2022年夏季《移动软件开发》实验报告

姓名：缪纬韬 学号：22020007160

姓名和学号	缪纬韬，22020007160
本实验属于哪门课程	中国海洋大学24夏《移动软件开发》
实验名称	实验2：天气查询小程序
Github仓库地址	https://github.com/spchara/remote-software-develop-lab.git

一、实验目标

1、完成天气查询小程序，学习api调用

二、实验步骤

2.1 申请api

打开网站和风天气，点击右上角的天气API，进行账号注册

随后按照操作文档的说明进行项目创建

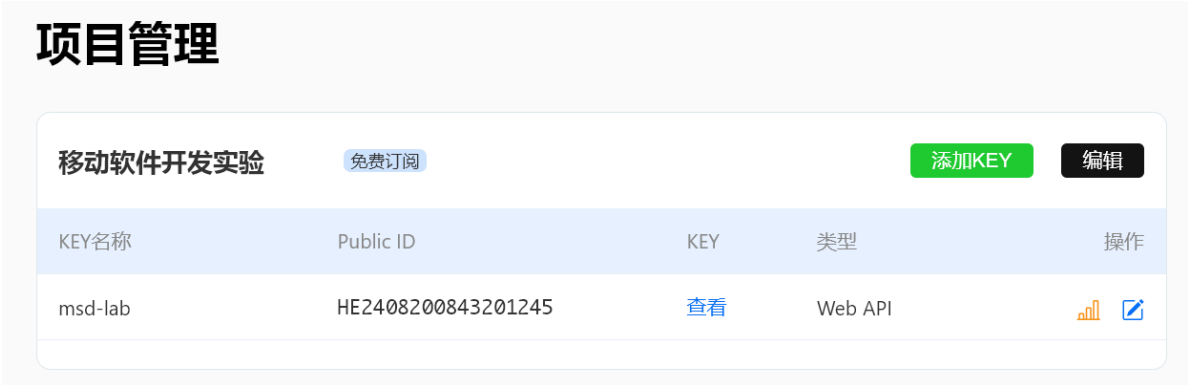
选择免费订阅、Web API

点击创建，即可申请到一个免费的api

第二步 创建项目和KEY

在控制台的左侧导航中，选择**项目管理**，并点击右上角的"创建项目"按钮。参考项目和KEY。

在创建项目时，需要同时选择一种订阅，例如：免费订阅、标准订阅或高级订阅，我们这里暂时以免费订阅为例。参考订阅。



我们可以尝试着浏览器直接调用一下

按照开发文档说明，填入必要参数：

- key
- location

在浏览器搜索框输入url `https://devapi.qweather.com/v7/weather/now?location=101120302&key=480f3b2af38e4d1dbea75b564807450c`，可以得到结果，显示正常，可以开始使用了


```
{
    "code": "200",
    "updateTime": "2024-08-20T10:36+08:00",
    "fxLink": "https://www.qweather.com/weather/zichuan-101120302.html",
    "now": {
        "obsTime": "2024-08-20T10:32+08:00",
        "temp": "26",
        "feelsLike": "30",
        "icon": "104",
        "text": "阴",
        "wind360": "135",
        "windDir": "东南风",
        "windScale": "1",
        "windSpeed": "3",
        "humidity": "92",
        "precip": "0.0",
        "pressure": "983",
        "vis": "11",
        "cloud": "99",
        "dew": "24"
    },
    "refer": {
        "sources": [
            "QWeather"
        ],
        "license": [
            "CC BY-SA 4.0"
        ]
    }
}
```



```
chara@bot:~$ curl -L -X GET --compressed 'https://devapi.qweather.com/v7/weather/now?location=101010100&key=480f3b2af38e4d1dbea75b564807450c'
{"code":"200","updateTime":"2024-08-20T08:42+08:00","fxLink":"https://www.qweather.com/weather/beijing-101010100.html","now":{"obsTime":"2024-08-20T08:40+08:00","temp":"25","feelsLike":"27","icon":"104","text":"阴","wind360":"90","windDir":"东风","windScale":"2","windSpeed":"8","humidity":"79","precip":"0.0","pressure":"1003","vis":"29","cloud":"100","dew":"21"},"refer":{"sources":["QWeather"],"license":["CC BY-SA 4.0"]}}chara@bot:~$
```


