

实验 5：高校新闻网

本实验来自于周文洁老师的《微信小程序开发实战》第十四章。在学习了小程序的基础知识和各类 API 以后，尝试独立动手创建一个小程序前端综合设计实例。我们将从零开始详解如何模仿网易新闻实现一个基于模拟数据的简易高校新闻小程序。

学习目标：1、综合所学知识创建完整的前端新闻小程序项目；能够在开发过程中熟练掌握真机预览、调试等操作。

注意事项：

- 1、提供 common.js、图片文件、以及 index 页面的代码，其它部分的代码大家自己完成，下载地址：https://gaopursuit.oss-cn-beijing.aliyuncs.com/2022/demo4_file.zip
- 2、detail 页面的 wxss 文件中，.poster image 中的 width 设置为 100% 的话，图片总是无法显示。修改为 700rpx 就可以显示，我不清楚什么原因，大家如果知道为什么可以告诉我。

15.1 项目创建

本项目一共需要 3 个页面，即首页、新闻页和个人中心页，其中首页和个人中心页需要以 tabBar 的形式展示，可以点击 tab 图标互相切换。

15.1.1 首页功能需求

首页功能需求如下：(1) 首页需要包含幻灯片播放效果和新闻列表；(2) 幻灯片至少要有 3 幅图片自动播放；(3) 点击新闻列表可以打开新闻全文。

15.1.2 新闻页功能需求

新闻页功能需求如下：(1) 阅读新闻全文的页面需要显示新闻标题、图片、正文和日期；(2) 允许点击按钮将当前阅读的新闻添加到本地收藏夹中；(3) 已经收藏过的新闻也可以点击按钮取消收藏。

15.1.3 个人中心页功能需求

个人中心页功能需求如下：(1) 未登录状态下显示登录按钮，用户点击以后可以显示

微信头像和昵称。(2) 登录后读取当前用户的收藏夹，展示收藏的新闻列表。(3) 收藏夹中的新闻可以直接点击查看内容。(4) 未登录状态下收藏夹显示为空。

15.2 项目创建

本项目创建选择空白文件夹 newsDemo，效果如图 15-1 所示。

视频讲解

单击“新建”按钮完成项目创建，然后准备手动创建页面配置文件。



图 15-1 小程序项目填写效果图

15.3 页面配置

15.3.1 创建页面文件

项目创建完毕后，在根目录中会生成文件夹 pages 用于存放页面文件。一般来说首页默认命名为 index，表示小程序运行的第一个页面；其他页面名称可以自定义。本项目需要创建 3 个页面文件，包括 index(首页页面)、detail(新闻页面)和 my(个人中心页面)。



视频讲解

具体操作如下：

- 将 app.json 文件 pages 属性中的 pages/logs/logs 删除；
- 在 app.json 文件 pages 属性中继续追加 pages/detail/detail 和 pages/my/my；
- 按快捷键 Ctrl+S 保存修改，然后会在 pages 文件夹下自动生成 detail 和 my 目录。

15.3.2 删 除和修改文件

全部完成后的目录结构如图 15-6 所示。



图 15-5 新建 JS 文件



图 15-6 页面文件创建完成

此时，页面文件配置全部完成，15.4 节将正式进行页面布局和样式设计。

15.4 视图设计

小程序默认导航栏是黑底白字的效果，可以通过在 `app.json` 中对 `window` 属性进行重新配置来自定义导航栏效果。更改后的 `app.json` 文件代码如下：

```
1. {
2.   "pages": [代码略],
3.   "window": {
4.     "navigationBarBackgroundColor": "#328EEB",
5.     "navigationBarTitleText": "我的新闻网",
6.     "navigationBarTextStyle": "white"
7.   }
8. }
```



视频讲解



上述代码可以更改导航栏背景色为蓝色、字体为白色，效果如图 15-7 所示。

图 15-7 自定义导航栏效果

15.4.2 tabBar 设计

首先在 app.json 中追加 tabBar 的相关属性代码,更改后的 app.json 文件代码如下:

```
1.  {
2.      "pages": [ 代码略 ],
3.      "window": { 代码略 },
4.      "tabBar": {
5.          "color": "#000",
6.          "selectedColor": "#328EEB",
7.          "list": [
8.              {
9.                  "pagePath": "pages/index/index",
10.                 "iconPath": "images/index.png",
11.                 "selectedIconPath": "images/index_blue.png",
12.                 "text": "首页"
13.             },
14.             {
15.                 "pagePath": "pages/my/my",
16.                 "iconPath": "images/my.png",
17.                 "selectedIconPath": "images/my_blue.png",
18.                 "text": "我的"
19.             }
20.         ]
21.     }
22. }
```



