移动软件开发 | 实验 2：天气查询小程序

洪佳荣 22070001035　　**高峰老师，2024 夏季 | 截止时间：**2024 年 8 月 20 日

源代码：https://github.com/hongjr03/MiniProgram
博客：https://www.jrhim.com/p/2024a/mini-program-2

一、实验目的

1. 掌握服务器域名配置和临时服务器部署；
2. 掌握 wx.request 接口的用法。

二、实验步骤

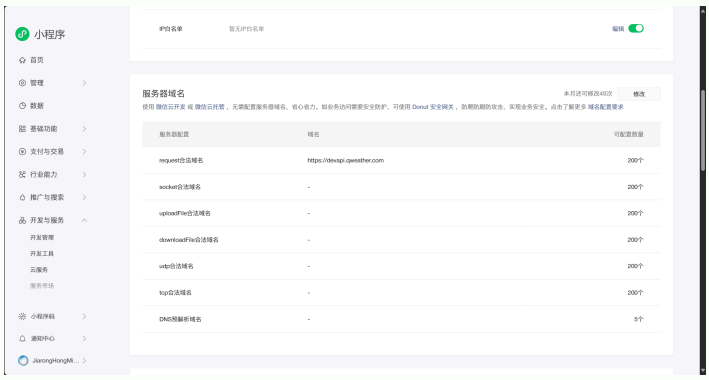
API 密钥申请和配置

首先从和风天气官网申请一个免费的 API 密钥，用于查询天气信息。

根据官网的 API 配置文档，要调取这个接口，需要使用以下 URL：

```
https://devapi.qweather.com/v7/weather/now?location=xxx&key=xxx
```

要正确使用这个接口，需要在微信开发者工具中配置服务器域名，将 `devapi.qweather.com` 添加到 request 合法域名中。



创建小程序

这里直接基于上一次实验的空白小程序继续开发。复制项目后，下载老师提供的图标素材，放到项目的对应目录下。目录结构如下：

```
.
├─images
│  ├─weather_icon_s1_bw
│  ├─weather_icon_s1_color
│  └─weather_icon_s2
├─pages
│  └─index
└─utils
```

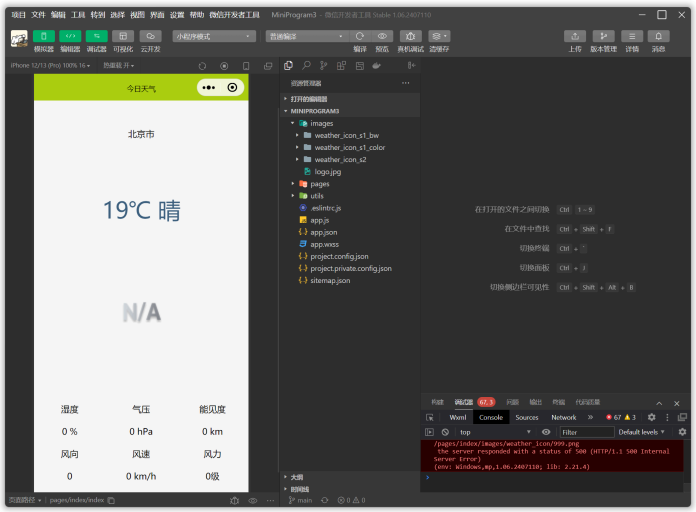
视图设计

根据实验文档，进行页面设计。主要包括：

- 页面中央的城市显示、温度、天气图标
- 页面下方的天气数据列表

和风天气图标有两种风格，分别是黑白和彩色，还有一种是简约风格。这里使用彩色风格的图标。其对应关系可以查看官方文档。根据实验文档这里使用 N/A 先代替。

涉及的代码可见 commit。



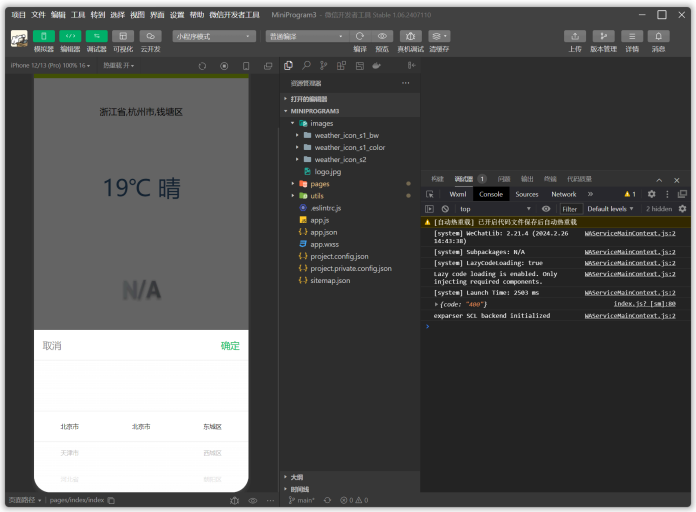
逻辑实现

在 index.wxml 中，添加页面元素。城市切换使用 picker 组件，即

```
<picker mode="region" bindchange="regionChange">
    <view>{{region}}</view>
</picker>
```

