46 最后一块拼图：底部菜单与错误提示页面

更新时间：2019-10-14 16:54:55

> 只有在那崎岖的小路上不畏艰险奋勇攀登的人,才有希望达到光辉的顶点。
>
> ——马克思

本节我们将配置小程序的底部菜单栏（tabBar），通过底部菜单栏将各个功能模块组合成一个完整的小程序。此外，我们还将实现异常提示页面。

1. 底部菜单栏

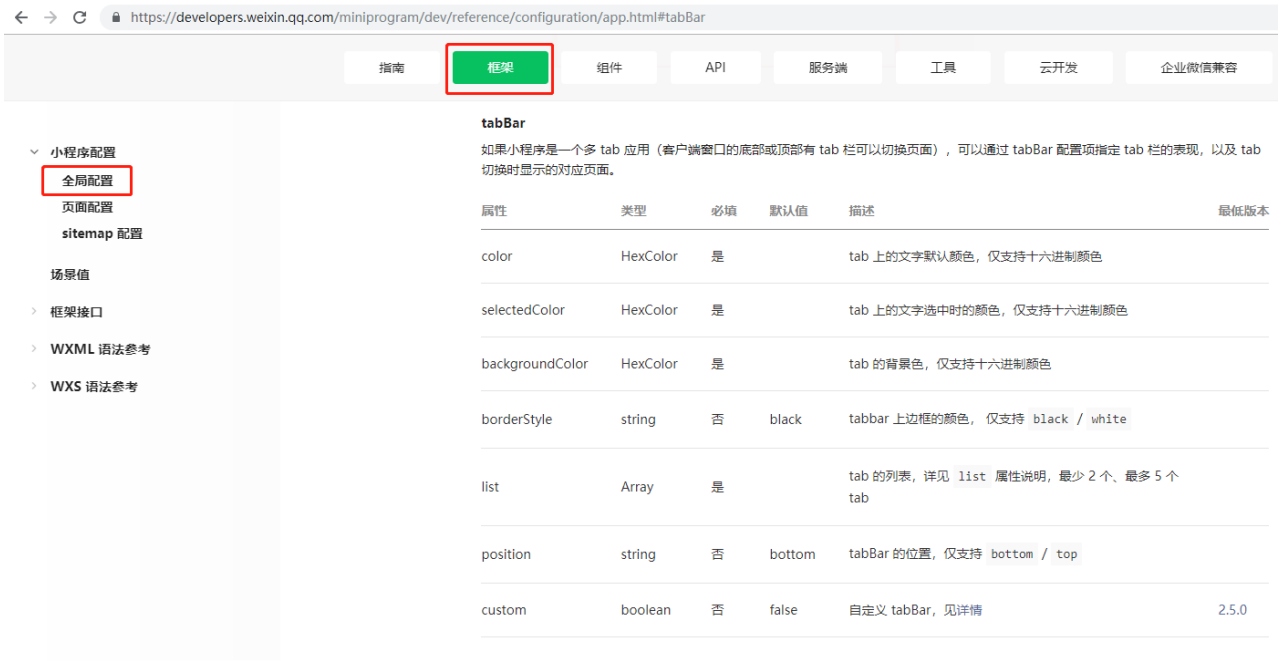
小程序的底部菜单栏叫做 tabBar ，tabBar 在 app.json 文件中进行配置。

在微信官方的 “小程序开发文档” 中对 tabBar 有详细解的说明，文档位置如下：

模板消息讲解的文档位置如下：

- 入口网址：**tabBar**，如果微信官方文档改版导致链接失效，请按图索骥。
- 入口位置：在 “小程序开发文档” 的 “框架” -> “小程序配置” -> “全局配置”，如图 13 红框标出部分。

图 13 tabBar 文档位置



在实战项目“会员制社交电商小程序”中，我们需要根据第四章第二节“3. 页面流程”的梳理在小程序的底部菜单栏配置“商城”、“社区”和“我的”三个菜单，这三个菜单对应的页面分别是商城首页、UGC 社区首页 和 个人中心页面（如图 14 中红框所示）。