视频讲解



运行效果如图 15-8 所示,此时已经可以切换首页和个人中心页了。

图 15-8 tabBar 完成效果图

15.4.3 页面设计

1 首页设计

首页主要包含两部分内容,即幻灯片滚动和新闻列表,页面设计如图 15-9 所示。



视频讲解

计划使用如下组件。

- 幻灯片: <swiper>组件;
- 新闻列表: <view>容器,内部使用数组循环。



图 15-9 首页设计图

WXML(pages/index/index. wxml)代码如下：

```
1. <!-- 幻灯片滚动 -->
2. <swiper indicator-dots autoplay interval="5000" duration="500"></swiper>
3. <!-- 新闻列表 -->
4. <view id='news-list'>这是新闻列表</view>
```

接着为组件添加 wx:for 属性循环显示幻灯片内容和新闻列表数据。

修改后的 WXML(pages/index/index. wxml)代码如下：

```
1. <!-- 幻灯片滚动 -->
2. <swiper indicator-dots="true" autoplay="true" interval="5000" duration="500">
3.   <block wx:for="{{swiperImg}}" wx:key='swiper{{index}}'>
4.     <swiper-item>
5.       <image src="{{item.src}}" class="slide-image" />
6.     </swiper-item>
7.   </block>
8. </swiper>
9. <!-- 新闻列表 -->
10. <view id='news-list'>
11.   <view class='list-item' wx:for="{{newsList}}" wx:for-item="news" wx:key="{{news.id}}">
12.     <image src="{{news.poster}}></image>
13.     <text>◇{{news.title}} -- {{news.add_date}}</text>
14.   </view>
15. </view>
```

相关 WXSS(pages/index/index. wxss)代码片段如下：

```
1. /* swiper 区域样式 */
2. /* 1-1 swiper 组件 */
```

```
3. swiper {
4.   height: 400rpx;
5. }
6. /* 1 - 2 swiper 中的图片 */
7. swiper image {
8.   width: 100% ;
9.   height: 100% ;
10.}
11./* 新闻列表区域样式 */
12./* 2 - 1 新闻列表容器 */
13. #news-list {
14.   min-height: 600rpx;
15.   padding: 15rpx;
16. }
17./* 2 - 2 列表项目 */
18. .list-item{
19.   display: flex;
20.   flex-direction: row;
21.   border-bottom: 1rpx solid gray;
22. }
23./* 2 - 3 新闻图片 */
24. .list-item image{
25.   width:230rpx;
26.   height: 150rpx;
27.   margin: 10rpx;
28. }
29./* 2 - 4 新闻标题 */
30. .list-item text{
31.   width: 100% ;
32.   line-height: 60rpx;
33.   font-size: 10pt;
34. }
```

下面代码中，swiperImg 的链接已经失效，我提供新的路径如下。临时新闻数据可以在 common.js 里面找一条来替换

```
JavaScript
/**
 * 页面的初始数据
 */
data: {
  //幻灯片素材
  swiperImg: [
    {src: 'https://gaopursuit.oss-cn-beijing.aliyuncs.com/2022/newsimage1.jpg'},
    {src: 'https://gaopursuit.oss-cn-beijing.aliyuncs.com/2022/newsimage2.jpg'},
    {src: 'https://gaopursuit.oss-cn-beijing.aliyuncs.com/2022/newsimage3.jpg'}
  ],
},
```

为了进行布局和样式效果的预览,还需要在 JS 文件的 data 中临时录入几个测试数据。

相关 JS(pages/index/index.js)代码片段如下:

```
1. Page({
2.   data: {
3.     //幻灯片素材
4.     swiperImg: [
5.       { src: 'http://www.ahnu.edu.cn/_local/A/C7/68/C2C9E5E2161A466A2D54D21A63C_DD3FEC40_4EBBB.jpg?e=.jpg'},
6.       { src: 'http://www.ahnu.edu.cn/_local/7/CC/BD/47349A7168770AC24EB5535B2AF_35829829_6E24D.png?e=.png'},
7.       { src: 'http://www.ahnu.edu.cn/_local/2/BB/2D/9FEB9B8D0CA5E059B2C1C7E65BD_67A0410C_3A71F.png?e=.png'}
8.     ],
9.     //临时新闻数据
10.    newsList:[{
11.      id: '264698',
12.      title: '俄罗斯联邦驻华大使杰尼索夫会见校党委书记顾家山一行并接受《力冈译文全集》赠予',
13.      poster: 'http://www.ahnu.edu.cn/_local/A/C7/68/C2C9E5E2161A466A2D54D21A63C_DD3FEC40_4EBBB.jpg?e=.jpg',
14.      add_date: '2018-03-05'
15.    }]
16.  }
17. })
```