2.2 服务器域名配置

微信小程序出于安全考虑，需要对所有外部服务器的请求进行明确的声明，每一个小程序在与指定域名地址进行网络通信前都必须将该域名地址添加到管理员后台白名单中

因此我们需要登录微信开发者的后台管理网站 `https://mp.weixin.qq.com/`，并在自己的测试号下添加我们需要使用的域名 `https://devapi.qweather.com;https://geoapi.qweather.com;`

值得注意的是，和风天气的城市天气api现在不再支持用城市名字作为参数进行查询，所以我们还需要使用它提供的另一个api，`geo api`，通过GeoAPI，我们可获取到需要查询城市或POI的基本信息，包括查询地区的Location ID，我们需要使用这个id去使用查询天气。因此我这里添加了两个域名到白名单

服务器域名

本月还可修改48次 修改

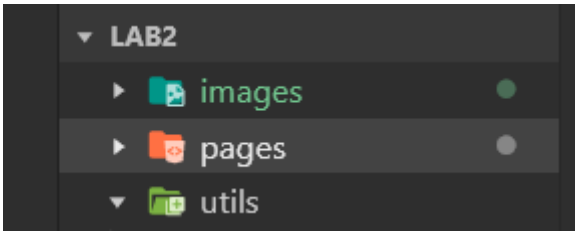
使用 微信云开发 或 微信云托管，无需配置服务器域名，省心省力。如业务访问需要安全防护，可使用 Donut 安全网关，防爬防刷防攻击，实现业务安全。点击了解更多 域名配置要求

服务器配置	域名	可配置数量
request合法域名	https://devapi.qweather.com https://geoapi.qweather.com	5个
socket合法域名	-	5个
uploadFile合法域名	-	5个

2.3 下载图标资源

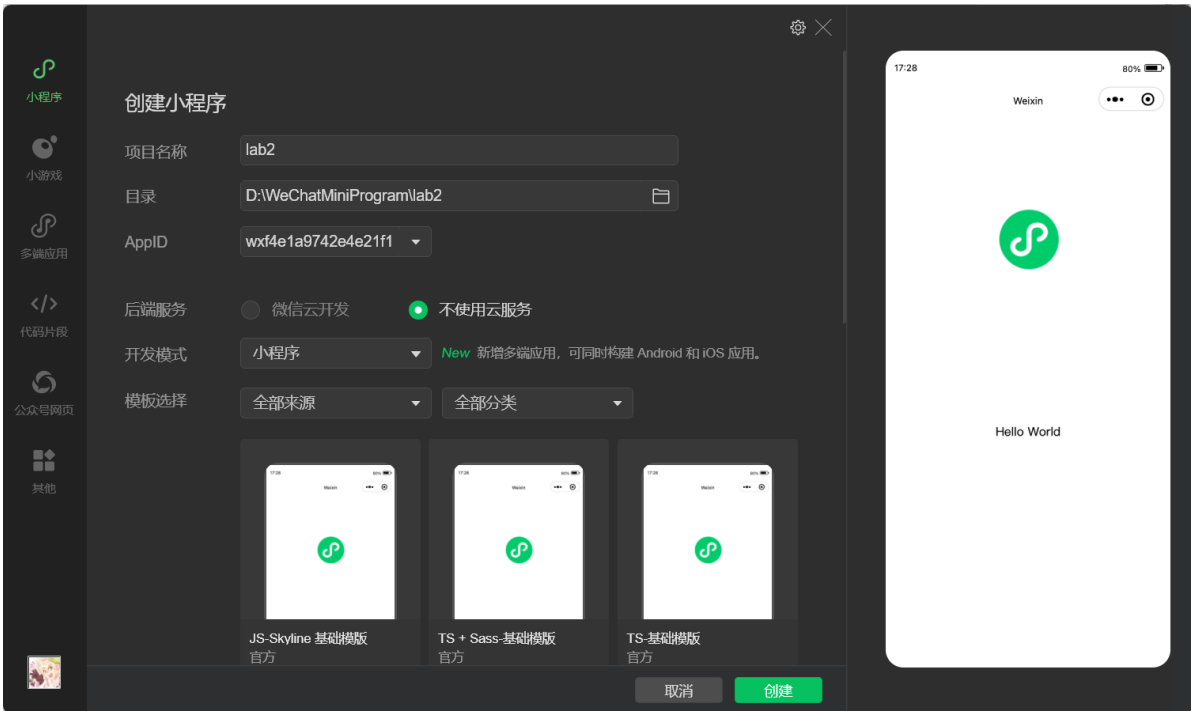
图标方面选择和风天气免费提供的图标，可以在[和风天气图标 ~ 开源、漂亮的天气图标库 (qweather.com)](qweather.com)进行下载

