福州大学至诚学院
FUZHOU UNIVERSITY ZHICHENG COLLEGE

前端框架应用

11. 大学生志愿服务项目开发

01 技术方案

02 工程化项目搭建

03 项目需求分析

04 项目功能开发

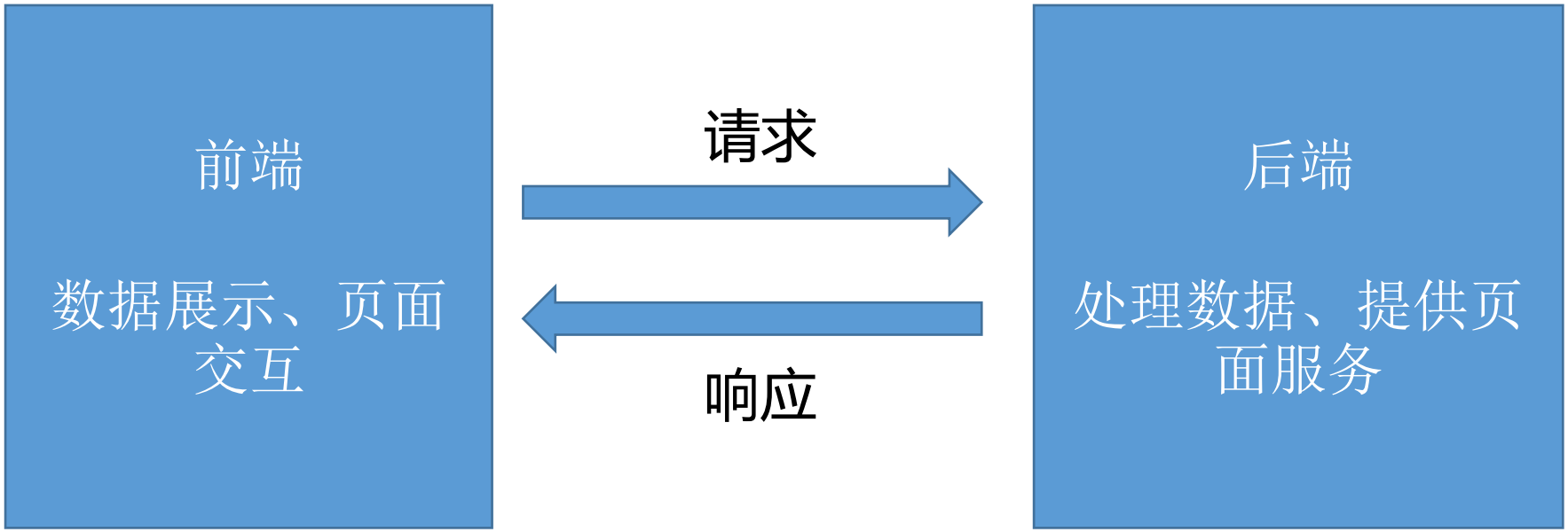
Part.1

技术方案

11.1 技术方案

本项目采用**前后端分离**的开发模式，前端专注于数据展示和页面交互，后端则专注于业务数据处理逻辑，前后端通过**API接口**实现数据的交互。

前端在开发过程中使用数据mock，**模拟后端的接口服务**，完成页面的开发。



11.1 技术方案

技术选型：

- **使用Vue3版本前端框架。**
- **使用vite工具搭建项目。**
- **使用基于Vue 3的Vant组件库开发页面。**
- **使用vue-router实现单页面开发。**
- **使用axios完成接口请求。**
- **使用mock提供接口数据服务。**

Part.2

工程化项目搭建

11.2 工程化项目搭建

- 创建一个项目名称为 student-vol 的项目，输入如下命令：

npm create vite@latest student-vol -- --template vue

```
> npx
> create-vite student-vol --template vue

│
◇  Scaffolding project in L:\Vue前端框架应用
│
└  Done. Now run:

  cd student-vol
  npm install
  npm run dev
```

11.2 工程化项目搭建

- **安装依赖库vant4和axios：**

```
cd student-vol

npm install vant --save

npm install axios --save

npm install vue-router@4 --save

npm install moment --save

npm install

npm run dev
```

11.2 工程化项目搭建

- **项目目录：**

  为了方便后续的开发，对项目的主要目录结构进行说明：

  public：存放公共文件

  src/assets：存放资源文件，如图片

  src/components：存放自定义组件

  src/router/index.js：存放路由配置文件

  src/views：存放页面级组件

  src/App.vue：存放项目的主组件

  src/main.js：存放项目的入口文件

11.2 工程化项目搭建

- 引入项目所需要的组件和样式文件，并在Vue.js实例上注册

src/main.js

```
import { Button, List, Tag, Field, CellGroup,Uploader,Popup, Picker,DatePicker } from 'vant';
import 'vant/lib/index.css';
const app = createApp(App)
app.use(Button)
    .use(List)
    .use(Tag)
    .use(Field)
    .use(CellGroup)
    .use(Uploader)
    .use(Popup)
    .use(Picker)
    .use(DatePicker);
app.mount('#app')
```

11.2 工程化项目搭建

- Axios是一个基于 promise 的网络请求库，用于在浏览器中向服务器发送 AJAX请求。

src/App.vue

```
<script>
import axios from 'axios';

export default {
  mounted(){
    axios.get('/user')
      .then(function (res){
        console.log(res);
      });
```

11.2 工程化项目搭建

src/App.vue

```
    axios.post('/addUser', {name:'张三', age:18})
      .then(function (res){
        console.log(res.data);
      });
  }
}
</script>
```

代码发送了两种请求(get、post），并在then方法中处理服务器返回的响应。

11.2 工程化项目搭建

- Mock 数据模拟后台数据，使得该项目可以独立于后端进行开发。通过项目中的vue.config.js可以实现Mock数据。

(1) 在项目根目录下创建mock/data.json:

```
{
    "error": 0,
    "data": [{
        "id":1,
        "name":"张三",
        "age":18
    },{
        "id":2,
        "name":"李四",
        "age":19
    }],
    "msg":"success"
}
```

11.2 工程化项目搭建

(2) 更新根目录下的 vite.config.js 文件：

```
import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'
import fs from 'fs'
import path from 'path’
```

11.2 工程化项目搭建

(2) 更新根目录下的 vite.config.js 文件：

```
function mockPlugin() {
  return {
    name: 'simple-mock-plugin',
    configureServer(server) {
      server.middlewares.use((req, res, next) => {
        const sendJSON = (file) => {
          const filePath = path.resolve(process.cwd(), `mock/${file}`)
          if (!fs.existsSync(filePath)) {
            res.statusCode = 404
            res.setHeader('Content-Type', 'application/json')
            return res.end(JSON.stringify({ error: `Mock file not found:
${file}` }))
          }
          const json = fs.readFileSync(filePath, 'utf-8')
          res.setHeader('Content-Type', 'application/json')
          res.end(json)
        }
```

11.2 工程化项目搭建

(2) 更新根目录下的 vite.config.js 文件：

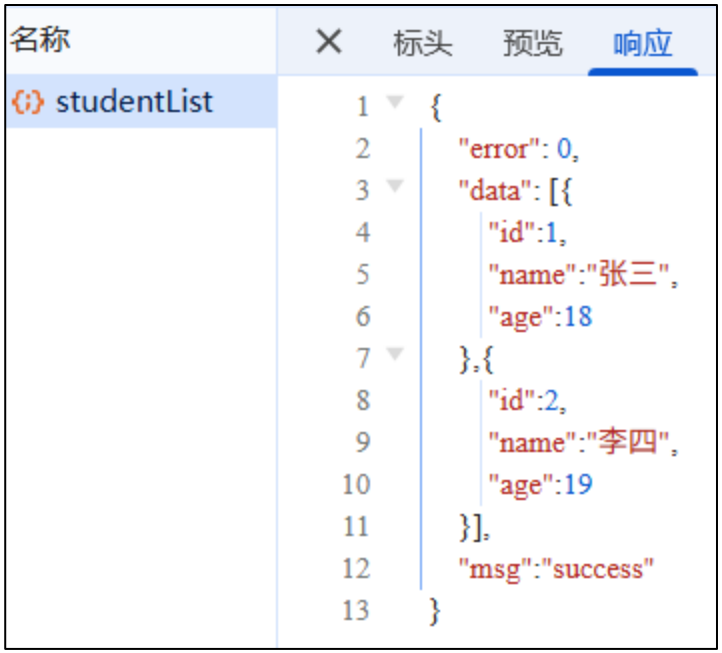
```
if (req.url === '/api/studentList' && req.method === 'GET') {
        return sendJSON('data.json')
      }



      next()
    })
  },
 }
}
```