当前效果如图 15-10 所示。

由图可见,此时可以显示幻灯片播放和一条临时新闻。由于尚未获得新闻数据,所以暂时无法显示完整的新闻列表,仅供作为样式参考。



视频讲解

2 个人中心页设计

个人中心页主要包含两个版块,即登录面板和“我的收藏”。登录面板用于显示用户的微信头像和昵称,“我的收藏”用于显示收藏在本地的新闻列表。页面设计如图 15-11 所示。



图 15-10 首页效果图



图 15-11 个人中心页设计图

计划使用<view>组件进行整体布局,对自定义的 id 名称解释如下。

- myLogin: 登录面板;
- myIcon: 微信头像图片;
- nickName: 微信昵称;
- myFavorites: 我的收藏。

WXML(pages/my/my.wxml)代码如下:

```
1. <!-- 登录面板 -->
2. <view id = 'myLogin'> </view>
3. <!-- 我的收藏 -->
4. <view id = 'myFavorites'> </view>
```

接着为这两个区域添加内容,修改后的 WXML(pages/my/my.wxml)代码如下:

```
1. <!-- 登录面板 -->
2. <view id = 'myLogin'>
```

```

3.   <block>
4.     <image id = 'myIcon' src = '{{src}}'></image>
5.     <text id = 'nickName'>{{nickName}}</text>
6.   </block>
7. </view>
8. <!-- 我的收藏 -->
9. <view id = 'myFavorites'>
10.  <text>我的收藏(1)</text>
11.  <!-- 收藏的新闻列表 -->
12.  <view id = 'news - list'>
13.    <view class = 'list - item' wx:for = "{{newsList}}" wx:for-item = "news" wx:key =
"{{news.id}}">
14.      <image src = '{{news.poster}}'></image>
15.      <text>◇{{news.title}} -- {{news.add_date}}</text>
16.    </view>
17.  </view>
18. </view>

```

WXSS(pages/my/my.wxss)代码如下：

```

1. /* 登录面板 */
2. #myLogin{
3.   background-color: #328EEB;
4.   height: 400rpx;
5.   display: flex;
6.   flex-direction: column;
7.   align-items: center;
8.   justify-content: space-around;
9. }
10. /* 1-1 头像图片 */
11. #myIcon{
12.   width: 200rpx;
13.   height: 200rpx;
14.   border-radius: 50% ;
15. }
16. /* 1-2 微信昵称 */
17. #nickName{
18.   color: white;
19. }
20. /* 我的收藏 */
21. #myFavorites{
22.   padding: 20rpx;
23. }

```

由于新闻列表的样式与首页完全相同，没有必要重复样式代码，否则会造成冗余，可以将index.wxss中新闻列表样式的相关代码挪到app.wxss中公共使用。

app.wxss的代码如下：

```

1. /* 新闻列表区域样式 */
2. /* 2-1 新闻列表容器 */
3. #news-list {
4.   min-height: 600rpx;
5.   padding: 15rpx;

```

```
6. }
7. /* 2-2 列表项目 */
8. .list-item{
9.   display: flex;
10.  flex-direction: row;
11.  border-bottom: 1rpx solid gray;
12. }
13. /* 2-3 新闻图片 */
14. .list-item image{
15.   width: 230rpx;
16.   height: 150rpx;
17.   margin: 10rpx;
18. }
19. /* 2-4 新闻标题 */
20. .list-item text{
21.   width: 100%;
22.   display: block;
23.   line-height: 60rpx;
24.   font-size: 10pt;
25. }
```

下面代码中的临时数据可以在 common.js 里面找一条来替换。

为了进行布局和样式效果的预览,还需要在 JS 文件的 data 中临时录入测试数据。

JS(pages/my/my.js)代码片段如下:

```
1. Page({
2.   data: {
3.     //临时微信用户昵称和头像
4.     nickName: '未登录',
5.     src: '/images/index.png',
6.     //临时收藏夹新闻数据
7.     newsList: [
8.       id: '264698',
9.       title: '俄罗斯联邦驻华大使杰尼索夫会见校党委书记顾家山一行并接受《力冈译文全集》赠予',
10.      poster: 'http://www.ahnu.edu.cn/_local/A/C7/68/C2C9E5E2161A466A2D54D21A63C_DD3FEC40_4EBB.jpg?e=.jpg',
11.      add_date: '2018-03-05'
12.    ]
13.  }
14. })
```