在项目根目录创建`/images/weather_icon`文件夹，并将下载的图标放置在内



2.4 创建项目

从模板创建，并按照上次实验的方式进行初始化



2.5 创建视图

首先修改导航栏样式

在`app.json`修改对应代码

```
  "window": {
    "navigationBarBackgroundColor": "#3883FA",
    "navigationBarTitleText": "今日天气"
    },
```

随后添加我们需要的元素：

- 地区选择框
- 温度
- 天气图标
- 详细信息

使用如下组件：

- 页面整体：<scroll-view>组件嵌套带有`class=container`属性的<view>组件
- 区域1：<picker>组件，并选择mode为region，嵌套<view>组件
- 区域2：<text>组件嵌套<view>组件
- 区域3：<image>组件
- 区域4：<view>组件，并定义`class='detail'`

- 区域4内单元行：4个<view>组件，并定义`class= 'bar'`
- 区域4内单元格：每行3个<view>组件，并定义`class='box'`

随后为它们各自添加一些css，具体代码放在第四部分

2.6 业务逻辑

需要达成的目标是当省市被改动时，小程序将自动查询当地的天气，并更新页面上的数据

为此我们首先需要做到：

- 页面数据可实时更新
- 使用api查询并处理结果
- 更换天气图标

前者可以通过js解决，本实验难点在于后者

1. 我们为picker组件绑定方法`<picker mode="region" bindchange="bindRegionChange" value="{{region}}">`，当重新选择时，就会运行`bindRegionChange()`
2. 定义data部分，这是我们需要在页面展示的变量，每次接收到api返回后，程序将修改这些值

```
data:{
    imageUrl:"",
    region:['山东省','青岛市','黄岛区'],
    regionID:"",
    updatetime : "",
    weather:{
      obsTime: "2024-08-20T08:32+08:00",
      temp: "25",
      feelsLike: "28",
      icon: "104",
      text: "阴",
      wind360: "45",
      windDir: "东北风",
      windScale: "1",
      windSpeed: "4",
      humidity: "80",
      precip: "0.0",
      pressure: "1003",
      vis: "19",
      cloud: "100",
      dew: "21"
    } ,
}
```

3. 定义两个用于api查询的函数，它们将利用picker保存地点结果查询天气

   `fetchRegion`使用`geo api`，他通过输入中文的地址，查询到对应地点的location id，用于下一步查询

   `fetchWeather`使用城市天气api，通过上一步查得的location id，也就是`region id`，用以查询对应的天气，然后修改`data`中的`weather`