让 mode 为 region，即可以选择省市区三级联动。绑定 change 事件，即选择后触发 regionChange 方法。

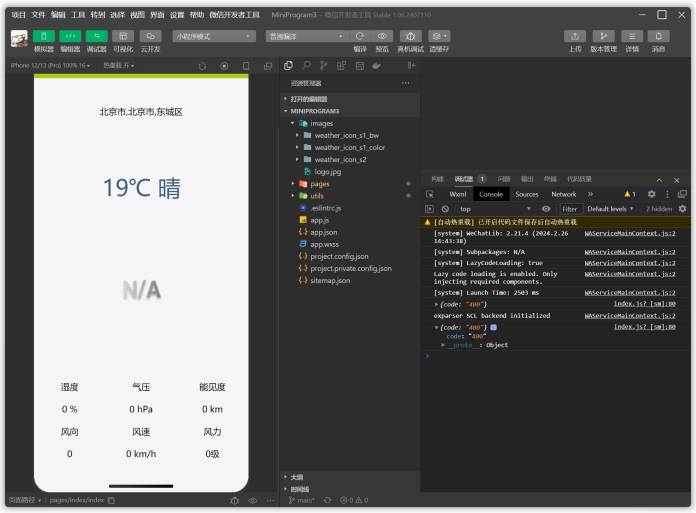
```
regionChange: function (e) {
  this.setData({
    region: e.detail.value,
  });
},
```



然后即可实现根据选择的城市获取天气信息。使用 wx.request 方法，传入 URL 和参数，即可获取天气信息。

```
getWeather: function () {
  var that = this;
  wx.request({
    url: "https://devapi.qweather.com/v7/weather/now",
    data: {
      location: that.data.region[1],
      key: "3e916ed15fc14f83a3xxxxxxxx237cb8"
    },
    success: function (res) {
      console.log(res.data);
    }
  });
}
```

然而仅这样处理并不能正确将城市信息转化为需要的 location，解决方法见 问题总结与体会。



如此运行后，观察调试终端可以看到，调用 API 后返回的是一个 JSON 对象，需要解析后才能使用，结构为：

```
{
  "code":"200","updateTime":"2024-08-20T00:03+08:00",
  "fxLink":"https://www.qweather.com/weather/hangzhou-101210101.html",
  "now":{
    "obsTime":"2024-08-20T00:00+08:00",
    "temp":"30",
    "feelsLike":"33",
    "icon":"151",
    "text":"多云",
    "wind360":"135",
    "windDir":"东南风",
    "windScale":"2",
    "windSpeed":"10",
    "humidity":"71",
    "precip":"0.0",
    "pressure":"999",
    "vis":"30",
    "cloud":"91",
    "dew":"25"
  },
  "refer":{
    "sources":["QWeather"],
    "license":["CC BY-SA 4.0"]
  }
}
```


根据返回的 JSON 对象，将数据渲染到页面上。

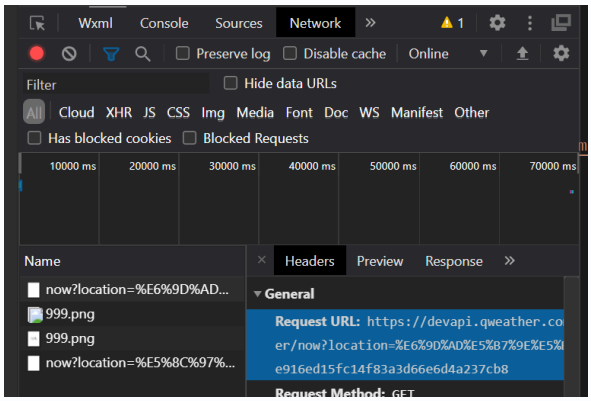
```
that.setData({
  temp: res.data.now.temp,
  weather: res.data.now.text,
  icon: res.data.now.icon,
  humidity: res.data.now.humidity,
  pressure: res.data.now.pressure,
  visibility: res.data.now.vis,
  windDir: res.data.now.windDir,
  windSpeed: res.data.now.windSpeed,
  windScale: res.data.now.windScale,
});
```

三、程序运行结果

四、问题总结与体会

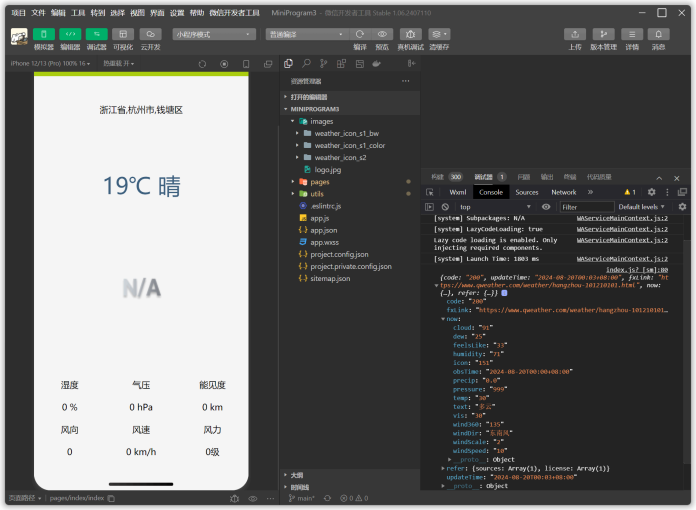
问题 1：调试终端里一直出现返回 400 的错误，天气返回为空白。

查看请求头排查问题。



发现 location 为一串字符，而根据 API 文档 可知 location 应该为查询地区的 LocationID，或以英文逗号分隔的经纬度坐标。

注意到老师给的文件中 utils 文件夹下有一个城市的 LocationID 数组和一个转换的方法，于是将这个方法引入到 index.js 中。使用 `location: util.getLocationID(that.data.region[1]),` 传入 location 参数。



修复后，可以正常获取天气信息。

也查询到，和风天气也提供了 GeoAPI，可以根据城市名获取 LocationID 或者模糊搜索城市名。这里使用老师提供的方法。

实验总结

本次实验主要是使用和风天气的 API 获取天气信息，涉及到对 wx.request 接口的掌握使用。相较于昨天的实验，今天的实验反倒更像是一个完善，真正做出一些可以用起来的小程序了！