当前效果如图 15-12 所示。

由图可见,此时可以显示完整的样式效果。由于尚未获得微信用户数据和收藏在本地的缓存数据,所以暂时无法显示实际内容,仅供作为样式参考。

3 新闻页设计

新闻页是用于给用户浏览新闻全文的,需要用户点击首页的新闻列表,然后在新窗口中打开该页面。新闻页包括新闻标题、新闻图片、新闻正文和新闻日期,页面设计如图 15-13 所示。

由于暂时没有做点击跳转的逻辑设计,所以可以在微信 web 开发者工具顶端工具栏中找到“普通编译”下拉选项,选择“添加编译模式”,然后追加对于 detail 页面的直接浏览效果,如图 15-14 所示。

此时预览就可以直接显示 detail 页面了,设计完毕后再切换回“普通编译”模式显示首页(index)即可。



视频讲解



图 15-12 个人中心页效果图



图 15-13 新闻页设计图



图 15-14 添加新闻页的编译模式

计划使用`<view>`组件进行整体布局,对自定义的 class 名称解释如下。

- container: 整体容器;
- title: 新闻标题区域;
- poster: 新闻图片区域;
- content: 新闻正文区域;
- add_date: 新闻日期区域。

WXML(pages/detail/detail.wxml)代码如下:

```
1. <view class = 'container'>
2.   <view class = 'title'>{{article.title}}</view>
3.   <view class = 'poster'>
4.     <image src = '{{article.poster}}' mode = 'widthFix'></image>
5.   </view>
6.   <view class = 'content'>{{article.content}}</view>
7.   <view class = 'add_date'>时间: {{article.add_date}}</view>
8. </view>
```

我实现了一下,下面代码中, .poster image 中的 width 设置为 100% 的话, 图片总是无法显示。修改为 700rpx 就可以显示。具体原因不是特别清楚,大家如果知道为什么

可以告诉我。

WXSS(pages/detail/detail.wxss)代码如下：

```
1. /* 整体容器 */
2. .container{
3.   padding: 15rpx;
4.   text-align: center;
5. }
6. /* 新闻标题 */
7. .title{
8.   font-size: 14pt;
9.   line-height: 80rpx;
10. }
11. /* 新闻图片 */
12. .poster image{
13.   width: 100% ;
14. }
15. /* 新闻正文 */
16. .content{
17.   text-align: left;
18.   font-size: 12pt;
19.   line-height: 60rpx;
20. }
21. /* 新闻日期 */
22. .add_date{
23.   font-size: 12pt;
24.   text-align: right;
25.   line-height: 30rpx;
26.   margin-right: 25rpx;
27.   margin-top: 20rpx;
28. }
```

为了进行布局和样式效果的预览，还需要在 JS 文件的 data 中临时录入一条测试数据。

JS(pages/detail/detail.js)代码片段如下：

```
1. Page({
2.   data: {
3.     article: {
4.       id: '264698',
5.       title: '俄罗斯联邦驻华大使杰尼索夫会见校党委书记顾家山一行并接受《力冈译文全集》赠予',
6.       poster: 'http://www.ahnu.edu.cn/_local/A/C7/68/C2C9E5E2161A466A2D54D21A63C_DD3FEC40_4EBBB.jpg?e=.jpg',
7.       content: '本网讯(校出版社)3月2日上午,俄罗斯驻华大使杰尼索夫在北京俄罗斯驻华大使馆会见了校党委书记顾家山,并接受了我校出版社赠予俄罗斯大使馆的十套《力冈译文全集》。俄罗斯驻华大使馆参赞梅利尼科娃、大使馆一秘伊戈尔、大使助理、塔斯社记者,我校校办主任曾黎明、出版社社长张奇才,我校杰出校友、俄罗斯人民友谊勋章和利哈乔夫院士奖获得者、中国俄罗斯文学研究会会长刘文飞教授等参加了会见。',
8.       add_date: '2018-03-05'
9.     }
10.   }
11. })
```



图 15-15 新闻页效果图

当前效果如图 15-15 所示。

由图可见,此时可以显示完整的样式效果。由于尚未获得新闻数据,所以暂时无法根据用户点击的新闻标题入口显示对应的新闻内容,仅供作为样式参考。

此时页面布局与样式设计就已完成,15.5节将介绍如何进行逻辑处理。

15.5 逻辑实现

15.5.1 公共逻辑

common.js 使用我提供的文件即可。

正常来说数据应该由网站群管理平台提供新闻接口,由于隐私安全、开发者条件限制等一系列问题,这里采用模拟数据进行代替。有条件的开发者可以使用第三方免费或付费新闻接口(例如聚合数据等),或自行搭建服务器提供接口。



