

基于 Vue.js 框架实现 移动终端数据可视化研究与实践

吴志霞, 叶根梅, 甘 丽, 周凌燕

摘 要: Vue.js 是一套构建可视化界面的渐进式框架, 非常容易与其他库或已有项目进行绑定和组合. 本研究应用 Vue.js 框架对校园管理系统移动端的新闻数据可视化模块进行二次开发, 实现了新闻数据在不同移动终端的可视化. 该研究对移动终端开发模型及实现方式进行了尝试和探索, 可为开展跨平台应用研究人员提供实践参考.

关键词: 移动终端; Vue.js; 可视化

中图分类号: TP39

文献标志码: A

文章编号: 1008-7974(2020)06-0062-05

DOI: 10.13877/j.cnki.cn22-1284.2020.06.011

随着数字化校园建设工作的快速推进, 以及移动网络终端的普及, 使得智能手机等移动终端成为校园信息发布和获取的最主要工具^[1], 应用程序的开发既要提供 PC 端也要提供移动端(包括不同操作系统平台的移动终端)成为一种趋势, 而移动终端的开发方案不外乎原生开发模式(Native App)、Web 开发模式(Web APP)以及混合开发模式(Hybrid App)当中的一种. 本研究以校园管理系统移动端的新闻数据可视化模块为实例, 采用 Web 开发模式, 运用 Vue.js 框架进行程序的二次开发, 实现数据的可视化. 文中详细描述了

实现步骤, 并阐述了 Vue 项目的前端组件化的设计思想.

1 开发实例及平台对外开放 API 接口介绍

1.1 开发实例介绍

开发实例是针对校园管理系统, 该系统后台服务器端采用 Spring+SpringMVC+MyBatis 框架技术实现, 对外提供的数据是 JSON 格式的数据, 但移动终端是采用原生态开发模式, 只提供了基于 Android 平台的移动客户端, 其中 Android 版的移动终端校园新闻列表如图 1 所示. 现针对校园管理系统移动终端的新闻

收稿日期: 2020-01-09

基金项目: 校级自然科学重点研究项目(2019xjzdky10); 安徽省自然科学基金项目(KJ2018A0905).

作者简介: 吴志霞, 女, 江西抚州人, 马鞍山师范高等专科学校副教授; 叶根梅, 甘丽, 周凌燕, 马鞍山师范高等专科学校(安徽马鞍山 243041).

数据可视化模块,采用Web开发模式,运用Vue.js框架进行程序的二次开发,实现数据在不同移动终端的可视化。



图1 Android版校园新闻列表

1.2 有关后台服务器端开放的JSON API接口介绍

校园管理系统后台服务器端对外提供能够返回JSON格式数据的API接口,开发者可以根据后台服务器对外提供的URL地址去获取JSON格式的数据。当前校园管理系统后台服务器的URL配置路径为:http://192.168.1.102:8088/ssmNews/,开发者可以根据配置路径去访问不同的数据资源。

(1)图片资源.如涉及访问图片,则使用服务器路径+字符串拼接方式获取,请求地址为:

http://192.168.1.102:8088/ssmNews/upload/article/images/2f/e2/8311c8105373.jpg

(2)数据资源.如要获取校园新闻,向后台服务器请求的地址为:

http://192.168.1.102:8088/ssmNews/client/news/getTopNews.action

请求如果成功,后台服务器返回JSON数据格式的数据,有关JSON数据的说明,如表1所示。

表1 获取新闻API

请求路径	http://192.168.1.102:8088/ssmNews/client/news/getTopNews.action	
HTTP请求方式	POST	
参数名	类型	描述
RESULT	varchar	成功S,失败F
DATA		
字段名	类型	描述
Id	int	id号
Title	varchar	标题
description	varchar	描述
Imgurl	varchar	图片
样例		
返回JSON	<pre>{ "DATA": [{ "id": 1, "title": "学习贯彻习近平总书记讲话精神", "description": "马鞍山师专学习贯彻习近平总书记在学校思想政治理论课教师座谈会上讲话的重要", "imgurl": "/upload/article/images/2f/e2/8311c8105373.jpg" }, { "id": 2, "title": "电动车校内上牌,服务到“家”", "description": "为加强校园电动车安全管理,落实《马鞍山市人民政府办公室关安交警开发区支队,组织开展了为期三天的校内电动车免费上牌服。", "imgurl": "/upload/article/images/2f/e2/8300c7104053.jpg" }], "RESULT": "S" }</pre>	

2 Vue.js 简介

Vue (读音/vju:/, 类似于 view) 是一套构建用户界面的渐进式 MVVM (Model-View-View-Model) 模式的 JavaScript 框架, 与其他重量级框架不同的是, Vue 采用自底向上增量开发的设计, 以数据驱动和组件化的思想构建^[2]. MVVM 衍生于 MVC 开发模式, 其本质是通过 View-Model 的数据绑定连接 View 和 Model, 让数据的更新自动映射为视图的变化^[3]. Vue.js 的 MVVM 架构图如图 2 所示^[4], 不同于原有的手动操作 DOM 节点方式, Vue.js 是响应式的, 是采用将 DOM 和数据绑定起来的方法, 而这种绑定是一种双向的绑定, 一旦创建了绑定, DOM 将会与数据保持同步, 数据一旦更新, 相应的 DOM 节点也会同步更新, 反之亦然.