图 14“会员制社交电商小程序”底部菜单

搜索

全部 Web设计 软件开发 编程语言 少

Vue.js项目实战

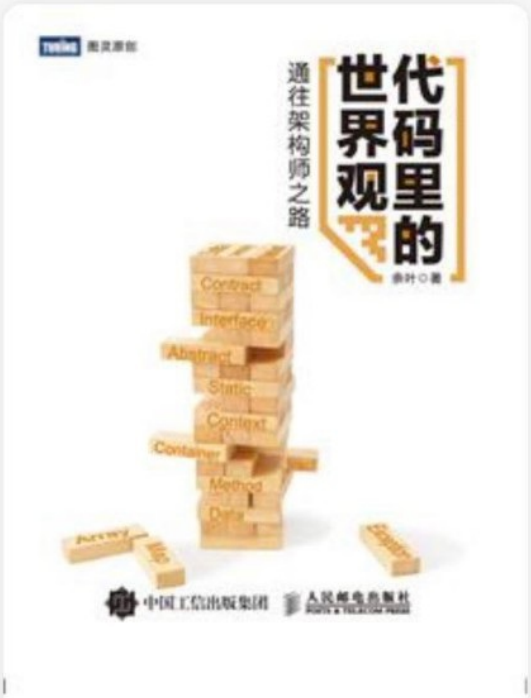
本书基于6个项目来引导读者深入理解Vue.js。书中首先...

p6900

Python深度学习

本书由Keras之父、现任Google人工智能研究员的弗...

p11900



代码里的世界观——通往架构师之路

本书分为两大部分，第一部分讲述程序员在编写程序和组...

Python基础教程（第3版）

本书包括Python程序设计的方方面面：首先从Python的安...

p9900

在阅读官方文档关于 tabBar 的说明后，我们可以按照说明在 app.json 中配置出图 14 所示的菜单：

```
"tabBar": {
  "color": "#353535",
  "selectedColor": "#09bb07",
  "borderStyle": "white",
  "backgroundColor": "#ffffff",
  "list": [
    {
      "pagePath": "pages/membership/pages/shop/index/index",
      "iconPath": "image/icon_home.png",
      "selectedIconPath": "image/icon_home_hl.png",
      "text": "商城"
    },
    {
      "pagePath": "pages/membership/pages/community/index/index",
      "iconPath": "image/icon_file.png",
      "selectedIconPath": "image/icon_file_hl.png",
      "text": "社区"
    },
    {
      "pagePath": "pages/membership/pages/my/index/index",
      "iconPath": "image/icon_user.png",
      "selectedIconPath": "image/icon_user_hl.png",
      "text": "我的"
    }
  ]
},
```

在 **app.json** 中配置底部菜单栏 **tabBar** 时需要注意以下几点：

pages 中定义的第一项（也就是小程序首页）一定要定义为 **tabBar** 中的一个菜单，否则菜单无法显示（即菜单一定要有一个初始激活态）。

例如：

- 如果我们在 app.json 中定义 pages 为：

```
"pages": [
  "pages/membership/pages/my/order/order",
  "pages/membership/pages/shop/index/index",

  ...略

],
```

则打开小程序后底部菜单不会显示。

- 如果我们在 app.json 中定义 pages 为：

```
"pages": [
  "pages/membership/pages/community/index/index",
  "pages/membership/pages/shop/index/index",

    ...略

],
```