视频讲解

假设已经获取到了数据,将其放在公共 JS 文件(utils/common.js)中,代码片段如下:

```
1. const news = [{
2.   id: '264698',
3.   title: '俄罗斯联邦驻华大使杰尼索夫会见校党委书记顾家山一行并接受《力冈译文全集》赠予',
4.   poster: 'http://www.ahnu.edu.cn/_local/A/C7/68/C2C9E5E2161A466A2D54D21A63C_DD3FEC40_4EBBB.jpg?e=.jpg',
5.   content: '本网讯(校出版社)3月2日上午,俄罗斯驻华大使杰尼索夫在北京俄罗斯驻华大使馆会见了校党委书记顾家山,并接受了我校出版社赠予俄罗斯大使馆的十套《力冈译文全集》。俄罗斯驻华大使馆参赞梅利尼科娃、大使馆一秘伊戈尔、大使助理、塔斯社记者,我校校办主任曾黎明、出版社社长张奇才,我校杰出校友、俄罗斯人民友谊勋章和利哈乔夫院士奖获得者、中国俄罗斯文学研究会会长刘文飞教授等参加了会见。',
6.   add_date: '2018-03-05'
7. },
8. {
9.   id: '305670',
10.  title: '我校学子在全省第八届少数民族传统体育运动会上喜获佳绩',
11.  poster: 'http://www.ahnu.edu.cn/_local/7/CC/BD/47349A7168770AC24EB5535B2AF_35829829_6E24D.png?e=.png',
12.  content: '本网讯(体育学院 邹华刚)11月18日至23日,由安徽省人民政府主办,省民委、省体育局和蚌埠市人民政府承办的安徽省第八届少数民族传统体育运动会在蚌埠市成功举行,全省16个地市代表团近1300名运动员、教练员、裁判员参加了本次运动会。本届运动会共设武术、民族式摔跤、毽球、蹴球、押加、高脚竞速、陀螺和板鞋竞速等8个项目。我校组建了由周慧雅、王和章、王强等23名少数民族学生组成的运动员队伍,代表芜湖市参加了高脚竞速、武术、蹴球、陀螺和板鞋竞速等5个大项的比赛。经过激烈角逐,我校运动健儿共获得4个一等奖(第1名),6个二等奖(第2~4名),9个三等奖(第5~8名)的优异战绩,出色地完成了比赛任务。',
13.  add_date: '2018-11-27'
14. },
15. {
16.   id: '304083',
17.   title: '我校学子代表国家队获中国羽毛球公开赛男双亚军',
18.   poster: 'http://www.ahnu.edu.cn/_local/2/BB/2D/9FEB9B8D0CA5E059B2C1C7E65BD_67A0410C_3A71F.png?e=.png',
19.   content: '本网讯(体育学院 徐梦涛)11月11日,世界羽联中国羽毛球公开赛在福州落下帷幕。在男子双打半决赛中,我校2018级运动训练专业学生谭强与搭档何济霆以2:0战胜印尼组合莫哈末·阿山/亨德拉,晋级决赛。决赛中,这对年轻组合以1:2负于现世界排名第一的印度尼西亚组合,获得了本次比赛的亚军。这也是谭强在本年度内获得的最好成绩。',
20.   add_date: '2018-11-14'
21. }
```

```
22. ];
```

这里用了 3 条新闻记录作为示范,开发者可以自行添加或修改新闻内容。

接下来在 common.js 中添加自定义函数 getNewsList 和 getNewsDetail, 分别用于获取新闻列表信息和指定 ID 的新闻正文内容。代码片段如下:

```
1. //获取新闻列表
2. function getNewsList() {
3.     let list = [];
4.     for (var i = 0; i < news.length; i++) {
5.         let obj = {};
6.         obj.id = news[i].id;
7.         obj.poster = news[i].poster;
8.         obj.add_date = news[i].add_date;
9.         obj.title = news[i].title;
10.        list.push(obj);
11.    }
12.    return list; //返回新闻列表
13. }
14.
15. //获取新闻内容
16. function getNewsDetail(newsID) {
17.     let msg = {
18.         code: '404', //没有对应的新闻
19.         news: {}
20.     };
21.     for (var i = 0; i < news.length; i++) {
22.         if (newsID == news[i].id) { //匹配新闻 ID 编号
23.             msg.code = '200'; //成功
24.             msg.news = news[i]; //更新当前新闻内容
25.             break;
26.         }
27.     }
28.     return msg; //返回查找结果
29. }
```

