

# 基于 Vue+Element 的第四课堂学分管理系统学生端的设计与实现

潘虹霖<sup>1</sup> 赵春<sup>2</sup>

四川大学锦城学院 计算机与软件学院 四川 成都 611731

DOI: 10.18686/jsjxt.v1i3.1266

**【摘要】**第四课堂学分管理系统的设计与实现主要是为了解决传统管理方式的缺陷,降低认定学生第四课堂学分的工作量,以及更高效、更系统地完成学分认定工作。本文针对使用 Vue+Element 技术框架对第四课堂学分管理系统学生端的前端页面设计,以及前端从后台请求数据的实现进行阐述。

**【关键词】**学分管理系统 VueElement 前端请求数据

## 1 引言

Vue 是近年来最流行的前端框架之一,由于它具有轻量、简单、接口友好、学习成本低、开发效率高特点,越来越多的开发人员开始使用 Vue 作为前端开发的技术框架;Element 作为一套基于 Vue2.0 的桌面端组件库,一般与 Vue 结合作为开发前端页面的技术框架。第四课堂学分管理系统正是基于这种 Vue+Element 前端开发框架而实现其前端功能的,总体分为教师端和学生端,下面就本系统的学生端页面的设计与实现展开介绍。

## 2 系统学生端的需求分析

(1)学分管理:学生点击“我的活动”页面。在这个页面中,默认显示已获得学分列表,点击切换按钮切换到未获得学分列表。在学分列表上具有筛选列表(学分类型、学分个数);点击学分系统弹出该学分的详细信息(参与活动信息、认证文件信息、认证时间)。

(2)在线活动报名:学生在系统中点击“学院活动”页面,在页面中的活动清单中点击“详情”,就会在前端弹出活动信息框(学生不能编辑活动信息框中的信息);学生在弹出框的下方,点击“报名”,即立刻进行报名该活动;系统将根据学生的信息自动生成一个报名记录填写在报名表中。报名按钮此时变为“退出”。

(3)站内通知:学生在首页下方展示该学生从入学到毕业时获取学分的时间轴;认证学分:学生点击第四课堂“学院活动”。在活动列表中,选择想要参与的活动,点击“详情”,在弹出框中点击“上传认证

材料”;在文件选择框中选择系统要求格式的认证材料(要求格式为 doc、docx、xls、xlsx、zip)。如果学生需要重新上传认证材料,直接上传即可,将覆盖系统中的旧文件。

(4)文件下载:学生在系统首页点击“下载历史认证材料”,系统将其已通过学分的历史认证数据打包成一个压缩包,并通过浏览器让学生下载。

## 3 系统学生端的功能设计

### 3.1 总体设计

按照上述的系统需求分析,系统学生端页面分为 5 大模块,由此可编写 5 个独立的 vue 档。vue 是一个自定义的文件类型,用类似于 HTML 的语法描述一个 vue 文件。每个 vue 档包含 3 种类型的顶级语言块<template>、<script>、<style>,这 3 个部分分别代表了 HTML、JavaScript、css 的功能和地位。

### 3.2 主要功能模块

主要的模块可通过下表 1 中的 5 个 vue 文件了解,具体如下:

文件名	功能描述
department. vue	学院活动页面
detail. vue	活动详情页面
myActivity. vue	我的活动页面
upload. vue	文件上传页面
myCredits. vue	我的学分页面

在学院活动页面中,学生使用者可以获得活动名称、活动报名时间、活动时间、活动描述、学分个数、活动人数的信息;在活动详情页面中,主要是所

选择的这一活动的具体活动信息,包括活动负责人、负责人联系方式、活动所属部门、学分类型、学分个数、报名时间段、活动时间段、提交数据时间段、活动具体描述以及认证要求;在我的活动页面中,主要包括活动日期、学分名称、相关活动、认证状态、认证批次以及下载认证数据和上传认证数据两个操作;在文件上传页面中,主要实现学生选取文件、上传文件的功能。

我的学分模块包含一个 vue 文件: myCredits.vue, 主要实现学分状态(认证通过、认证未通过、认证未开始、认证中), 展示学生已获取到的学分信息, 包括活动日期、学分名称、相关活动、认证状态、认证批次, 以及实现下载认证通知的操作。

### 3.3 主要接口

系统主要以 API 接口的形式连接前后端, 主要用到的接口如下:

- (1) 查询活动: /api/activities/find;
- (2) 查看活动详情: /api/activities/detail/{id};
- (3) 学生查询学分: /api/credits/student/list;
- (4) 学生下载已经获得学分的通知文件: /api/file/student/downloadAll;
- (5) 上传多个文件: /api/file/multiUpload;
- (6) 以压缩包的形式下载多个文件: /api/file/downloadMore;

## 4 系统学生端的实现

### 4.1 系统关键组件选择

为了使本学管理系统的学生端页面的操作流程更简单, 同时满足开发成本低、开发效率高、系统性能好等要求, 主要选择 Vue.js、vue-router、axios、Element、components 等 Vue 这套渐进式框架的关键组件对前端进行开发。

Vue.js 是一套构建用户接口的渐进式框架, 只关注视图层并且采用自底向上增量开发的设计, 同时它是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件的, 而且与第三方库或既有项目整合, 主要解决前端资料绑定的问题。传统的前端开发, 主要基于 jQuery 通过各种复杂的选择器来操作 DOM。<sup>[1]</sup> 另外还具有双向绑定特性, 高效地将数据反映到页面模型上, 同时 Vue.js 更高效地处理页面 DOM 操作, 提升后台应用的性能。

vue-router 是 Vue.js 官方的路由管理器, 它一般与 Vue.js 的核心深度集成, 从而简化构建页面应用; 与传统的超链接页面跳转不同的是, 它是动态