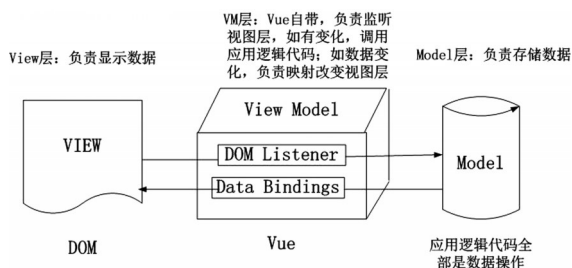


图 2 Vue.js 的 MVVM 架构图

3 采用 Vue.js 框架实现数据可视化

3.1 路由配置

编辑 src/router 下的 index.js 文件, 进行路由的设置, 明确访问新闻列显 (Home) 组件的访问路径. 有关路由配置文件 index.js 的核心配置代码如下:

```
import Vue from 'vue'
import Router from 'vue-router'
import Home from '@pages/home/Home'
Vue.use(Router)
export default new Router({
  routes: [
    {
      path: '/',
```

```
name: 'Home',
component: Home
}
])
})
```

3.2 父组件的定义

通过下方 Home 组件的定义, 可知 Home 组件使用了两个子组件, 分别是轮播图组件 (HomeSwiper) 与推荐新闻组件 (HomeRecommend). 通过父组件的定义, 明确前端组件化的设计思想, 每一个组件就是 Vue 的一个实例, 一个 Vue 项目由多个组件组成, 即一个 Vue 项目是由多个 Vue 实例拼装组合而成, Vue 实例自动结合模版及数据生成最终要展示的内容, 并将内容放到挂载点处. 有关父组件——Home 组件的核心代码如下:

```
<!--1. 模板的定义-->
<template>
  <div>
    <!--轮播图组件-->
    <HomeSwiper :swiperList="swiperList"></HomeSwiper>
    <!--推荐新闻组件-->
    <HomeRecommend :recommendList="recommendList"></HomeRecommend>
  </div>
</template>
<script>
//导入组件
import HomeSwiper from './components/Swiper'
import HomeRecommend from './components/Recommend'
import axios from 'axios'
//2.vue实例的定义(可提供数据、方法及事件的定义)
export default {
  name: 'Home',
  components: {
    HomeSwiper,
    HomeRecommend
  },
  data () {
    return {
      swiperList: [],
      recommendList: []
    }
  }
}
```

```

},
methods: {
  getHomeInfo () {
    //发送 axios 网络请求, 应答成功进入自定义的方法, 处理数据
    axios.get('/api/index.json')
      .then(this.getHomeInfoSucc)
  },
  getHomeInfoSucc (res) {
    //处理请求返回的数据
    res = res.data
    if (res.ret && res.data) {
      const data = res.data
      this.swiperList = data.swiperList
      this.recommendList = data.recommendList
      console.log(this.recommendList)
    }
    console.log(res)
  }
},
mounted () { //声明组件加载时调用的方法
  this.getHomeInfo()
}
}
</script>
<!--3.样式的定义-->
<style>
</style>

```

3.3 父组件向子组件传递内容

通过上方 Home.vue 与下方 Recommend.vue 的定义, 明确父组件向子组件传递内容是通过属性的形式给予组件传递值, 子组件接收内容再在模板上显示. 有关子组件——Recommend 组件的核心代码如下:

```

<template>
<div>
  <div class="recommend-title">推荐信息</div>
  <ul>
    <router-link tag="li" class="item border-bottom"
      v-for="item of recommendList"
      :key="item.id"
      :to="/detail/" + item.id">
      
      <div class="item-info">
        <p class="item-title">{{item.title}}</p>

```

```

    <p class="item-desc">{{item.desc}}</p>
    <button class="item-button">查看详情</button>
  </div>
</router-link>
</ul>
</div>
</template>
<script>
export default {
  name: 'HomeRecommend',
  props: {
    recommendList: Array
  }
}
</script>
<style lang="stylus">
  //样式文件的编写
</style>

```

3.4 运行效果

基于 Vue.js 校园新闻列表移动终端的运行效果如图 3 所示, 将图 3 与图 1 进行比对, 可以观察出采用 Web 开发模式, 运用 Vue.js 框架实现的新闻列表移动端, 能达到原生 Android 的开发效果, 可视化体验效果更佳, 开发效率也比较快捷。



图3 Vue.js 版校园新闻列表

4 结束语

本研究成功地运用 Vue.js 主流移动框架实现了校园管理系统移动终端的新闻数据可视化模块,可避免不同系统同一应用反复开发的人力和物力的浪费,为开展跨平台应用研究的人员提供了一种实践思路.

参考文献:

- [1] 吴志霞.XML 在 Android 与 Web 双模式教学平台中数据交换的研究与应用[J].通化师范学院学报, 2017(4):53-56.
- [2] Vue.js 官网[EB/OL].<https://cn.vuejs.org/>.
- [3] 易剑波.基于 MVVM 模式的 WEB 前端框架的研究[J].信息与电脑,2016(19):76-77.
- [4] 刘翔宇.基于 Vue 的数据可视化系统的设计与实现[D].北京:北京邮电大学,2018.

(责任编辑:王前)

Research and Practice of Mobile Terminal Data Visualization Based on Vue.js

WU Zhi-xia, YE Gen-mei, GAN Li, ZHOU Ling-yan

(Maanshan Teacher's College, Ma'anshan 243041, China)

Abstract: Vue.js, which is very easy to bind and combine with other libraries or existing projects, is a progressive framework for building visual interfaces. In this study, Vue.js framework is applied to the secondary development of news data visualization module in the mobile terminal of campus management system, which realizes the visualization of news data in different mobile terminals. This research explores the development model and implementation mode of mobile terminal, and provides practical reference for cross platform application researchers.

Keywords: android; Vue.js; visualization