最后需要在 common.js 中使用 module.exports 语句暴露函数出口,代码片段如下:

```
1. module.exports = {
2.     getNewsList: getNewsList,
3.     getNewsDetail: getNewsDetail
4. }
```

现在就完成了公共逻辑处理的部分。

然后需要在各页面的 JS 文件顶端引用公共 JS 文件,引用代码如下:

```
var common = require('../utils/common.js') //引用公共 JS 文件
```

需要注意这里暂时还不支持绝对路径引用,只能使用相对路径。

15.5.2 首页逻辑

首页主要有两个功能需要实现,一是展示新闻列表,二是点击新闻标题可以跳转对应的内容页面进行浏览。

1 新闻列表展示

新闻列表展示使用了{{newsList}}，因此需要在页面 JS 文件的 onLoad 函数中获取新闻列表，并更新到 data 属性的 newsList 参数中。

相关 JS(pages/index/index.js)代码片段如下：

```
1. Page({  
2.   onLoad: function(options) {  
3.     //获取新闻列表  
4.     let list = common.getNewsList()  
5.     //更新列表数据  
6.     this.setData({newsList:list})  
7.   }  
8. })
```

此时页面效果如图 15-16 所示。

2 点击跳转新闻内容

若希望用户点击新闻标题即可实现跳转，需要首先为新闻列表项目添加点击事件。

相关 WXML(pages/index/index.wxml)代码片段修改如下：

```
1. <!-- 新闻列表 -->  
2. <view id='news-list'>  
3.   <view class='list-item' wx:for="{{newsList}}"  
wx:for-item="news" wx:key="{{news.id}}">  
4.     <image src="{{news.poster}}'/></image>  
5.     <text bindtap='goToDetail' data-id="{{news.  
id}}>◆{{news.title}}--{{news.add_date}}</text>  
6.   </view>  
7. </view>
```

具体修改为第 5 行加粗字体部分，为< text >组件添加了自定义触摸事件函数 goToDetail，并且使用 data-id 属性携带了新闻 ID 编号。

然后在对应的 detail.js 文件中添加 goToDetail 函数的内容，代码片段如下：

```
1. Page({  
2.   /**  
3.    * 自定义函数 -- 跳转新页面浏览新闻内容  
4.    */  
5.   goToDetail: function(e) {  
6.     //获取携带的 data-id 数据  
7.     let id = e.currentTarget.dataset.id;  
8.     //携带新闻 ID 进行页面跳转  
9.     wx.navigateTo({  
10.       url: '../detail/detail?id=' + id  
11.     })  
12.   }  
13. })
```

此时已经可以点击跳转到 detail 页面，并且成功携带了新闻 ID 数据，但是仍在 detail 页面进行携带数据的接收处理才可显示正确的新闻内容。



视频讲解



图 15-16 首页新闻列表展示

15.5.3 新闻页逻辑

新闻页主要有两个功能需要实现,一是显示对应新闻,二是可以添加/取消新闻收藏。



视频讲解

在首页逻辑中已经实现了页面跳转并携带了新闻 ID 编号,现在需要在新闻页接收 ID 编号,并查询对应的新闻内容。

相关 JS(pages/detail/detail.js)代码片段如下:

```
1. Page({  
2.   onLoad: function(options) {  
3.     //获取页面跳转来时携带的数据  
4.     let id = options.id  
5.     let result = common.getNewsDetail(id)  
6.     //获取到新闻内容  
7.     if(result.code == '200'){  
8.       this.setData({article:result.news})  
9.     }  
10.  }  
11. })
```

此时重新从首页点击新闻跳转就可以发现已经能够正确显示标题对应的新闻内容了。运行效果如图 15-17 所示。



图 15-17 在首页列表中浏览新闻的效果

2 添加/取消新闻收藏

修改 detail.wxml 代码,追加两个<button>组件作为添加/取消收藏新闻的按钮,并使用

wx:if 和 wx:else 属性使其每次只存在一个。

WXML(pages/detail/detail.wxml)代码片段添加如下：

```
1. <view class = 'container'>
2.   <view class = 'title'>{{article.title}}</view>
3.   <view class = 'poster'>
4.     <image src = '{{article.poster}}' mode = 'widthFix'></image>
5.   </view>
6.   <view class = 'content'>{{article.content}}</view>
7.   <view class = 'add_date'>时间: {{article.add_date}}</view>
8.   <button wx:if = '{{isAdd}}' plain bindtap = 'cancelFavorites'>♥已收藏</button>
9.   <button wx:else plain bindtap = 'addFavorites'>♥点击收藏</button>
10. </view>
```