11.2 工程化项目搭建

(3) 更新App.vue:

```
export default {
  mounted(){
    axios.get('/api/studentList')
      .then(function (res){
        console.log(res.data.data);
      });
  }
}
```

在浏览器中访问http://localhost:5173/，按F12打开“开发者工具”，切换到Network菜单下，src/App.vue文件发送的get请求得到的Mock数据如图

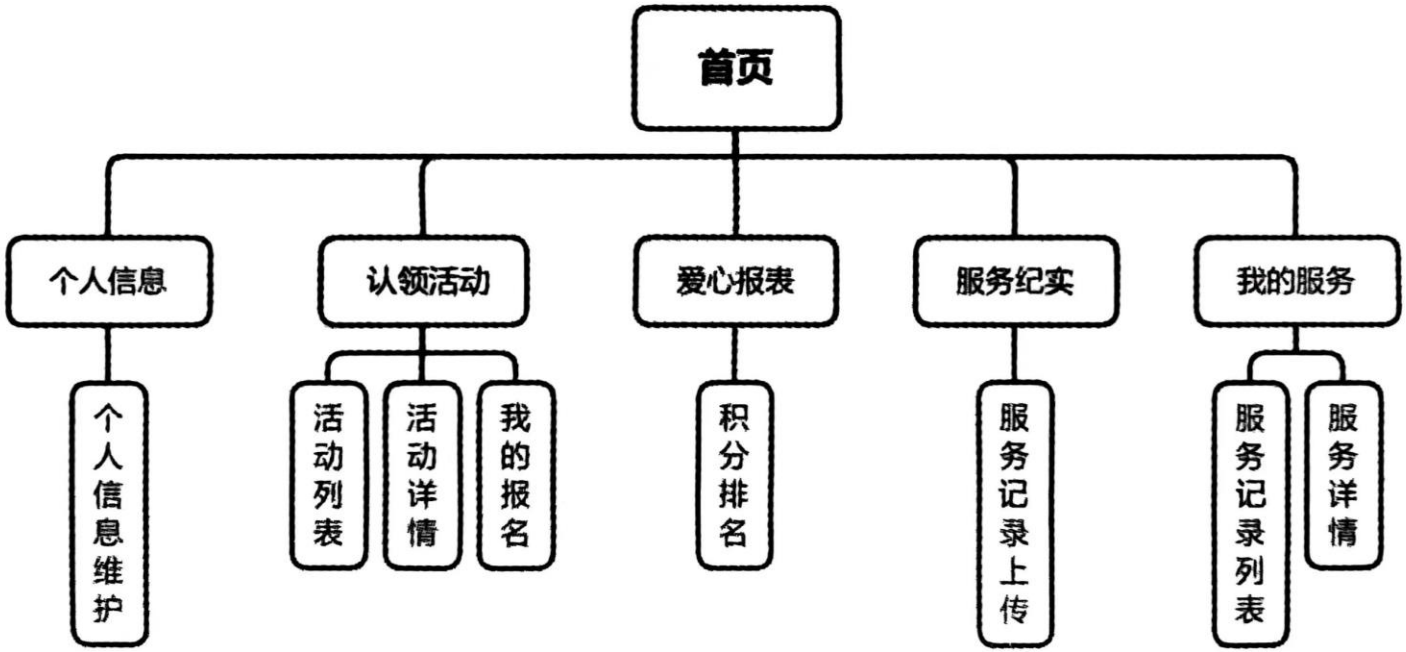


Part.3

项目功能展示

10.3 项目功能展示

本项目创建的是**移动端的HTML**页面，用户进入首页后，根据首页提供的入口，进入**个人信息、认领活动、爱心报表、服务纪实、我的服务**等页面。



10.3 项目功能展示

一、首页

10.3 项目功能展示

二、个人信息维护

10.3 项目功能展示

三、积分排名

10.3 项目功能展示

四、活动列表

10.3 项目功能展示

五、活动详情

10.3 项目功能展示

六、我的报名

10.3 项目功能展示

七、服务纪录上传

10.3 项目功能展示

八、服务纪录列表

10.3 项目功能展示

九、服务详情

Part.4

项目功能开发

10.4 项目功能开发

1. 页面入口

App.vue为页面入口文件，定义了项目的根组件。

本项目是一个单页面应用，当访问不同的页面时，并不会重新加载整个页面，而是更新根组件中展示的内容，也就是在根组件中切换要展示的子组件。

所以需要在App.vue文件中插入路由子组件，并且配置相关路由。

10.4 项目功能开发

1. 页面入口

路由管理文件 src/router/index.js，用户根据项目需要添加路由配置即可。初始内容为：

```
import { createRouter, createWebHistory } from 'vue-router'

const routes = [
]

const router = createRouter({
  history: createWebHistory(),
  routes
})

export default router
```

10.4 项目功能开发

1. 页面入口

App.vue文件代码为：

```
<template>
  <router-view/>
</template>
```

main.js文件中引入路由配置：

```
import router from './router'
```

```
.use(DatePicker)
.use(router);
```

10.4 项目功能开发

2. 首页

首页由顶部的**个人信息**、中间的**功能页面**入口、底部的**活动列表**构成。

在src文件夹内的views子文件夹下创建HomeView.vue。

在src文件夹内的 components 子文件夹创建文件夹Home，并在其下创建三个子组件：User.vue、 Navs.vue和ActItem.vue，分别用来表示个人信息、功能页面以及活动。

10.4 项目功能开发

2. 首页

● 个人信息User.vue

```
<template>
  <div class="header-inner">
    <div class="userInfo">
      <span class="avatar">
        <!-- 用户头像 -->
        <img :src="user.avatar" />
      </span>
      <span>
        <!-- 用户姓名 -->
        {{ user.name }}
      </span>
    </div>
```

```
  <!-- 跳转用户信息页面，先用/占位 -->
    <router-link to="/">
      <button class="rArrow"></button>
    </router-link>
  </div>
</template>
<script>
export default {
  props: ["user"]
}
</script>
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
<script>
import axios from 'axios';
import User from '../components/Home/User.vue';
export default {
    components: {
        User,
    },
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
<template>
  <div class="home">
    <div class="header">
      <h1>大学生志愿者活动</h1>
      <!-- 用户信息 -->
      <User :user="userInfo" />
    </div>
  </div>
</template>
```

定义属性：userInfo，存储用户信息

定义方法：通过axios获取用户信息

初始化时：调用方法得到用户信息

10.4 项目功能开发

2. 首页

- HomeView.vue

```
data() {
    return {
        userInfo: {
            avatar: "",
            name: ""
        },
    }
},
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
methods: {
        fetchUserInfo() {
            const that = this;
            axios.get("/api/userInfo")
                .then(function (response) {
                    const { error, data = {} } = response.data;
                    if (error === 0) {
                        that.userInfo = {
                            avatar: data.avatar,
                            name: data.name
                        }
                    }
                })
        }
    },
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
mounted() {
        this.fetchUserInfo();
    }
}
</script>
```

- vite.config.js

```
if (req.url === '/api/userInfo' && req.method === 'GET') {

        return sendJSON('userInfo.json')

}
```

10.4 项目功能开发

2. 首页

- router/index.js

```
import HomeView from '../views/HomeView.vue'
const routes = [
    {
        path: '/',
        name: 'home',
        component: HomeView
    },
]
```