则打开小程序后底部菜单默认选中“社区”，页面显示 UGC 社区首页。

tabBar 不能使用 ColorUI 或者其他第三方 UI 组件的 CSS 图标，只能使用图片。

在专栏源代码的 image 文件夹中已经准备好了三个菜单的 6 张图片（ 每个菜单选中状态和未选中状态的图片是两张），具体位置见本节末尾图 16 。

app.json 是一个数据文件，里面的所有内容都是不能包含注释的。

如果将刚才 tabBar 的配置写成如下这样包含注释的话，小程序将无法运行：

```
"tabBar": {
  "color": "#353535", //未选中菜单的文字颜色
  "selectedColor": "#09bb07", //选中菜单的文字颜色
  "borderStyle": "white", //底部菜单边框颜色
  "backgroundColor": "#ffffff",  //底部菜单背景色
  "list": [
    {
      "pagePath": "pages/membership/pages/shop/index/index", //菜单对应的页面
      "iconPath": "image/icon_home.png", //未选中菜单的图标
      "selectedIconPath": "image/icon_home_hl.png", //选中菜单的图标
      "text": "商城" //菜单名称
    },
    {
      "pagePath": "pages/membership/pages/community/index/index",
      "iconPath": "image/icon_file.png",
      "selectedIconPath": "image/icon_file_hl.png",
      "text": "社区"
    },
    {
      "pagePath": "pages/membership/pages/my/index/index",
      "iconPath": "image/icon_user.png",
      "selectedIconPath": "image/icon_user_hl.png",
      "text": "我的"
    }
  ]
},
```

2. 错误提示页面

当程序出现异常时，程序员看到的是 IDE 显示的 error 信息。

在处理程序异常时，新手程序员常见的错误做法是弹出含有程序 error 信息的提示框。更有甚者，直接不处理任何异常，任由 web 服务器的 error 信息直接显示在网页上（我就经常在打开一些小网站时，直接看到 web 服务器的 debug 错误信息，PHP、Java、.Net 的都有）。

作为立志成为大神的各位同学，要明白一件事情：程序不会永远按照你编写的逻辑正确执行，断网、网速慢、数据库查询超时、程序版本冲突、修复 BUG 导致新的 BUG 等等你能想到的和完全想不到的各种意外情况都会导致你编写的程序运行出错。

因此，**在一个程序中，异常处理代码编写得好不好，很大程度上决定了这个程序的质量高低，也能直接反映出一个程序员编程水平的高低。**

要想成为大神，首先需要树立正确的程序异常处理观念：

- 所有的程序异常都应该被捕获并处理，所有可能出错的代码段都需要考虑是直接捕获异常并进行处理，还是向上抛出异常；
- 我们应该对用户说人话，在程序出现异常时向用户展示异常提示页面而不是弹出含有程序 error 信息的提示框；

- 在对用户说人话的同时，还需要将程序出现异常时的程序 error 信息保存到错误日志中，通过分析错误日志可以发现和修复更多未知的 BUG。

更多有关异常处理的知识，请各位同学搜索更专业的文章或者书籍进行学习。

在实战项目“会员制社交电商小程序”中，我们已经对访问云函数或云数据库失败、出错等异常进行了捕获，并在异常处理中设置了页面跳转到错误提示页面。

例如在第九章第六节的 `getMyUpvote` 方法就会在 `catch` 中向控制台输出 error 信息，并跳转页面到错误提示页面，`catch` 代码段如下：

```
.catch(err => {
  //访问数据库失败 的 系统异常处理
  //跳转出错页面
  wx.redirectTo({
    url: '../../errorpage/errorpage'
  })
  //向控制台输出 error 信息
  //在实际产品开发中会调用后端接收错误日志的接口,传送错误信息到后端错误日志中保存
  console.error(err)
})
```

因此，实战项目“会员制社交电商小程序”的最后一块拼图是编写如图 15 所示的错误提示页面 errorpage 。

图 15 错误提示页面

出错啦

哎呀，好像我抽风啦，请“返回”再试一下~~

返回

错误提示页面可以基于 WeUI 的 Msg 组件的失败提示页做一些修改实现。

WXML 模板代码如下：

```
<!-- 访问云函数或云数据库失败、出错，显示错误提示页面 -->
<view class="errorpage">
  <view class="weui-msg">
    <view class="weui-msg__icon-area">
      <icon type="warn" size="93"></icon>
    </view>
    <view class="weui-msg__text-area">
      <view class="weui-msg__title">出错啦</view>
      <view class="weui-msg__desc">哎呀，好像我抽风啦，请“返回”再试一下~~</view>
    </view>
    <view class="weui-msg__opr-area">
      <view class="weui-btn-area">
        <!-- 在错误提示页面点击返回，回到小程序首页 -->
        <navigator open-type="switchTab" url="../shop/index/index">
          <button class="weui-btn" type="primary">返回</button>
        </navigator>
      </view>
    </view>
  </view>
</view>
```