   最后在`bindRegionChange`中调用它们，即可完成

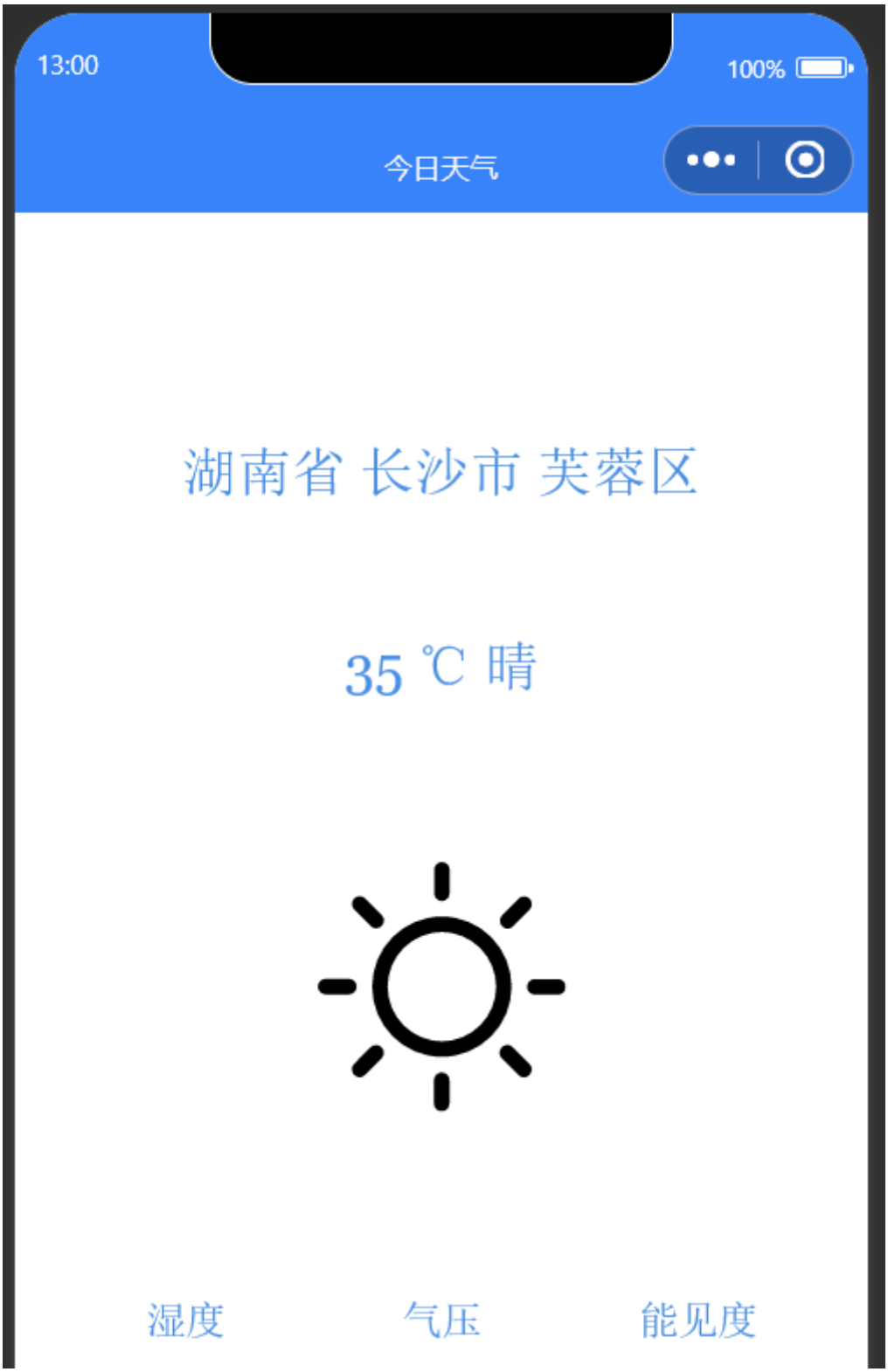
```
 bindRegionChange: function(e) {
    this.setData({
      region: e.detail.value
```

```
    }, () => {
      console.log(this.data.region);
      this.fetchRegion();
    });
  },


  fetchRegion: function() {
    wx.request({
      url: 'https://geoapi.qweather.com/v2/city/lookup',
      data: {
        location: this.data.region[2],
        adm: this.data.region[1],
        key: "api key省略",
      },
      success: (res) => {
        console.log(res.data.location[0].id);
        if (res.statusCode === 200 && res.data.location &&
res.data.location.length > 0) {
          this.setData({
            regionID: res.data.location[0].id
          }, () => {
            this.fetchWeather();
          });
        }
      }
    });
  },


  fetchWeather: function(){
    wx.request({
      url: 'https://devapi.qweather.com/v7/weather/now',
      data:{
        location:this.data.regionID,
        key:"api key省略"
      },
      success: (res) =>{
        if (res.statusCode === 200) {
          console.log(res);
          this.setData({
            weather:res.data.now
          });
          this.setImageUrl();
        }
      }
    });

  },
```