10.4 项目功能开发

2. 首页

- **userInfo.json**


```
{
    "error": 0,
    "data": {
        "avatar": "./imgs/avatar.png",
        "name": "李华",
        "code": 12343434,
        "gender": 0,
        "phoneNum": "18787698789",
        "school": "福州大学至诚学院",
        "profession": "软件工程",
        "userType": 0
    }
}
```

10.4 项目功能开发

2. 首页

- 功能页面Navs.vue

```
<template>
    <ul class="navs">
        <li>
        <router-link to="/loveActs">
            <img src="../../assets/p1.png" />
            <p>认事</p>
        </router-link>
        </li>
        <li>
        <router-link to="/loveReport">
            <img src="../../assets/p2.png" />
            <p>爱心报表</p>
        </router-link>
        </li>
```

10.4 项目功能开发

2. 首页

● 功能页面Navs.vue

```
    <li>
        <router-link to="/UploadService">
            <img src="../../assets/p3.png" />
            <p>服务纪实</p>
        </router-link>
        </li>
        <li>
        <router-link to="/serviceRecords">
            <img src="../../assets/p5.png" />
            <p>我的服务</p>
        </router-link>
        </li>
    </ul>
</template>
```

```
<script>
export default {
}
</script>
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
import Navs from '../components/Home/Navs.vue';

export default {
    components: {
        User,
        Navs
    },
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
<div class="home">
    <div class="header">
        <h1>大学生志愿者活动</h1>
        <!-- 用户信息 -->
        <User :user="userInfo" />
    </div>
  </div>
  <!-- 功能页面入口组件 -->
  <Navs />
```

10.4 项目功能开发

2. 首页

● 活动子组件ActItem.vue

- 该页面活动的信息包括：

id(隐藏)、名称、开始时间、结束时间、活动地点、是否校内活动、活动状态(进行中/已结束)、图片，可通过 props 传递。

- 单击活动可跳转至活动详情页面，可由父组件响应。

10.4 项目功能开发

2. 首页

- 活动子组件ActItem.vue

```
<template>
  <div class="actItem" @click="detailsClick(data.id)">
    <div class="actItem-pic">
      <!-- 活动图片 -->
      <img :src="data.actPic"/>
      <!-- 根据状态决定样式 -->
      <span class="actItemtatus" :style="getTagStyle(data)">
        <!-- 根据状态展示标签 -->
        {{ getTagTxt(data) }}
      </span>
    </div>
```

10.4 项目功能开发

2. 首页

● 活动子组件ActItem.vue

```
<div class="actItem-body">
        <p class="actItem-title">
            {{ data.title }} <!-- 标题 -->
        </p>
        <p> <!-- 开始时间至结束时间 -->
            {{ formatDate(data.startTime) }}
            至
            {{ formatDate(data.endTime) }}
        </p>
        <p class="actItem-tag"> <!-- 活动地点 -->
            {{ data.isYourSchool ? "本人所在学校活动" : data.palce }}
        </p>
    </div>
  </div>
</template>
```

10.4 项目功能开发

2. 首页

● 活动子组件ActItem.vue

```
<div class="actItem-body">
        <p class="actItem-title">
            {{ data.title }} <!-- 标题 -->
        </p>
        <p> <!-- 开始时间至结束时间 -->
            {{ formatDate(data.startTime) }}
            至
            {{ formatDate(data.endTime) }}
        </p>
        <p class="actItem-tag"> <!-- 活动地点 -->
            {{ data.isYourSchool ? "本人所在学校活动" : data.palce }}
        </p>
    </div>
  </div>
</template>
```

10.4 项目功能开发

2. 首页

- 活动子组件ActItem.vue

```
<script>
import moment from "moment";
export default {
    props: ["data"],
    methods: {
        getTagStyle(data) {
            return { backgroundColor: data.canApply ? "#07c160" : "};
        },
        formatDate(value) {
            return value ? moment(value).format("YYYY-MM-DD hh:mm") : "";
        },
```

10.4 项目功能开发

2. 首页

- 活动子组件ActItem.vue

```
        detailsClick(id) {
            this.$emit("goDetails", id);
        },
        getTagTxt(data) {
            return data.canApply ? "进行中" : "已结束";
        }
    }
}
</script>
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
import ActItem from '../components/Home/ActItem.vue';

export default {
    components: {
        User,
        Navs,
        ActItem
    },
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
<div class="list">
    <div class="list-header">
        <span>志愿活动</span>
        <div class="list-more" @click="goMoreActs">
            查看更多
            <img src="../assets/rArrow.png" />
        </div>
    </div>
    <div class="actList">
        <!-- 推荐列表组件 -->
        <ActItem v-for="item in list" :key="item.id" :data="item"
@goDetails="goDetails"/>
    </div>
  </div>
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
data() {
    return {
        userInfo: {
            avatar: "",
            name: ""
        },
        list: []
    }
},
```

定义活动列表：list

定义方法：通过axios获取活动列表

初始化时：调用方法得到活动列表信息

10.4 项目功能开发

2. 首页

- HomeView.vue

```
fetchActList() {
        const that = this;

        axios.get("/api/actList")
            .then(function (response) {
                const { error, data = {} } = response.data;

                if (error === 0) {
                    that.list = data.list;
                }
            })
},
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
//根据id跳转至活动详情页面
goDetails(id) {
        this.$router.push('/actDetails?id=' + id);
},
/查看更多
goMoreActs() {
        this.$router.push("/loveActs");
},
```

10.4 项目功能开发

2. 首页

- HomeView.vue

```
mounted() {
        this.fetchUserInfo();
        this.fetchActList();
}
```

- vite.config.js

```
if (req.url === '/api/userInfo' && req.method === 'GET') {

        return sendJSON('userInfo.json')

}
```

10.4 项目功能开发

2. 首页

- actList.json


```
{
    "error": 0,
    "data": {
        "current": 1,
        "pageSize": 10,
        "pageCount": 1,
        "list": [
```

```
        {
            "id": 0,
            "title": "运动会志愿者招募",
            "startTime": 1671494400000,
            "endTime": 1671697800000,
            "palce": "福州大学至诚学院",
            "isYourSchool": true,
            "canApply": true,
            "actPic": "./imgs/actImg.jpeg"
        },
```

10.4 项目功能开发

2. 首页

10.4 项目功能开发

3. 个人信息维护

- 个人信息页面由表单、按钮构成。
- 该页面加载后，向后端发送 get 请求，获取个人信息；
- 用户更新个人信息后，通过“提交”按钮向后端发送 post 请求，提交更新后的个人信息。

10.4 项目功能开发

3. 个人信息维护

在 src 文件夹内的 views 子文件下创建 UserCenter.vue

```
<template>
  <div class="container">
    <div class="inner">
      <h1>个人信息</h1>
      <ul>
        <li>
          <p class="label"><span>*</span>姓名</p>
          <span class="txt">{{ name }}</span>
        </li>
        <li>
          <p class="label"><span>*</span>编号</p>
          <span class="txt">{{ code }}</span>
        </li>
```

10.4 项目功能开发

3. 个人信息维护

```
<li>
  <p class="label"><span>*</span>性别</p>
  <input type="radio" value="0" v-model="gender" />男
  <input type="radio" value="1" v-model="gender" />女
</li>
<li>
  <p class="label"><span>*</span>手机号</p>
  <input placeholder="请输入手机号" v-model.trim="phoneNum" />
</li>
<li>
  <p class="label"><span>*</span>学校</p>
  <input placeholder="请输入学校名称" v-model.trim="school" />
</li>
<li>
  <p class="label"><span>*</span>专业</p>
  <input placeholder="请输入专业名称" v-model.trim="profession" />
</li>
```

