**实验报告五**

学号：2019329621004 姓名：梅雨欣

班级：计算机科学与技术19（1）班 联系电话：13007239653

日期：2022.5.1 上课节次：周五12

**注意：**

1.上方的姓名、学号、班级、联系电话、日期、上课节次必须填写清晰准确；

2.文件名为“学号­\_姓名\_实验五.docx”，PDF格式也可以；

3.本次实验请在7天内完成，及时提交到蓝墨云班课。

1. **实验目标：**

1.掌握使用Vue-CLI脚手架工具在自己的电脑上建立项目，并运行调试工具。

2.学习axios的使用及跨域请求时反向代理的设置。

1. **实验内容：**

1.Vue-CLI脚手架工具搭建一个Web项目；

2.在vue.config.js文件中设置反向代理；

3.在项目中使用axios跨域请求获取天气预报数据，编辑制作一个天气预报页面，要求适合手机屏幕上浏览。

推荐使用天气预报网络接口：http://www.tianqiapi.com/index，也使用其他网络接口。

1. **实验过程：（描述详细步骤和成果截图）**

**1、使用脚手架创建项目**

|  |
| --- |
| **vue create myaxios** |

**2、安装并全局配置axios**

|  |
| --- |
| **安装命令：npm install axios –save**  **全局配置：在main.js中引入axios** |

**3、跨域请求数据**

**本实验在对应的VueWeather.vue组件中写入以下代码获取杭州的天气数据，天气api为易客云天气API。**

|  |
| --- |
| export default {    data() {      return {        // 未来7日天气        weatherList:[],      }    },      created() {      this.getWeatherList()    },    methods: {      getimg(sky){        return 'assets/'+sky+'.png'      },      async getWeatherList(){        const {data:res}=await this.$axios.get('https://v0.yiketianqi.com/api',        {params:{          appid:'xxxxxxxx',          appsecret:'xxxxx',          version:'v9',          cityidL:'1012100101',          unescape:1        }})        this.weatherList=res.data        console.log(res)      }    },  } |

**在vue.config.js中设置反向代理。**

|  |
| --- |
| const { defineConfig } = require('@vue/cli-service')  module.exports = defineConfig({    transpileDependencies: true,    devServer: {      //port: 8080,      // 配置代理      proxy: {        '/api': {          // 以 “/api” 开头的url代理到下边的target属性的值          target: 'http://127.0.0.1:8080/',          changeOrigin: true,          // ws: true, // 是否启用websockets          pathRewrite: {            // 路径重写            '^/api': 'https://v0.yiketianqi.com/api'          }        }      }    },  }) |

**4、编辑新组件VueWeather.vue，发出axios请求获取数据，并显示在页面中。最后在其他页面使用VueWeather.vue组件。**

|  |
| --- |
| <!--   \* @Description:   \* @Author: Diredre   \* @Date: 2022-05-01 16:41:56   \* @LastEditTime: 2022-05-01 20:20:43   \* @LastEditors: Diredre  -->  <template>    <div>      <h2>杭州</h2>未来七天天气      <ul id="nav">        <li v-for="(item,i) in weatherList" :key="i">          <h4>{{item.date}}</h4>          <div>            <span>{{item.week}}</span>            <span>{{item.tem}}°C</span>            <span>{{item.wea}}</span>          </div>          <img v-if="item.wea\_img=='qing'" src="../assets/sunny.png" alt="">          <img v-else-if="item.wea\_img=='yin'" src="../assets/grey.png" alt="">          <img v-else-if="item.wea\_img=='yu'" src="../assets/rainy.png" alt="">          <img v-else src="../assets/cloudy.png" alt="">        </li>      </ul>    </div>  </template>  <script>  export default {    data() {      return {        // 未来7日天气        weatherList:[],      }    },      created() {      this.getWeatherList()    },    methods: {      getimg(sky){        return 'assets/'+sky+'.png'      },      async getWeatherList(){        const {data:res}=await this.$axios.get('https://v0.yiketianqi.com/api',        {params:{          appid:'xxxxxxx',          appsecret:'xxxxxx',          version:'v9',          cityidL:'1012100101',          unescape:1        }})        this.weatherList=res.data        console.log(res)      }    },  }  </script>  <style scoped>  #nav {    height: 205px;    background-color:rgb(127, 202, 255);  }  #nav ul{    list-style: none;    margin-left: 20px;  }  #nav li{    display: block;    float: left;    padding-left: 10px;    padding-right: 10px;    padding-bottom: 20px;    margin-left: 10px;    margin-right: 10px;  }  #nav li img{    display: block;    margin: 0 auto;    width: 100px;    height: 100px;  }  #nav li:hover{    background-color: rgb(170, 198, 223);  }  li div span{    margin-right: 20px;  }  </style> |

**5、最后效果**

|  |
| --- |
| **鼠标滑过hover效果：**    **手机适配情况：** |

1. **心得体会：**

**通过此次实验，我学会了axios的基本用法，以及跨域请求时反向代理的设置。其实Axios是基于promise的HTTP库，类似jQuery的Ajax，用于请求数据，可以用于浏览器和node.js之类。它安装简单，使用丰富，除了此次学习了get/post请求，其实它还能写拦截器，以后可以尝试拓展一下。除此之外，我了解到vue应用使用Nginx部署，它有专门的反向代理设置，网上有很多配置方法，比如在vue.config.js中配置，或者在index.js中配置proxyTable，总接口文件中配置生产模式的域名调整，再在接口内调用即可。这些方法以后可以试试看。总的来说，这次实验收获颇丰。**