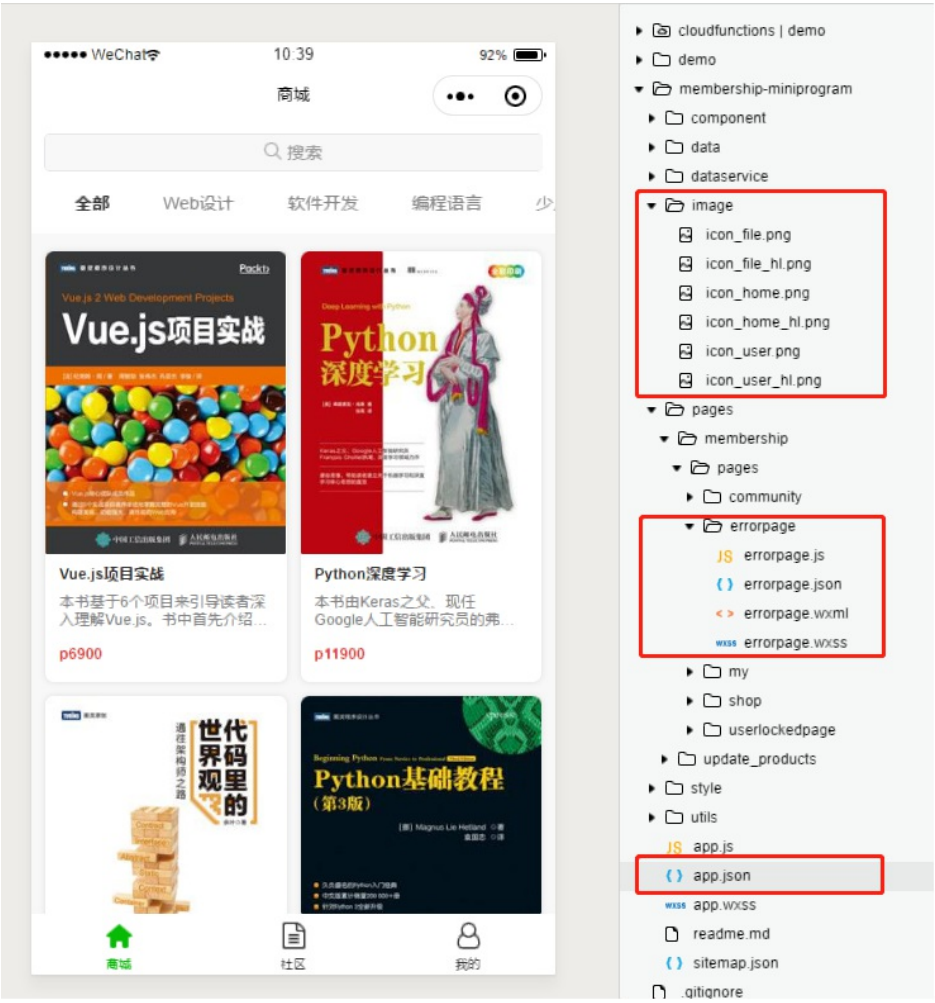
3. 专栏源代码

本专栏源代码已上传到 GitHub ， 请访问以下地址获取：

https://github.com/liujiec/Membership-ECommerce-Miniprogram

本节源代码内容在图 16 红框标出的位置。

图 16 本节源代码位置



下节预告

到本节为止，我们已经完成了“会员制社交电商小程序”的实战开发。

从下一节开始，我们将一起来考虑：如何将实战开发结果制作成作品，并使用这个作品帮助我们的职业发展。

实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 完成图 **14** 所示的小程序菜单配置，如碰到问题，请阅读本专栏源代码学习如何实现。
- 学有余力的同学可以思考：在已经完成的“会员制社交电商小程序”中，有哪些类型的错误或异常，对这些错误或异常应该分别显示什么样的错误提示内容，才可以让小程序有更好的用户体验。

}

45 无小程序不营销：自定义转发与模板消息

47 为什么你发布小程序的申请被驳回了？