10.4 项目功能开发

3. 个人信息维护

```
      <li>
        <p class="label"><span>*</span>人员属性</p>
        <div class="userTypeList">
          <span :class="{ active: userType === 0 }" @click="changeUserType(0)">群众</span>
          <span :class="{ active: userType === 1 }" @click="changeUserType(1)">团员</span>
          <span :class="{ active: userType === 2 }" @click="changeUserType(2)">党员</span>
        </div>
      </li>
    </ul>
    <button class="subBtn" @click="submitClick">提交</button>
  </div>
</div>
</template>
```

10.4 项目功能开发

3. 个人信息维护

```
<script>
import axios from 'axios';
import { showToast, showSuccessToast, showFailToast} from 'vant';
export default {
  data() {
    return {
      name: '',
      code: null,
      gender: 0,
      phoneNum: '',
      school: '',
      profession: '',
      userType: 0
    }
  },
```

10.4 项目功能开发

3. 个人信息维护

```
methods: {
  changeUserType(value) {
    this.userType = value;
  },
```

```
fetchUsrInfo() {
      const that = this;
      axios.get('/api/userInfo')
        .then(function (response) {
          const { error, data = {} } = response.data;
          if (error === 0) {
            that.name = data.name;
            that.code = data.code;
            that.gender = data.gender;
            that.phoneNum = data.phoneNum;
            that.school = data.school;
            that.profession = data.profession;
            that.userType = data.userType;
          }
        })
    },
```

10.4 项目功能开发

3. 个人信息维护

```
check() {
  var phoneNum = this.phoneNum;
  var school = this.school;
  var profession = this.profession;
  if (phoneNum === '') {
    showToast("手机号不可为空");
    return;
  }
  if (school === '') {
    showToast("学校不可为空");
    return;
  }
  if (profession === '') {
    showToast("专业不可为空");
    return;
  }
  const payload = {
    name: this.name,
    code: this.code,
    gender: this.gender,
    phoneNum,
    school,
    profession,
    userType: this.userType
  };

  return payload;
},
```

10.4 项目功能开发

3. 个人信息维护

```
submitClick() {
  // 1. 检查是否有输入为空
  var payload = this.check();
  if (!payload) {
    return;
  }
```

```
// 2. 提交修改后的数据
    axios.post('/api/userInfo/submit')
      .then(function (response) {
        const { error, msg } = response.data;
        if (error === 0) {
          showSuccessToast("操作成功");
        } else {
          showFailToast(msg || '网络错误，请稍后重试');
        }
      })
  }
```

10.4 项目功能开发

3. 个人信息维护

```
  mounted() {
    this.fetchUsrInfo();
  }
}
</script>
```

10.4 项目功能开发

3. 个人信息维护

- vite.config.js

增加一条响应

```
if (req.url === '/api/userInfo/submit' && req.method === 'POST') {
        return sendJSON('success.json')
}
```

10.4 项目功能开发

3. 个人信息维护

- router/index.js

增加一条路由

```
import UserCenter from '../views/UserCenter.vue’
…
    {
        path: '/userCenter',
        name: 'userCenter',
        component: UserCenter
    },
```

10.4 项目功能开发

3. 个人信息维护

10.4 项目功能开发

4. 积分排名

- 积分排名由**顶部的切换按钮、中间的积分和排名总览、下方的服务标兵**排行榜构成。
- 根据当前选中的积分类型（总积分、年度积分）向后端发送 get 请求，获取用户的积分和排名，以及前10名的服务标兵列表。

10.4 项目功能开发

4. 积分排名

在src 文件夹内的views子文件夹创建 LoveReport.vue

```
<template>
  <div class="container">
    <div class="tabs"> <!--积分类型切换-->
      <div :class="{ tabItem: true, active: curTab === 0 }" @click="tabClick(0)">
        总积分
      </div>
      <div :class="{ tabItem: true, active: curTab === 1 }" @click="tabClick(1)">
        年度积分
      </div>
    </div>
```

10.4 项目功能开发

4. 积分排名

```
<div class="dataList"> <!--积分、排名总览-->
    <div class="dataItem">
        <p class="dataItem-score">{{ provinceRank }}</p>
        <p class="dataItem-label">全省的排名</p>
    </div>
    <div class="dataItem">
        <p class="dataItem-score">{{ totalScore }}</p>
        <p class="dataItem-label">服务总积分</p>
    </div>
    <div class="dataItem">
        <p class="dataItem-score">{{ grandeRank }}</p>
        <p class="dataItem-label">本校本年级排名</p>
    </div>
</div>
```

10.4 项目功能开发

4. 积分排名

```
<div class="rank">
  <div class="rank-hd">
      <p class="rank-title">本校本年级服务标兵</p>
      <p class="rank-range">1200人参与排名</p>
  </div>
```

10.4 项目功能开发

4. 积分排名

```
        <ul class="rank-list">
            <li v-for="(item, idx) in dataList" :key="item.id">
                <div class="rank-student">
                    <span class="rank-idx">{{ idx + 1 }}</span>
                    <span class="rank-avatar">
                        <img :src='item.avatar' />
                    </span>
                    <span class="rank-name">{{ item.name }}</span>
                </div>
                <div class="rank-score">{{ item.score }}分</div>
            </li>
        </ul>
    </div>
  </div>
</template>
```

10.4 项目功能开发

4. 积分排名

```
import axios from 'axios';

export default {
    data() {
        return {
            curTab: 0, // 0: 总积分, 1: 年度积分
            dataList: [],
            provinceRank: '',
            totalScore: '',
            grandeRank: ''
        }
    },
```

10.4 项目功能开发

4. 积分排名

```
methods: {
        tabClick(value) {
            this.curTab = Number(value);
            this.fetchOverview(this.curTab);
            this.fetchRankList(this.curTab);
        },
```

10.4 项目功能开发

4. 积分排名

```
fetchOverview(type) {
        const that = this;
        axios.get('/api/report/myOverview', {
            params: { type }
        }).then(function (response) {
            const { error, data = {} } = response.data;

            if (error === 0) {
                that.provinceRank = data.provinceRank;
                that.totalScore = data.totalScore;
                that.grandeRank = data.grandeRank;
            }
        })
    }
```

10.4 项目功能开发

4. 积分排名

```
fetchRankList(type) {
        const that = this;
        const payload = {
            currentPage: 1,
            pageSize: 10,
            type,
        };
        axios.get('/api/report/rankList', { params: payload })
            .then(function (response) {
                const { error, data = {} } = response.data;
                if (error === 0) {
                    that.dataList = data.list || [];
                }
            })
    }
```

10.4 项目功能开发

4. 积分排名

```
mounted() {
        this.fetchOverview(this.curTab);
        this.fetchRankList(this.curTab);
    }
```

10.4 项目功能开发

4. 积分排名

- vite.config.js

增加两条响应

```
if (req.url.startsWith('/api/report/myOverview') && req.method === 'GET') {
  return sendJSON('scoreOverview.json')
}
if (req.url.startsWith('/api/report/rankList') && req.method === 'GET') {
  return sendJSON('rank.json')
}
```

10.4 项目功能开发

4. 积分排名

- router/index.js

增加一条路由

```
import LoveReport from '../views/LoveReport.vue';
…
    {
        path: '/loveReport',
        name: 'loveReport',
        component: LoveReport
    },
```

10.4 项目功能开发

4. 积分排名

10.4 项目功能开发

5. 活动列表

- 活动列表页面由顶部的搜索框、下方的活动列表构成；
- 根据搜索的关键字(可为空)向后端发送get请求，获取活动列表，当页面上拉时，可加载更多活动列表数据；
- 点击活动跳转至活动详情页面；
- 点击“我的报名”跳转至“我的报名”页面。

10.4 项目功能开发

5. 活动列表

在src 文件夹内的views子文件夹下创建 LoveActs.vue

```
<template>
  <div class="container">
    <div class="searchBox">
      <div class="search">
        <span class="search-icon"></span>
        <input placeholder="请输入地点、活动名称等关键字" v-model="keyword" />
      </div>
    </div>
```

