城市信息查询应用程序设计和实现报告

2251323 朱煜昊

项目概述

本项目旨在开发一个城市信息查询工具，使用户能够通过输入城市名称来获取该城市的全面信息。信息包括城市简介、经济指标、环境状况、社会新闻、技术发展、在线地图以及最新的城市新闻。该工具采用Vue 3 + Vite框架和Element Plus组件库进行前端开发，后端数据通过API调用获取。

技术栈

- **前端框架**：Vue 3 + Vite
- **UI组件库**：Element Plus
- 数据源：
  - Wikipedia API：城市简介
  - 高德地图API：天气和地图组件
  - 聚合API：城市新闻
  - APISpace：空气质量
- 其他工具：
  - AMapLoader：高德地图API调用
  - 代理服务器：解决跨域问题

实现细节

1. 城市简介

- **数据源**：Wikipedia API（文档：[API:Get the contents of a page - MediaWiki]）
- **实现方式**：通过Wikipedia API获取城市简介信息，并在前端展示。
- **具体实现**：

```
export async function fetchGeneralInfo(cityName) {
    console.log("开始获取基本数据")
    const params = {
        action: "query",
        format: "json",
        prop: "revisions",
        titles: cityName,
        rvprop: "content",
        rvslots: "*",
        formatversion: "2",
    };
```

```
    const url = "/wikipediaAPI/w/api.php";
    try {
        const response = await axios.get(url, { params }); // 使用 await 等待
Promise 解决
        const content =
response.data.query.pages[0].revisions[0].slots.main.content;
        const summary = await getSummary(content);//简单处理获得简介
        return summary; // 返回处理后的数据
    } catch (error) {
        // 处理错误情况
        if (error.code === 'ECONNABORTED') {
            console.log('连接超时，请检查您的网络连接或服务器状态。');
        } else {
            console.log('请求失败:', error);
        }
        return null; // 在错误情况下返回 null 或适当的错误信息
    }
}
```

2. 天气和地图组件

- **数据源**：高德地图
    - 地图API文档：行政区查询-服务插件和工具-进阶教程-地图 JS API 2.0 | 高德地图API (amap.com)
    - 天气API文档：天气-服务插件和工具-进阶教程-地图 JS API 2.0 | 高德地图API (amap.com)
- **实现方式**：安装AMapLoader包后，通过AMapLoader调用高德地图API，获取并展示城市的天气、空气质量和地图信息。

```
import AMapLoader from "@amap/amap-jsapi-loader";
```

- **具体实现**：

```
async function getLngLatAndDrawBoundsFn() {
if (!district) {
    let opts = {
      subdistrict: 0,
      extensions: 'all',
      level: 'district',
    };
    district = new AMap.DistrictSearch(opts);
}
console.log("开始定位" + cityName.value);
district.setLevel("district");
const result = await new Promise((resolve, reject) => {
    district.search(cityName.value, function (status, result) {
      if (status === "complete") {
        resolve(result);
      } else {
        reject('搜索失败，请正确填写名称或更新其他名称');
      }
    });
```

```
});
if (polygon) {
    map.value.remove(polygon);
    polygon = null;
}
if (!result || !result.districtList || !result.districtList[0]) {
    console.warn('请正确填写名称或更新其他名称');
    return;
}
let bounds = result.districtList[0].boundaries;
Lng = result.districtList[0].center.lng;
Lat = result.districtList[0].center.lat;
if (bounds) {
    for (let i = 0; i < bounds.length; i++) {
      bounds[i] = [bounds[i]];
    }
    polygon = new AMap.Polygon({
      strokeWeight: 1,
      path: bounds,
      fillOpadistrict: 0.4,
      fillColor: '#80d8ff',
      strokeColor: '#0091ea',
    });
    map.value.add(polygon);
    map.value.setFitView(polygon);
}
return { Lng, Lat };
};

//获取天气
async function getWeatherFn() {
let Weather = new AMap.Weather();
const weather = await new Promise((resolve, reject) => {
    Weather.getLive(cityName.value, function (err, data) {
      if (!err) {
        resolve(data);
      }
      else {
        reject("天气获取失败");
      }
    });
});
const weatherTable = [
    {
      prop: "数据时间",
      data: weather.reportTime
    },
    {
      prop: "温度",
      data: weather.temperature
    },
    {
      prop: "天气",
      data: weather.weather
```

```
    },
    {
      prop: "风向",
      data: weather.windDirection
    },
    {
      prop: "风力",
      data: weather.windPower
    },
]
console.log(weather);
return weatherTable;
}
```

3. 城市新闻

- **数据源**：聚合API（文档：地区新闻-地区新闻API接口-免费API接口-聚合数据 (juhe.cn)）
- **实现方式**：调用聚合API获取城市的最新新闻，并在前端以列表形式展示。
- **具体实现**：

```
export async function getNews(cityName) {
    const data = cityName.slice(0, -1);//去除“省/市”字样
    const requestParams = {
        key: apiKey,
        areaname: data,
        word: '',
        page: '20',
    };

    // 发起接口网络请求
    const response = await axios.get(apiUrl, { params: requestParams });
    const responseResult = response.data.result.list;
    console.log(responseResult);
    return responseResult;
}
```

4. 空气质量

- **数据源**：APISpace（文档：空气质量查询-APISpace-API数据接口-API接口大全-免费API接口服务）
- **实现方式**：调用APISpace的API获取城市的当前空气质量，并在前端以表格形式展示。
- **具体实现**：

```
export async function fetchEnvironment(LngLat) {
    const url =
`https://eolink.o.apispace.com/34324/air/v001/aqi?lonlat=${LngLat.Lng},${LngLa
t.Lat}`;
    const headers = {
        'X-APISpace-Token': '9eenljb5pg26t3p028da518axi7feqt4'
    };
    const response = axios.get(url, { headers });
    console.log(response);
```

```
        return (await response).data.result.realtimeAqi;
    }
```

挑战与解决方案

跨域问题

- **城市简介**：通过设置代理服务器，解决了从Wikipedia API获取数据时的跨域问题。
- **城市新闻**：采用CORS（跨源资源共享）策略，允许前端应用从聚合API获取数据。

数据整合

- **数据一致性**：确保从不同API获取的数据在前端展示时格式一致，用户体验连贯。
- **数据更新**：设计了数据缓存机制，减少API调用频率，提高应用性能。

结论

本项目成功实现了一个功能齐全的城市信息查询工具，通过Vue 3 + Vite框架和Element Plus组件库，为用户提供了一个直观、易用的界面。通过解决跨域问题和数据整合挑战，确保了应用的稳定性和用户体验。未来，我们可以考虑增加更多数据源和功能，以进一步提升应用的实用性和吸引力。