视频讲解

对应的 WXSS(pages/detail/detail.wxss)代码片段添加如下：

```
1. /* “点击收藏”按钮 */
2. button{
3.   width: 250rpx;
4.   height: 100rpx;
5.   margin: 20rpx auto;
6. }
```

对应 JS(pages/detail/detail.js)中的 onLoad 函数代码片段修改如下：

```
1. onLoad: function(options) {
2.   //获取页面跳转来时携带的数据
3.   let id = options.id
4.
5.   //检查当前新闻是否在收藏夹中
6.   var article = wx.getStorageSync(id)
7.   //已存在
8.   if (article != '') {
9.     this.setData({
10.       article:article,
11.       isAdd: true
12.     })
13.   }
14.   //不存在
15.   else {
16.     let result = common.getNewsDetail(id)
17.     //获取到新闻内容
18.     if (result.code == '200') {
19.       this.setData({
20.         article: result.news,
21.         isAdd: false
22.       })
23.     }
24.   }
25. },
```

继续在 detail.js 文件中追加 addFavorites 和 cancelFavorites 函数，用于点击添加/取消新闻收藏。对应 JS(pages/detail/detail.js)中的函数代码片段修改如下：

```
1. Page({
2.   //添加到收藏夹
3.   addFavorites: function(options) {
```

```

4.     let article = this.data.article;           //获取当前新闻
5.     wx.setStorageSync(article.id, article);   //添加到本地缓存
6.     this.setData({ isAdd: true });             //更新按钮显示
7.   },
8.   //取消收藏
9.   cancelFavorites: function() {
10.     let article = this.data.article;          //获取当前新闻
11.     wx.removeStorageSync(article.id);         //从本地缓存删除
12.     this.setData({ isAdd: false });           //更新按钮显示
13.   },
14. }

```

现在从首页开始预览，选择其中任意一篇新闻进入 detail 页面，并尝试点击按钮收藏和取消收藏。

此时页面效果如图 15-18 所示。



图 15-18 新闻页中“点击收藏”按钮的使用效果

15.5.4 个人中心页逻辑

个人中心页主要有3个功能需要实现,一是获取微信用户信息;二是获取收藏列表;三是浏览收藏的新闻。

1 获取微信用户信息

修改 my.wxml 代码,追加<button>组件作为登录按钮,并且使用 wx;if 和 wx:else 属性让未登录时只显示按钮,登录后只显示头像和昵称。

WXML(pages/my/my.wxml)代码
片段修改如下:

```
1. <view id = 'myLogin'>
2.   <block wx:if = '{{isLogin}}'>
3.     <image id = 'myIcon' src = '{{src}}'></image>
4.     <text id = 'nickName'>{{nickName}}</text>
5.   </block>
6.   <button wx:else>未登录,点此登录</button>
7. </view>
```

此时页面效果如图 15-19 所示。

在 my.wxml 页面修改<button>组件的代码,为其追加获取用户信息事件,代码片段如下:

```
1. <button wx:else open-type = 'getUserInfo'
bindgetuserinfo = 'getMyInfo'>
2.   未登录,点此登录
3. </button>
```

其中 open-type='getUserInfo'表示激活获取微信用户信息功能,然后使用 bindgetuserinfo 属性表示获得的数据将传递给自定义函数 getMyInfo,开发者也可以使用其他名称。

在 my.js 文件的 Page() 内部追加 getMyInfo 函数,代码片段如下:

```
1. getMyInfo: function(e) {
2.   console.log(e.detail.userInfo)
3. },
```

保存后预览项目,单击按钮后如果 Console 控制台能够成功输出用户信息数据,则说明获取成功,如图 15-20 所示。

修改 my.js 文件中 getMyInfo 函数的代码,将信息更新到动态数据上,代码片段如下:

```
1. //获取微信用户信息
2. getMyInfo: function(e) {
3.   let info = e.detail.userInfo;
4.   this.setData({
5.     isLogin:true, //确认登录状态
6.     src: info.avatarUrl, //更新图片来源
7.     nickName: info.nickName //更新昵称
8.   })
9. },
```



视频讲解



图 15-19 个人中心未登录状态效果图



图 15-20 Console 控制台输出内容

此时就已完成登录功能，预览效果如图 15-21 所示。



图 15-21 个人中心获取微信用户信息效果图

2 获取收藏列表

修改 my.wxml 代码，将“我的收藏”后面的数字更改为动态数据效果。

WXML(pages/my/my.wxml)代码片段如下：

```
<text>我的收藏({{num}})</text>
```

对应 JS(pages/detail/detail.js)中的 data 属性代码片段修改如下：

```
1. Page({
2.   data: {
3.     ...
4.     num: 0
5.   }
6. })
```