10.4 项目功能开发

5. 活动列表

```
    <div class="body">
        <van-list v-model:loading="loading" :finished="finished" finished-text="没有更多了" @load="onLoad">
            <ActItem v-for="item in actList" :key="item.id" :data="item" @goDetails="goDetails"></ActItem>
        </van-list>
    </div>
    <button class="fBtn" @click="goMyApply">我的报名</button>
  </div>
</template>
```

10.4 项目功能开发

5. 活动列表

```
<script>
import axios from 'axios';
import ActItem from '../components/Home/ActItem.vue';

export default {
    components: {
        ActItem
    },
```

10.4 项目功能开发

5. 活动列表

```
data() {
        return {
            actList: [], // 活动列表
            loading: false, // 加载状态
            finished: false, // 是否全部加载完成
            curentPage: 1, // 当前数据请求到哪一页
            keyword: '' // 搜索的关键字
        }
    },
```

10.4 项目功能开发

5. 活动列表

```
methods: {
        fetchActList(currentPage = 1) {
            const that = this;
            const payload = {
                currentPage,
                pageSize: 10,
                keyword: this.keyword
            };
```

10.4 项目功能开发

5. 活动列表

```
// 请求数据时，则处于加载状态
this.loading = true;
axios.get('/api/actList', { params: payload })
    .then(function (response) {
        const { error, data = {} } = response.data;
        if (error === 0) {
            const currentPage = data.current;
            const list = data.list;
            // 当前为第一页，则覆盖已有的活动列表
            if (currentPage === 1) {
                that.actList = list;
            } else {
                // 在已有的活动列表后添加新的数据
                that.actList.push(...list);
            }
```

10.4 项目功能开发

5. 活动列表

```
                that.currentPage = currentPage;
                // 当前数据的页数等于总页数，则说明没有更多数据了
                that.finished = data.pageCount === currentPage;
            }
        })
        .finally(function () {
            // 请求结束后，处于非加载状态
            that.loading = false;
        })
    },
```

10.4 项目功能开发

5. 活动列表

```
 onLoad() {
    this.fetchActList(this.currentPage + 1);
  },
  goMyApply() {
    this.$router.push("/myApply");
  },
  goDetails(id) {
    this.$router.push('/actDetails?id=' + id);
  }
},
```

10.4 项目功能开发

5. 活动列表

```
watch: {
    keyword() {
        this.curentPage = 1;
        this.fetchActList(1);
    }
},
mounted() {
    this.fetchActList(1);
}
```

10.4 项目功能开发

5. 活动列表

- router/index.js

增加一条路由

```
import LoveActs from '../views/LoveActs.vue';
…
  {
    path: '/loveActs',
    name: 'loveActs',
    component: LoveActs
  },
```

10.4 项目功能开发

5. 活动列表

- vite.config.js

修改响应

```
if (req.url.startsWith('/api/actList') && req.method === 'GET') {
  return sendJSON('actList.json')
}
```

10.4 项目功能开发

5. 活动列表

10.4 项目功能开发

6. 活动详情

- 活动详情展示了活动的具体信息，若用户申请了该活动，并且处于审核状态，底部展示“撤销申请”按钮；
- 如果用户尚未申请该活动，则底部展示“立即报名”按钮；
- 页面根据链接的参数(活动的id)向后端发送get请求，来获取活动详情数据。

10.4 项目功能开发

6. 活动详情

在src 文件夹内的 views 子文件夹下创建 ActDetails.vue

```
<template>
  <div class="container">
    <div class="head">
      <p class="title">{{ data.title }}</p>
      <div class="status">
        <van-tag :color="getActStatusColor(data)" round style="margin-right:
10px">
          {{ data.canApply ? '进行中' : '已结束' }}
        </van-tag>
```

10.4 项目功能开发

6. 活动详情

```
    <van-tag :color="getActApplyColor(data)" round>
      {{ applyStatusMap[data.applyStatus] }}
    </van-tag>
  </div>
</div>
<div class="dataItem">
  <span class="label">活动时间</span>
  <p>
    {{ formatDate(data.startTime) }} - {{ formatDate(data.endTime) }}
  </p>
</div>
```

10.4 项目功能开发

6. 活动详情

```
 <div class="dataItem">
  <span class="label">活动地点</span>
  <p>{{ data.palce }}</p>
</div>
<div class="dataItem">
  <span class="label">活动简介</span>
  <p>{{ data.content }}</p>
</div>
<div class="dataItem">
  <span class="label">活动来源</span>
  <p>{{ data.publisher }}</p>
</div>
```

10.4 项目功能开发

6. 活动详情

```
<div class="dataItem">
      <span class="label">服务时长</span>
      <p>{{ data.hour }}小时</p>
    </div>
    <div class="dataItem">
      <span class="label">招募人数</span>
      <p>{{ data.total }}人</p>
    </div>
    <button class="fBtn" v-if="[2, 3].indexOf(data.applyStatus) === -1"
@click="applyClick">
      {{ data.applyStatus === 1 ? '撤销申请' : '立即报名' }}
    </button>
  </div>
</template>
```

10.4 项目功能开发

6. 活动详情

```
<script>
import axios from 'axios';
import moment from 'moment';
import { showSuccessToast, showFailToast } from 'vant';

export default {
    data() {
        return {
        data: {},
        applyStatusMap: ['', '审核中', '审核通过', '审核拒绝']
        }
    },
```

10.4 项目功能开发

6. 活动详情

```
methods: {
    formatDate(value) {
        return value ? moment(value).format('YYYY-MM-DD hh:mm') : '--'
    },
    getActStatusColor(data) {
        return data.canApply ? '#07C160' : '#ff976a'
    },
    getActApplyColor(data) {
        const list = ['', '#1989fa', '#07c160', '#f6352c'];

        return list[data.applyStatus];
    },
```

10.4 项目功能开发

6. 活动详情

```
fetchDetails() {
        const that = this;
        const payload = { id: this.$route.query.id };
        axios.get('/api/actDetails/details', {
            params: payload
        }).then(function (response) {
            const { error, data = {} } = response.data;
            if (error === 0) {
            that.data = data.details || {};
            }
        })
    },
```

10.4 项目功能开发

6. 活动详情

```
applyClick() {
        const that = this;
        const payload = {
 // applyStatus的值为1，则表示申请正在审核中，点击该按钮是需要撤销申请
// applyStatus的值不为1，则表示尚未申请参加该活动的服务，点击按钮是需要申请
            isApplay: this.data.applyStatus !== 1,
            id: this.$route.query.id
        };
```

10.4 项目功能开发

6. 活动详情

```
axios.post('/api/actDetails/apply', payload)
  .then(function (response) {
  const { error, msg } = response.data;
  if (error === 0) {
    showSuccessToast('操作成功');
    // 刷新页面数据
    that.fetchDetails();
  } else {
    showFailToast(msg || '网络错误，请稍后重试');
  }
  })
}
```

10.4 项目功能开发

6. 活动详情

```
mounted() {
        this.fetchDetails();
    }
```

10.4 项目功能开发

6. 活动详情

- vite.config.js

增加两条响应

```
if (req.url.startsWith('/api/actDetails/details') && req.method === 'GET') {
  return sendJSON('actDetails.json')
}
if (req.url.startsWith('/api/actDetails/apply') && req.method === 'POST') {
  return sendJSON('success.json')
}
```

10.4 项目功能开发

6. 活动详情

- router/index.js

增加一条路由

```
import ActDetails from '../views/ActDetails.vue';
…
    {
        path: '/actDetails',
        name: 'actDetails',
        component: ActDetails
    },
```

10.4 项目功能开发

6. 活动详情

10.4 项目功能开发

7. 我的报名

- 我的报名由选项卡、用户报名的活动列表列表组成；
- 根据选项卡的切换，向后端发送 get 请求，获取对应的活动列表；
- 当页面上拉时，可加载更多活动列表数据。

10.4 项目功能开发

7. 我的报名

在 src 文件夹内的 views 子文件夹创建 MyApply.vue

```
<template>
    <div class="container">
        <ul class="tabList">
            <li @click="tabClick(0)">
                <span :class="curTab === 0 ? 'active' : ''">全部</span>
            </li>
            <li @click="tabClick(1)">
                <span :class="curTab === 1 ? 'active' : ''">审核中</span>
            </li>
```

