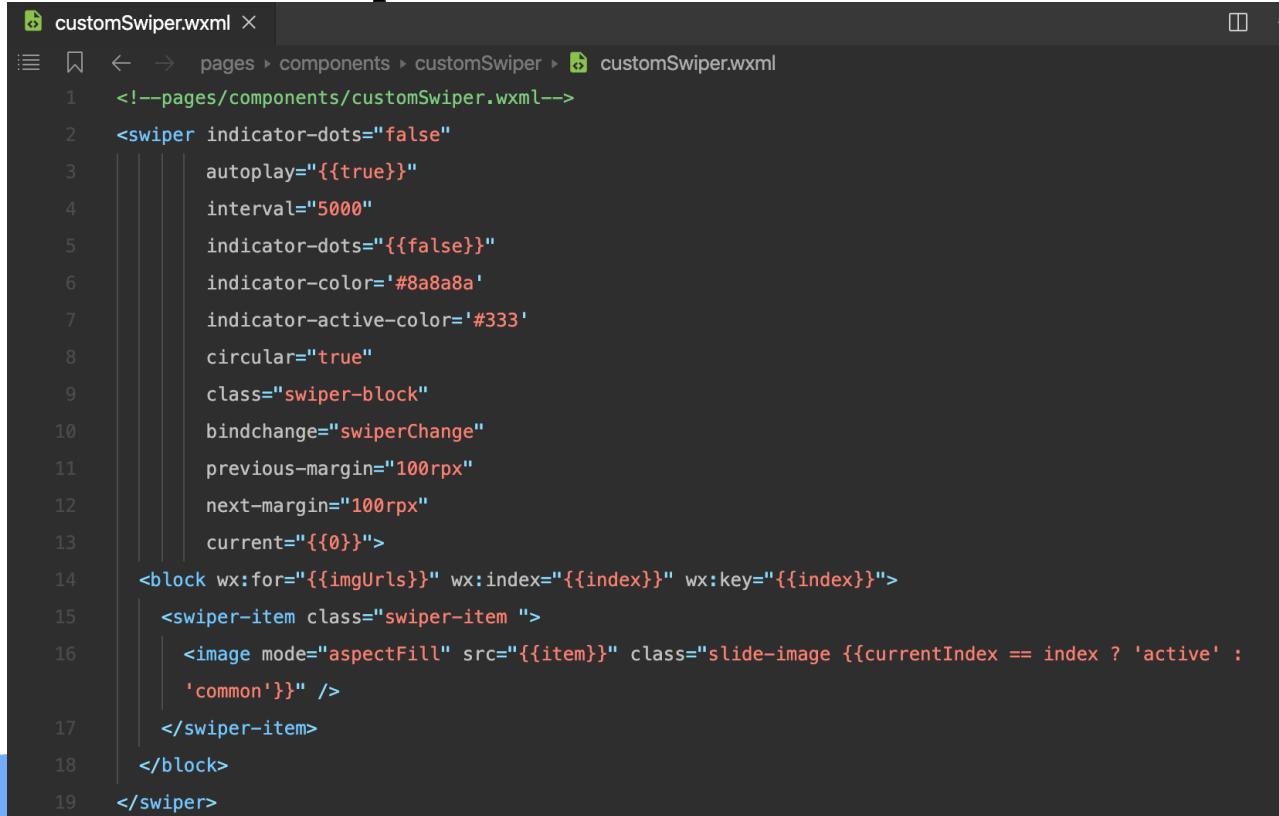
```
book.js
// pages/book/book.js
Page({
  /**
   * 页面的初始数据
   */
  data: {
    cover: null,
    bookName: null,
    price: null,
    number: 0
  },
  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function (options) {
    var obj = JSON.parse(options.obj)
    // testObj:本js文件中的对象
    this.setData({
      cover: obj.cover,
      bookName: obj.bookName,
      price: obj.price
    })
  }
})
```

- **customerSwiper.json**



```
{...} customSwiper.json ×
pages › components › customSwiper › ... customSwiper.json › ...
1 {
2   "component": true,
3   "usingComponents": {}
4 }
```

- **customerSwiper.wxml**



```
customSwiper.wxml ×
pages › components › customSwiper › ... customSwiper.wxml
1 <!--pages/components/customSwiper.wxml-->
2 <swiper indicator-dots="false"
3   |   autoplay="{{true}}"
4   |   interval="5000"
5   |   indicator-dots="{{false}}"
6   |   indicator-color="#8a8a8a"
7   |   indicator-active-color="#333"
8   |   circular="true"
9   |   class="swiper-block"
10  |  bindchange="swiperChange"
11  |  previous-margin="100rpx"
12  |  next-margin="100rpx"
13  |  current="{{0}}">
14  |  <block wx:for="{{imgUrls}}" wx:index="{{index}}" wx:key="{{index}}">
15  |    <swiper-item class="swiper-item ">
16  |      <image mode="aspectFill" src="{{item}}" class="slide-image {{currentIndex == index ? 'active' :
17  |        'common'}}" />
18  |    </swiper-item>
19  </block>
</swiper>
```

# Book Store – customerSwiper



REliable, INtelligent & Scalable Systems

- customerSwiper.js

```
JS customSwiper.js ×
pages/components/customSwiper/customSwiper.js

1 // pages/components/customSwiper.js
2 Component({
3   /**
4    * 组件的属性列表
5    */
6   properties: {
7     imgUrls: Array,
8   },
9
10  /**
11   * 组件的初始数据
12   */
13  data: {
14    currentIndex: 0
15  },
16
17  /**
18   * 组件的方法列表
19   */
20  methods: {
21    swiperChange(e) {
22      this.setData({
23        currentIndex: e.detail.current
24      });
25    }
26  }
27})
```

# Book Store – util.js



REliable, INtelligent & Scalable Systems

```
JS util.js      X
☰ ⌂ ← →  utils ▶ JS util.js ▶ ...
1 const formatTime = date => {
2   const year = date.getFullYear()
3   const month = date.getMonth() + 1
4   const day = date.getDate()
5   const hour = date.getHours()
6   const minute = date.getMinutes()
7   const second = date.getSeconds()
8
9   return [year, month, day].map(formatNumber).join('/') + ' ' + [hour, minute, second].map(formatNumber)
10  .join(':')
11
12  const formatNumber = n => {
13    n = n.toString()
14    return n[1] ? n : '0' + n
15  }
16
17  module.exports = {
18    ...
19    formatTime: formatTime
}
```

# 编码完成之后的工作



REliable, INtelligent & Scalable Systems

- 上传代码
- 提交审核
- 发布小程序

# 需要注意的点

- 发布时候需要填入自己服务器的域名
- 开发过程中把域名检测关闭掉，否则wx.request发不出去
- 微信开发工具->工具->项目详情



- 微信小程序开发文档
  - <https://developers.weixin.qq.com/miniprogram/dev/framework/>
- 微信小程序开发者社区
  - <https://developers.weixin.qq.com/>



- *Web*开发技术
- *Web Application Development*

Thank You!