调整页面路径从而实现页面切换, 这提升了用户的体验度。

axios 相当于一个 node.js 的 HTTP 客户端, 主要实现用同步的代码形式来实现异步的请求操作, 方便前端向后台请求数据时的中间操作——权限校验。

Element 和 components 是一组页面应用的组件库, 具有简洁、轻便、可移植的优点, 当构建组件时, 直接从官方组件库里直接获取到所需要的组件的源代码即可, 这样可以减少工作量, 提高开发效率。

总的来说, 整个系统的前端主要采用 vue.js 作为基础视图层框架, 使用 vue-router 实现页面路由的跳转功能, 并且将 Element 和 components 作为接口的基础组件, 进行权限校验后, 从后台请求到所需要的数据。

### 4.2 页面实现

(1) 信息展示: 在学院活动页面中, 可以创建一个多选框窗体来展示上述的主要功能模块与实现相应功能, 这里使用到了 Element 中 el-table-column 组件, 将其 type 属性设置为 selection 即可实现多选功能。

(2) 上传文件: 采用 <el-upload> 组件实现上传操作的前端入口, <el-button> 组件实现学生上传与下载文件的前端操作。<el-button> 是一个按钮组件, 在页面中被点击时会触发一个函数, 实现其相应的函数功能。核心代码实现如下: <el-button size="small" type="primary">上传认证材料 </el-button>。限制上传文件的格式、大小等, 则需要在 <el-upload> 中增添如下代码: multiple: accept="doc, docx, xls, xlsx, zip": http-request="uploadFile": auto-upload="true": limit="10"。multiple 表示支持多选文件, accept 表示接受上传的文件类型, http-request="uploadFile" 覆盖默认的上传行为为 uploadFile, auto-upload="true" 表示在选取文件后立即进行上传, limit 设置最大允许上传文件个数。

(3) 报名、退出与查看详情: 采用 <el-button> 组件实现学生对活动的报名与退出的前端操作。核心代码实现如下: <el-button type="text" size="small" @click="signUp(row, activityId)">报名 </el-button>, 将其 type 值设置为文本, 并带有一个 click(点击)属性 signUp() 函数把 activityId 作

为参数,实现学生对某一活动的报名;同样地,退出活动、查看详情功能的实现同理,将 click 属性中的函数分别写为 signOut() 和 detail(),再编写相应函数即可。

### 4.3 页面路由

(1)配置 router 文件,这里以学院活动为例,核心代码实现如下:

导入学院活动页:import student\_activity\_department from '@/views/student/activity/department';启用路由:Vue.use(Router)。

(2)在 Vue 的实例中导入路由,这里以学院活动为例,核心代码实现如下:

```
export default new Router({ routes: [{ .....
children: [{
  path: '/student/activity/department',
  name: '学院活动', component: student_activity_department, meta: {requireAuth: true} }]}] })
```

### 4.4 权限校验

由于第四课堂学分管理系统的用户分为教师和学生,所以在设计时需要请求数据的对象进行权限管理和校验。系统通过 axios 来实现此过程,以执行 POST 请求为例,核心代码如下:

```
const post = (url, params) => { return axios({method: 'post', url: url, .....}).then(res => res.data); }
```

### 4.5 请求数据

前端在向后台请求数据时,直接调用接口,进行权限校验,再在加载页面渲染数据即可将后台数据展示在前端页面,大致工作机制流程图如下:

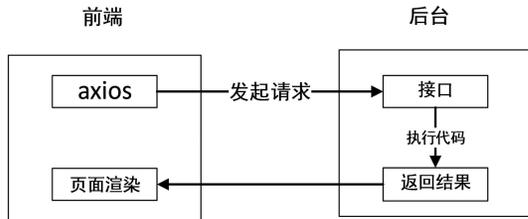


图 1 学生端请求数据工作机制流程

## 【参考文献】

[1]旷志光,纪婷婷,吴小丽.基于 Vue.js 的后台单页应用管理系统的研究与实现[J].现代计算机(专业版),2017(30):51-55.

以请求活动资料为例,具体代码实现如下:

(1)在 studentMG.js 文件中导出请求学院活动信息:

```
export const studentActivity = (param) => {return req('post', "/api/activities/find", param)}
```

(2)调用学生页面的相关接口:

```
import studentActivity from "../../api/studentMG";
```

(3)向后台请求活动资料:

```
studentActivity ( this.studentActivityFindForm).then(res => {
  res.data.records.forEach (item => {
    let tempItem = {}//定义一个 Item 变量存储窗体的列信息
    tempItem [' activityName '] = item. activityName//将活动名称赋值给一个 Item))})
```

这里需要在前面的 <el-table> 标签中添加上: data="departmentActivityData",用于将这个窗体与学院活动数据进行绑定,从而使活动页面在加载时就能够自动将活动数据渲染到页面的窗体中。

## 5 结束语

作为一种当前流行的前端开发框架,Vue+Element 实现了第四课堂学分管理系统的页面设计与实现,包括页面视图的构建、组件的运用部署以及页面从后台请求数据的操作,其实质是利用 Vue.js 对前端页面应用的实现,对相关组件的封装,并配置相关文件实现前后端联动。这种架构在一定程度上减少了前后端的代码量,减少了开发人员的工作量,有效地提升了系统的稳定性和健壮性。