10.4 项目功能开发

7. 我的报名

```
    <li @click="tabClick(2)">
        <span :class="curTab === 2 ? 'active' : ''">审核通过</span>
    </li>
    <li @click="tabClick(3)">
        <span :class="curTab === 3 ? 'active' : ''">审核拒绝</span>
    </li>
</ul>
```

10.4 项目功能开发

7. 我的报名

```
        <div class="list">
            <van-list v-model:loading="loading" :finished="finished" finished-text="
没有更多了" @load="onLoad">
                <ActItem v-for="item in actList" :key="item.id" :data="item"
@goDetails="goDetails" from="applyStatus"></ActItem>
            </van-list>
        </div>
    </div>
</template>
```

10.4 项目功能开发

7. 我的报名

```
<script>
import axios from 'axios';
import ActItem from '../components/Home/ActItem.vue';

export default {
    components: {
        ActItem
    },
```

10.4 项目功能开发

7. 我的报名

```
data() {
        return {
            curTab: 0, // 0: 全部、1：审核中、2：审核通过、3：审核拒绝
            actList: [],
            loading: false,
            finished: false,
            curentPage: 1,
        }
    },
```

10.4 项目功能开发

7. 我的报名

```
methods: {
        tabClick(value) {
            this.curTab = value;
            this.currentPage = 1;
            this.fetchApplyList(1);
        },
```

10.4 项目功能开发

7. 我的报名

```
fetchApplyList(currentPage = 1) {
        const that = this;

        const payload = {
            currentPage,
            pageSize: 10,
            type: this.curTab
        };

        // 请求数据时，则处于加载状态
        this.loading = true;
```

10.4 项目功能开发

7. 我的报名

```
axios.get('/api/myApplyList', { params: payload })
 .then(function (response) {
    const { error, data = {} } = response.data;
    if (error === 0) {
        const currentPage = data.current;
        const list = data.list;
        // 当前为第一页，则覆盖已有的活动列表
        if (currentPage === 1) {
            that.actList = list;
        } else {
            // 在已有的活动列表后添加新的数据
            that.actList.push(...list);
        }
```

10.4 项目功能开发

7. 我的报名

```
                that.currentPage = currentPage;
                // 当前数据的页数等于总页数，则说明没有更多数据了
                that.finished = data.pageCount === currentPage;
            }
        })
        .finally(function () {
            // 请求结束后，处于非加载状态
            that.loading = false;
        })
    },
```

10.4 项目功能开发

7. 我的报名

```
        onLoad() {
            this.fetchApplyList(this.currentPage + 1);
        },
        goDetails(id) {
            this.$router.push('/actDetails?id=' + id);
        }
    },
    mounted() {
        this.fetchApplyList(1);
    }
}
</script>
```

10.4 项目功能开发

7. 我的报名

- vite.config.js

增加一条响应

```
if (req.url.startsWith('/api/myApplyList') && req.method === 'GET') {
 return sendJSON('myApplyList.json')
}
```

10.4 项目功能开发

7. 我的报名

- router/index.js

增加一条路由

```
import MyApply from '../views/MyApply.vue';
…
    {
        path: '/myApply',
        name: 'myApply',
        component: MyApply
    },
```

10.4 项目功能开发

7. 我的报名

10.4 项目功能开发

8. 服务纪实

- 页面由表单、按钮组成；
- 其中活动来源、服务时间、服务时长的分别为下拉选择框；
- 用户填写个人参与志愿者活动的信息后，通过“提交”按钮向后端发送 post 请求来上传活动参与详情。

10.4 项目功能开发

8. 服务纪实

在 src 文件夹内的 views 子文件夹下创建 UploadService.vue

```
<template>
  <div>
    <div class="frm">
      <van-field v-model="pushlisher.text" is-link readonly label="活动来源"
placeholder="请选择活动来源" @click="fieldCLick(0)" />
      <van-field v-model="date" is-link readonly label="服务时间" placeholder="
请选择日期" @click="fieldCLick(1)" />
      <van-field v-model="duration.text" is-link readonly label="服务时长"
placeholder="请选择时长" @click="fieldCLick(2)" />
```

10.4 项目功能开发

8. 服务纪实

在 src 文件夹内的 views 子文件夹下创建 UploadService.vue

```
    <van-field v-model="content" label="服务内容" placeholder="请对服务内容进行简要描述" label-align="top" type="textarea"
      maxlength="100" show-word-limit />
    <van-uploader v-model="fileList" multiple max-count="2" :upload-text="uploadTxt" image-fit="cover"
      style="padding: 10px" />
  </div>
  <div class="footer">
    <van-button type="primary" color="#ff5d23" block @click="submit">提交</van-button>
  </div>
```

10.4 项目功能开发

8. 服务纪实

在 src 文件夹内的 views 子文件夹下创建 UploadService.vue

```
    <van-popup v-model:show="showPicker" round position="bottom">
      <van-picker :columns="columns" @cancel="pickerCancel"
@confirm="pickerConfirm" />
    </van-popup>
    <van-popup v-model:show="showDatePicker" round position="bottom">
      <van-date-picker v-model="currentDate" title="选择日期" :min-
date="minDate" :max-date="maxDate" @cancel="pickerCancel"
        @confirm="pickerConfirm" />
    </van-popup>
  </div>
</template>
```

10.4 项目功能开发

8. 服务纪实

```
<script>
import { showToast, showSuccessToast, showFailToast } from 'vant';
import axios from 'axios';

function dateRange() {
  const curTimeStamp = new Date();
  const tYear = curTimeStamp.getFullYear();
  const tMonth = curTimeStamp.getMonth();
  const date = curTimeStamp.getDate();

  return {
    currentDate: [tYear, tMonth, date],
    minDate: new Date(tYear - 2, tMonth, date),
    maxDate: new Date(tYear, tMonth, date),
  };
}
```

10.4 项目功能开发

8. 服务纪实

```
let pushlisherList = []; // 存储活动来源选项
let durationList = []; // 存储活动时长选项

export default {
  data() {
    return {
      columns: [],
      pickerId: null, // 0: 活动来源, 1: 服务时间，  2：  服务时长
      pushlisher: { value: null, text: '' }, // 选择的服务社区
      duration: { value: null, text: '' }, // 服务时长
      date: null, // 服务时间
      content: '', // 服务内容
      fileList: [], // 上传的图片
      currentDate: null, // 选择的服务时间
      minDate: null, // 时间选择的最小值
      maxDate: null, // 时间选择的最大值
    }
  },
```

10.4 项目功能开发

8. 服务纪实

```
computed: {
    showPicker() {
      return this.pickerId === 0 || this.pickerId === 2;
    },
    showDatePicker() {
      return this.pickerId === 1;
    },
    uploadTxt() {
      const length = this.fileList.length;

      return '添加照片' + length + '/2';
    },
  },
```

10.4 项目功能开发

8. 服务纪实

```
methods: {
    fieldCLick(value) {
      this.pickerId = value;

      if (value === 0) {
        this.columns = pushlisherList;
      } else if (value === 2) {
        this.columns = durationList;
      }

      console.log(this.columns);
    },
```

10.4 项目功能开发

8. 服务纪实

```
pickerCancel() {
  this.pickerId = null;
},
pickerConfirm({ selectedValues, selectedOptions }) {
  const curPickerId = this.pickerId;

  this.pickerId = null;
  // 当前选择的是服务时间
  if (curPickerId === 1) {
    const data = selectedValues;
    // 更新服务时间的数据
    this.date = data.join('.');
    return;
  }
```

10.4 项目功能开发

8. 服务纪实

```
    // 非服务时间（活动来源、服务时长）
    const data = selectedOptions[0];
    const payload = {
      value: data.value,
      text: data.text
    }

    // 如果是选择的活动来源
    if (curPickerId === 0) {
      // 更新活动来源的数据
      this.pushlisher = payload;

      return;
    }
    // 选择的是服务时长，更新服务时长的数据
    this.duration = payload;
  },
```