4. 用字符串拼接出图片路径达到更换图片的效果

```
  setImageUrl: function() {
    const baseImgUrl = "/images/weather_icon/";
    const imgName = this.data.weather.icon + ".svg";
    const fullUrl = baseImgUrl + imgName;
    console.log(fullUrl);
    this.setData({
      imageUrl: fullUrl
    });
  },
```

三、程序运行结果

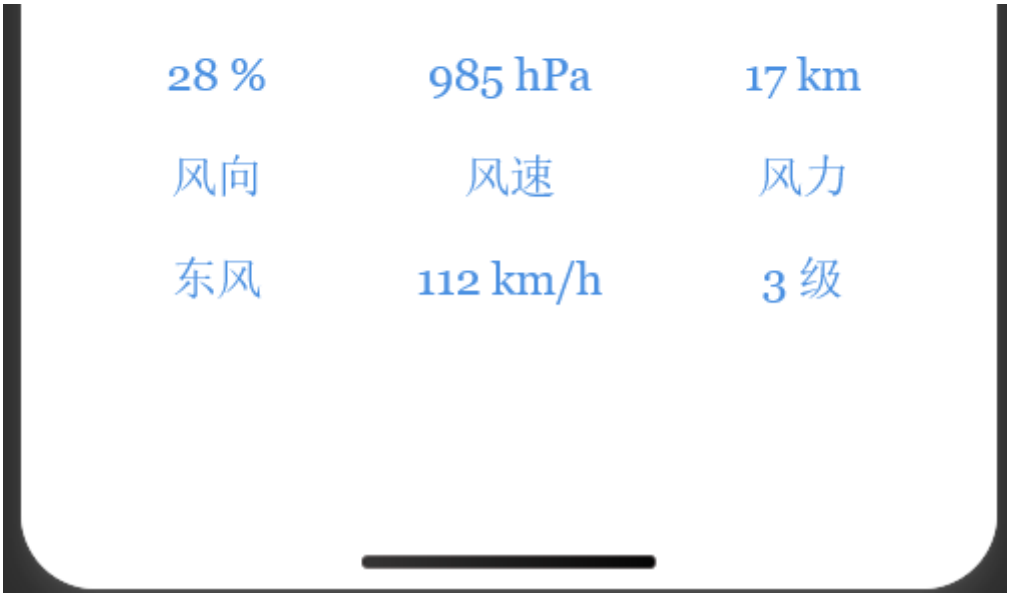


35 %	1001 hPa	16 km
风向	风速	风力
西南风	17 km/h	2 级

吉林省 长春市 南关区

28 ℃ 阴

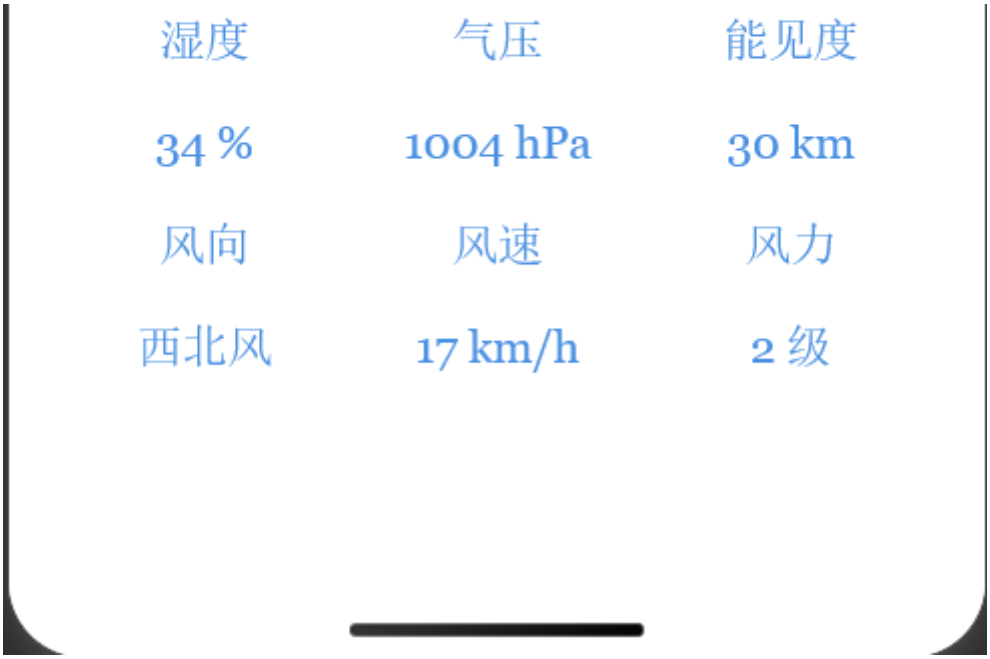
湿度	气压	能见度


28 %
985 hPa
17 km
风向
风速
风力
东风
112 km/h
3 级


13:01
100%
今日天气
山东省 青岛市 黄岛区
26 ℃ 中雨
取消
确定

上海市 上海市 长宁区

34 ℃ 阴



四、主要代码

index.wxml

```
<!--index.wxml-->
<scroll-view class="scrollarea" scroll-y type="list">
  <view class="container">
    <picker mode="region" bindchange="bindRegionChange" value="{{region}}">
      <view class="region">
        {{region[0]}}  {{region[1]}}  {{region[2]}}
      </view>
    </picker>
    <view class="tempInfo">
      <text class="temp">{{weather.temp}} ℃ {{weather.text}}</text>
    </view>
    <image src="{{imageUrl}}" class="icon"></image>

    <view class="detail">
      <view class="bar">
        <view class="box">湿度</view>
        <view class="box">气压</view>
        <view class="box">能见度</view>
      </view>
      <view class="bar">
        <view class="box">{{weather.temp}} %</view>
        <view class="box">{{weather.pressure}} hPa</view>
        <view class="box">{{weather.vis}} km</view>
      </view>
      <view class="bar">
        <view class="box">风向</view>
        <view class="box">风速</view>
        <view class="box">风力</view>
      </view>
      <view class="bar">
        <view class="box">{{weather.windDir}}</view>
        <view class="box">1{{weather.windSpeed}} km/h</view>
```

```
        <view class="box">{{weather.windScale}} 级</view>
      </view>

    </view>
  </view>
</scroll-view>