视频讲解

继续在 detail.js 文件中追加 getMyFavorites 函数,用于展示真正的新闻收藏列表。对应的 JS(pages/my/my.js)代码片段如下:

```
1.  Page({
2.    //获取收藏列表
3.    getMyFavorites: function() {
4.      let info = wx.getStorageSync(); //读取本地缓存信息
5.      let keys = info.keys; //获取全部 key 信息
6.      let num = keys.length; //获取收藏新闻数量
7.
8.      let myList = [];
9.      for (var i = 0; i < num; i++) {
10.        let obj = wx.getStorageSync(keys[i]);
11.        myList.push(obj); //将新闻添加到数组中
12.      }
13.      //更新收藏列表
14.      this.setData({
15.        newsList: myList,
16.        num: num
17.      });
18.    }
19.  })
```

修改 my.js 中的 getMyInfo 函数,为其追加关于 getMyFavorites 函数的调用。对应的 JS(pages/my/my.js)代码片段如下:

```
1.  Page({
2.    //获取微信用户信息
3.    getMyInfo: function(e) {
4.      let info = e.detail.userInfo;
5.      this.setData({
6.        isLogin: true, //确认登录状态
7.        src: info.avatarUrl, //更新图片来源
8.        nickName: info.nickName //更新昵称
9.      })
10.     //获取收藏列表
11.     this.getMyFavorites();
12.   }
13. })
```

现在从首页开始预览,选择其中任意两篇新闻进入 detail 页面,并尝试点击收藏。然后退出切换到个人中心页,登录后查看收藏效果。

此时页面效果如图 15-22 所示。

考虑到登录成功后用户还可以手动更改新闻的收藏状态,因此修改 my.js 中的 onShow 函数,判断如果是登录状态就刷新一下收藏列表。

对应的 JS(pages/my/my.js)代码片段如下:

```
1.  Page({
2.    /**
3.     * 生命周期函数--监听页面显示
```



图 15-22 “我的收藏”列表效果

```
4.      * /
5.      onShow: function() {
6.          //如果已经登录
7.          if(this.data.isLogin){
8.              //更新收藏列表
9.              this.getMyFavorites()
10.         }
11.     },
12. })
```

3 浏览收藏的新闻

点击浏览已经收藏的新闻和首页的点击跳转新闻内容功能类似,首先修改 my. wxml 收藏列表的代码如下:



视频讲解

```
1.      <!-- 收藏的新闻列表 -->
2.      <view id = 'news-list'>
3.          <view class = 'list-item' wx:for = "{{newsList}}" wx:for-item = "news" wx:key = "{{news.id}}">
4.              <image src = '{{news.poster}}'></image>
5.              <text bindtap = 'goToDetail' data-id = '{{news.id}}'>
6.                  ◇{{news.title}}—{{news.add_date}}
7.              </text>
8.          </view>
```

具体修改为第 5 行加粗字体部分,为< text >组件添加了自定义触摸事件函数 goToDetail,并且使用 data-id 属性携带了新闻 ID 编号。

然后在对应的 my. js 文件中添加 goToDetail 函数的内容,代码片段如下:

```
1.  Page({
2.      goToDetail: function(e) {
3.          //获取携带的 data-id 数据
4.          let id = e.currentTarget.dataset.id;
5.          //携带新闻 ID 进行页面跳转
6.          wx.navigateTo({
7.              url: '../detail/detail?id=' + id
8.          })
9.      }
10. })
```

运行效果如图 15-23 所示。

15.5.5 清除临时数据

最后需要清除或注释掉一开始为了测试样式录入的临时数据,以免影响整体逻辑效果。



视频讲解

需要清除的数据如下。

- 首页(index.js): data 中的临时新闻列表数据(newsList);
- 新闻页(detail.js): data 中的临时新闻数据(article);
- 个人中心页(my. js): data 中的临时收藏夹新闻数据(newsList)、临时昵称(nickName)以及临时头像路径地址(src)。

此时带有模拟新闻数据的小程序前端项目“高校新闻网”就全部完成,后续章节将为其追加后端服务器和数据库的相关内容。



(a) “我的收藏”列表

(b) 浏览收藏的新闻

图 15-23 浏览已收藏新闻的效果

本实验到这里就全部完成了🎉，你完成的怎么样呢？要继续加油啊！😊