10.4 项目功能开发

8. 服务纪实

```
fetchPublisherlist() {
  axios.get('/api/act/publisherList')
    .then(function (response) {
      const { error, data = {} } = response.data;

      if (error === 0) {
        pushlisherList = data.list || [];
      }
    })
},
```

10.4 项目功能开发

8. 服务纪实

```
fetchDurations() {
  axios.get('/api/act/durationsList')
    .then(function (response) {
      const { error, data = {} } = response.data;

      if (error === 0) {
        durationList = data.list || [];
      }
    })
},
```

10.4 项目功能开发

8. 服务纪实

```
submit() {
    if (!this.pushlisher.text) {
      showToast('未选择活动来源');
      return;
    }

    if (!this.date) {
      showToast('未选择服务时间');
      return;
    }

    if (!this.duration.text) {
      showToast('未选择服务时长');
      return;
    }
```

10.4 项目功能开发

8. 服务纪实

```
if (!this.content) {
  showToast('未填写服务内容');
  return;
}

if (!this.fileList.length) {
  showToast('未上传服务照片');
  return;
}
```

10.4 项目功能开发

8. 服务纪实

```
const payload = {
    pushlisher: this.pushlisher.value,
    date: +new Date(this.date),
    duration: this.duration.value,
    content: this.content,
    fileList: this.fileList,
  };
```

10.4 项目功能开发

8. 服务纪实

```
    axios.post('/api/act/uploadService', payload)
      .then(function (response) {
        const { error, msg } = response.data;

        if (error === 0) {
          showSuccessToast('操作成功')
        } else {
          showFailToast(msg || '网络错误，请稍后重试');
        }
      })
  }
},
```

10.4 项目功能开发

8. 服务纪实

```
created() {
    const { currentDate, minDate, maxDate } = dateRange();

    this.currentDate = currentDate;
    this.minDate = minDate;
    this.maxDate = maxDate;

    this.fetchDurations();
    this.fetchPublisherlist()
  }
};
</script>
```

10.4 项目功能开发

8. 服务纪实

- vite.config.js

增加三条响应

```
if (req.url.startsWith('/api/act/publisherList') && req.method === 'GET') {
  return sendJSON('publishers.json')
}
if (req.url.startsWith('/api/act/durationsList') && req.method === 'GET') {
  return sendJSON('durationList.json')
}
if (req.url.startsWith('/api/act/uploadService') && req.method === 'POST') {
  return sendJSON('success.json')
}
```

10.4 项目功能开发

8. 服务纪实

- router/index.js

增加一条路由

```
import UploadService from '../views/UploadService.vue';
…
  {
    path: '/UploadService',
    name: 'UploadService',
    component: UploadService
  },
```

10.4 项目功能开发

8. 服务纪实

10.4 项目功能开发

9. 我的服务

- 用户选择不同的年份后，页面向后端发送 get 请求，来获取对应的数据，并更新页面内容。
- 页面顶端显示用户信息及年份；
- 中间显示当年度的服务次数、服务时长和服务积分；
- 底部则列表服务的列表信息。

10.4 项目功能开发

9. 我的服务

- 设计三个子组件：User.vue、Overview.vue和RecordItem.vue，存放于src文件夹内components中的ServiceRecords。

10.4 项目功能开发

9. 我的服务

- 子组件User.vue

```
<template>
  <div class="container">
    <div class="user">
      <img class="user-avatar" :src="userData.avatar" />
      <div>
        <p class="user-name">{{ userData.name }}</p>
        <span class="user-score">{{ userData.totalScore }}积分</span>
      </div>
    </div>
    <van-button round hairline size="mini" icon="arrow-down" color="#ff5d23"
@click="showPicker = true">
      {{ date.text }}
    </van-button>
    <van-popup v-model:show="showPicker" round position="bottom">
      <van-picker :columns="dateList" @cancel="cancel" @confirm="confirm" />
    </van-popup>
  </div>
</template>
```

10.4 项目功能开发

9. 我的服务

- 子组件User.vue

```
<script>
import axios from 'axios';

export default {
props: ['dateList', 'date'],
    data() {
        return {
            showPicker: false,
            userData: {}
        }
    },
```

10.4 项目功能开发

9. 我的服务

- 子组件User.vue

```
methods: {
    fetchUserData() {
        const that = this;

        axios.get('/api/service/userScore')
            .then(function (response) {
                const { error, data = {} } = response.data;

                if (error === 0) {
                    that.userData = data || {};
                }
            })
    },
```

10.4 项目功能开发

9. 我的服务

- 子组件User.vue

```
    cancel() {
        this.showPicker = false;
    },
    confirm({ selectedOptions }) {
        const data = selectedOptions[0];
        this.showPicker = false;
        this.$emit('updateYear', data);
    },
  },
  mounted() {
    this.fetchUserData();
  },

}
</script>
```

10.4 项目功能开发

9. 我的服务

在 src 文件夹内的 views 子文件夹创建 ServiceRecords.vue

```
<template>
  <div>
    <div class="header">
      <User @updateYear="updateYear" :date="year" :dateList="dateList">
      </User>
    </div>
</template>
<script>
import axios from 'axios';
import User from '../components/ServiceRecords/User.vue’;

export default {
  components: {
    User,
    Overview,
    RecordItem
  },
```

10.4 项目功能开发

9. 我的服务

```
data() {
  return {
    year: {}, // 当前选中的年份数据
    dateList: [] // 年份列表
  }
},
```

10.4 项目功能开发

9. 我的服务

```
methods: {
  fetchYearList() {
    const that = this;
    axios.get('/api/service/yearList')
      .then(function (response) {
        const { error, data = {} } = response.data;

        if (error === 0) {
          const list = data.list || [];
          that.dateList = list;
          that.year = list[list.length - 1];
        }
      })
  },
```

10.4 项目功能开发

9. 我的服务

```
 updateYear(yearValue) {
      this.year = yearValue;
    }
  },
  mounted() {
    this.fetchYearList();
  }
}
```

10.4 项目功能开发

9. 我的服务

- vite.config.js

增加两条响应

```
if (req.url.startsWith('/api/service/userScore') && req.method === 'GET') {
  return sendJSON('userScore.json')
}
if (req.url.startsWith('/api/service/yearList') && req.method === 'GET') {
  return sendJSON('yearList.json')
}
```

10.4 项目功能开发

9. 我的服务

- router/index.js

增加一条路由

```
import ServiceRecords from '../views/ServiceRecords.vue';
…
    {
        path: '/serviceRecords',
        name: 'serviceRecords',
        component: ServiceRecords
    },
```

10.4 项目功能开发

9. 我的服务

- 子组件Overview.vue

```
<template>
  <ul class="overview">
    <li>
      <p class="num">
        <span class="highColor">{{  data.times }}</span>次
      </p>
      <span class="label">服务次数</span>
    </li>
    <li>
      <p class="num">
        <span class="highColor">{{  data.duration }}</span>小时
      </p>
      <span class="label">服务时长</span>
    </li>
```

10.4 项目功能开发

9. 我的服务

- 子组件Overview.vue

```
        <li>
          <p class="num">
            <span class="highColor">{{  data.score }}</span>分
          </p>
          <span class="label">服务积分</span>
        </li>
    </ul>
</template>
```

10.4 项目功能开发

9. 我的服务

- 子组件Overview.vue

```
<script>
import axios from 'axios';
export default {
    props: ['year'],
    data() {
        return  {
            data: {}
        }
    },
```

10.4 项目功能开发

9. 我的服务

- 子组件Overview.vue

```
methods: {
        fetchYearScore() {
            const that = this;
            const year = this.year;
            axios.get('/api/service/yearScore', {
                params: { year }
            })
                .then(function (response) {
                    const { error, data = {} } = response.data;

                    if (error === 0) {
                        that.data = data || {};
                    }
                })
        },
    },
```