```

index.js

```
// index.js

Page({
  data:{
    imageUrl:"",
    region:['山东省','青岛市','黄岛区'],
    regionID:"",
    updatetime : "",
    weather:{
      obsTime: "2024-08-20T08:32+08:00",
      temp: "25",
      feelsLike: "28",
      icon: "104",
      text: "阴",
      wind360: "45",
      windDir: "东北风",
      windScale: "1",
      windSpeed: "4",
      humidity: "80",
      precip: "0.0",
      pressure: "1003",
      vis: "19",
      cloud: "100",
      dew: "21"
    } ,

  },

  onLoad: function(options) {
    this.setImageUrl();
    this.fetchRegion();
  },

  setImageUrl: function() {
    const baseImgUrl = "/images/weather_icon/";
    const imgName = this.data.weather.icon + ".svg";
    const fullUrl = baseImgUrl + imgName;
    console.log(fullUrl);
    this.setData({
      imageUrl: fullUrl
    });
  },

  bindRegionChange: function(e) {
    this.setData({
      region: e.detail.value
    }, () => {
```

```
      console.log(this.data.region);
      this.fetchRegion();
    });
  },


  fetchRegion: function() {
    wx.request({
      url: 'https://geoapi.qweather.com/v2/city/lookup',
      data: {
        location: this.data.region[2],
        adm: this.data.region[1],
        key: "480f3b2af38e4d1dbea75b564807450c",
      },
      success: (res) => {
        console.log(res.data.location[0].id);
        if (res.statusCode === 200 && res.data.location &&
res.data.location.length > 0) {
          this.setData({
            regionID: res.data.location[0].id
          }, () => {
            this.fetchWeather();
          });
        }
      }
    });
  },


  fetchWeather: function(){
    wx.request({
      url: 'https://devapi.qweather.com/v7/weather/now',
      data:{
        location:this.data.regionID,
        key:"480f3b2af38e4d1dbea75b564807450c"
      },
      success: (res) =>{
        if (res.statusCode === 200) {
          console.log(res);
          this.setData({
            weather:res.data.now
          });
          this.setImageUrl();
        }
      }
    });

  },
  }
)
```

index.wxss

```
/**index.wxss**/
page {
  height: 100vh;
  display: flex;
```

```
  flex-direction: column;
}
.scrollarea {
  flex: 1;
  overflow-y: hidden;
}

.icon{
  width: 220rpx;
  height: 220rpx;
  object-fit: cover;
  border-radius: 20px;
  margin: 30rpx;
}

.detail{
  width: 90%;
  display: flex;
  flex-direction: column;
}

.bar{
  display: flex;
  flex-direction: row;
  margin: 20rpx 0;
}

.box{
  width: 33.3%;
  text-align: center;
  font-size: 18px;
  font-family: "Georgia", serif;
  color: #4A90E2;
}

.region{
  text-align: center;
  font-size: 25px;
  font-family: "Georgia", serif;
  color: #4A90E2;
}

.temp{
  text-align: center;
  font-size: 25px;
  font-family: "Georgia", serif;
  color: #4A90E2;
}
```

五、问题总结与体会

- **Q：和风天气的api无法直接用地名作为参数查询**

  A：现在和风天气提供了另一个geo api，专门用于地名查询，先用geoapi查询地名得到对应的location id，再用id去查询天气。除了id，它还可以查询城市或POI的基本信息，包括查询地区的Location ID，多语言名称、经纬度、时区、海拔、Rank值、归属上级行政区域、所在行政区域等。除此之外它还可以

    - 避免重名城市的困扰
    - 支持名称模糊搜索
    - 在你的应用或网站中根据用户输入的名称返回多个城市结果，便于用户选择准确的城市并返回该城市天气
    - 在你的应用或网站中展示热门城市
    - 不需要维护城市列表，城市信息更新实时获取
- **Q：即使将api域名添加到了request白名单，还是显示访问受限**

  A：通过重启项目解决

六、总结

第二次实验总的来说还算顺利，虽然我个人因为一些比较低级的问题，比如变量前面忘记了this，导致没法正确指向数据，检查了半天才发现的这种错误浪费了不少时间。但总归困难都通过查阅资料解决了。希望在接下来的课程中，能够更好地掌握这门技术！