10.4 项目功能开发

9. 我的服务

- 子组件Overview.vue

```
watch: {
        year() {
            this.fetchYearScore();
        }
    }
}
</script>
```

10.4 项目功能开发

9. 我的服务

ServiceRecords.vue

```
<div class="overview">
    <Overview :year="year.value"></Overview>
</div>


import Overview from '../components/ServiceRecords/Overview.vue';

export default {
  components: {
    User,
    Overview,
  },
```

10.4 项目功能开发

9. 我的服务

- vite.config.js

增加一条响应

```
if (req.url.startsWith('/api/service/yearScore') && req.method === 'GET') {
  return sendJSON('yearScore.json')
}
```

10.4 项目功能开发

9. 我的服务

- 子组件RecordItem.vue

```
<template>
  <div class="container">
    <div class="details" @click="detailsClick">
      <div class="imgBox">
        <img :src="data.pic" />
        <p class="status" :style="{ backgroundColor: bgColorMap[data.status] }">
          {{ statusMap[data.status] }}
        </p>
      </div>
      <div class="main">
        <p class="content">{{ data.content }}</p>
        <p class="bottomTxt">
          <span style="margin-right: 10px">{{ data.publisher }}</span>
          <span>{{ formatDate(data.time) }}</span>
        </p>
      </div></div></div></template>
```

10.4 项目功能开发

9. 我的服务

- 子组件RecordItem.vue

```
<script>
import moment from 'moment';

export default {
    name: 'recordItem',
    props: ['data', 'goDetails'],
    data() {
        return {
            statusMap: ['审核中', '审核通过', '审核驳回'],
            bgColorMap: ['#1989fa', '#07c160', '#f6352c']
        }
    },
```

10.4 项目功能开发

9. 我的服务

- 子组件RecordItem.vue

```
    methods: {
        formatDate(value) {
            return value ? moment(value).format('YYYY-MM-DD hh:mm') : '--'
        },
        detailsClick() {
            this.$emit('goDetails', this.data.id);
        }
    }
}
</script>
```

10.4 项目功能开发

9. 我的服务

ServiceRecords.vue

```
<van-list v-model:loading="loading" :finished="finished" finished-text="没有更多了"
@load="onLoad">
    <RecordItem v-for="item in dataList" :key="item.id" :data="item"
@goDetails="goDetails">
    </RecordItem>
</van-list>


import RecordItem from '../components/ServiceRecords/RecordItem.vue';

export default {
  components: {
    User,
    Overview,
    RecordItem
  },
```

10.4 项目功能开发

9. 我的服务

ServiceRecords.vue

```
data() {
  return {
    dataList: [],
    loading: false,
    finished: false,
    curentPage: 1,
    year: {}, // 当前选中的年份数据
    dateList: [] // 年份列表
  }
},
```

10.4 项目功能开发

9. 我的服务

ServiceRecords.vue

```
fetchServiceRecords(currentPage = 1) {
    const that = this;

    const payload = {
      currentPage,
      pageSize: 10,
      year: this.year.value
    };

    // 请求数据时，则处于加载状态
    this.loading = true;

    axios.get('/api/service/list', { params: payload })
      .then(function (response) {
        const { error, data = {} } = response.data;
```

10.4 项目功能开发

9. 我的服务

ServiceRecords.vue

```
    if (error === 0) {
      const currentPage = data.current;
      const list = data.list;

      // 当前为第一页，则覆盖已有的活动列表
      if (currentPage === 1) {
        that.dataList = list;
      } else {
        // 在已有的活动列表后添加新的数据
        that.dataList.push(...list);
      }

      that.currentPage = currentPage;
      // 当前数据的页数等于总页数，则说明没有更多数据了
      that.finished = data.pageCount === currentPage;
    }
  })
```

10.4 项目功能开发

9. 我的服务

- vite.config.js

增加一条响应

```
if (req.url.startsWith('/api/service/list') && req.method === 'GET') {
  return sendJSON('serviceList.json')
 }
```

10.4 项目功能开发

9. 我的服务

10.4 项目功能开发

10. 服务列表详情

- 页面根据链的的参数（活动的id）向后端发送 get 请求，来获取服务记录的详情数据。

10.4 项目功能开发

10. 服务列表详情

在 src 文件夹内的 views 子文件下创建 ServiceDetai.vue

```
<template>
  <div class="container">
    <div class="dataItem">
      <span class="label">服务时间</span>
      <span>
        {{ formatDay(data.actTime) }}
      </span>
    </div>
    <div class="dataItem">
      <span class="label">服务地点</span>
      <span>
        {{ data.place }}
      </span>
    </div>
```

10.4 项目功能开发

10. 服务列表详情

```
<div class="dataItem">
    <span class="label">服务内容</span>
    <span>{{ data.content }}</span>
</div>
<div class="dataItem">
    <span class="label">服务照片</span>
    <div>
        <img class="serviceImg" v-for="(url) in data.imgList" :src="url" :key="url"/>
    </div>
</div>
<div class="dataItem">
    <span class="label">服务时长</span>
    <span>
        {{ data.hour }}小时
    </span>
</div>
```

10.4 项目功能开发

10. 服务列表详情

```
<div class="dataItem">
    <span class="label">纪实时间</span>
    <span> {{ formatDate(data.uploadTime) }}</span>

</div>
<div class="dataItem">
    <span class="label">审核状态</span>
    <span>
        <span class="status" :style="{ backgroundColor: bgColorMap[data.status] }">
            {{ statusMap[data.status] }}
        </span>
    </span>
</div>
```

10.4 项目功能开发

10. 服务列表详情

在 src 文件夹内的 views 子文件下创建 ServiceDetai.vue

```
    <div class="footer">
        <p class="title">{{ titleMap[data.status] }}</p>
        <p class="score" v-if="data.status === 1">4分</p>
    </div>
  </div>
</template>
```

10.4 项目功能开发

10. 服务列表详情

```
<script>
import axios from 'axios';
import moment from 'moment';

export default {
    data() {
        return {
            data:{},
            statusMap: ['审核中', '审核通过', '审核驳回'],
            bgColorMap: ['#1989fa', '#07c160', '#f6352c'],
            titleMap: ['正在审核中，请耐心等待', '感谢您的服务，请收下服务积分', '审核被驳回了哟']
        }
    },
```

10.4 项目功能开发

10. 服务列表详情

```
methods: {
    formatDate(value) {
        return value ? moment(value).format('YYYY-MM-DD hh:mm') : '--'
    },
    formatDay(value) {
        return value ? moment(value).format('YYYY-MM-DD') : '--'
    },
    fetchDetails() {
        const that = this;
        const payload = { id: this.$route.query.id };

        axios.get('/api/service/details', {
            params: payload
        }).then(function (response) {
            const { error, data = {} } = response.data;
```

10.4 项目功能开发

10. 服务列表详情

```
            if (error === 0) {
                that.data = data.details || {};
            }
        })
    },
  },
  mounted() {
    this.fetchDetails();
  }
}

</script>
```

10.4 项目功能开发

10. 服务列表详情

- vite.config.js

增加一条响应

```
if (req.url.startsWith('/api/service/details') && req.method === 'GET') {
  return sendJSON('recordDetails.json')
}
```

10.4 项目功能开发

10. 服务列表详情

- router/index.js

增加一条路由

```
import ServiceDetail from '../views/ServiceDetail.vue';
…
  {
    path: '/serviceDetail',
    name: 'serviceDetail',
    component: ServiceDetail
  },
```

10.4 项目功能开发

10. 服务列表详情

服务时间	2022-12-25
服务地点	中山公园
服务内容	参与2022年马拉松志愿者,及时清理路边的纸杯、饮料瓶等垃圾，不让美丽的花海受到污染，确保了比赛生态、文明、安全、和谐
服务照片	
服务时长	3小时
纪实时间	2022-12-25 08:30
审核状态	审核通过

感谢您的服务，请收下服务积分

4分