武汉纺织大学

Web应用开发课程设计

**Javaweb天气预报系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11804**

**姓 名： 童鑫**

**学 号： 1804280412**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月18日**

目录

**1.需求分析1**

1.1课程设计的目的1

1.2功能分析1

1.3原理和思路1

**2系统分析1**

2.1可行性分析1

2.2功能性分析1

2.3采用的软件和设备1

**3.系统实现2**

3.1项目结构2

3.2 jsonp3

3.3 getweather()3

3.4 changeImg()5

3.5 createUrl()6

**4.系统测试6**

**5.实验总结7**

# 1 需求分析

## 1.1课程设计的目的

**熟练的运用web前后端所学的知识制作一个功能完整的，界面优美的小型的系统。并且要求能够查出未来几天某个城市的天气情况的详细信息，并展示到页面上。**

## 1.2功能分析

**根据老师所给的链接和参考图片，通过搜索框查询或选择某个城市，显示该城市的天气情况和和一些别的参数如最高气温、最低气温、风向等等，而天气信息可以使用网上提供的接口去调用来获得。**

## 1.3原理和思路

**在思考该题目时，最主要的就是数据的封装和解析，通过中国天气网的数据解析，可以看出，是通过查询接口查询到数据之后，通过jsonp传递到前端页面上。所以我在实现该功能的时候，通过找到网上的免费的天气查询接口，然后按照相应的格式封装后，依据你选择的城市去查询该城市的，并根据这个调用天气预报查询接口进行查询，并将结果返回形成相应的结果数据，**

# 2 系统分析

## 2.1 可行性分析

在软件开发前就进入到进行可行性的研究阶段。并不是所有问题都有具有简单的解决办法，许多问题不能在预定的规模之内解决。因此通过可行性的研究分析可以得知问题所在。从而比避免人力的浪费。系统的目标为天气预报，可以分阶段依次完成。

## 2.2技术可行性

根据设想的系统功能及实现系统的各项约束条件，根据新系统目标来衡量所需要的技术是否具备，本系统是一个天气查询，现有的技术和条件都以成熟，能利用现有技术完全达到改系统的功能目标但由于自身的能力和知识的缺陷可能使得该系统存在不完善之处。

**2.3 采用的软件设备**

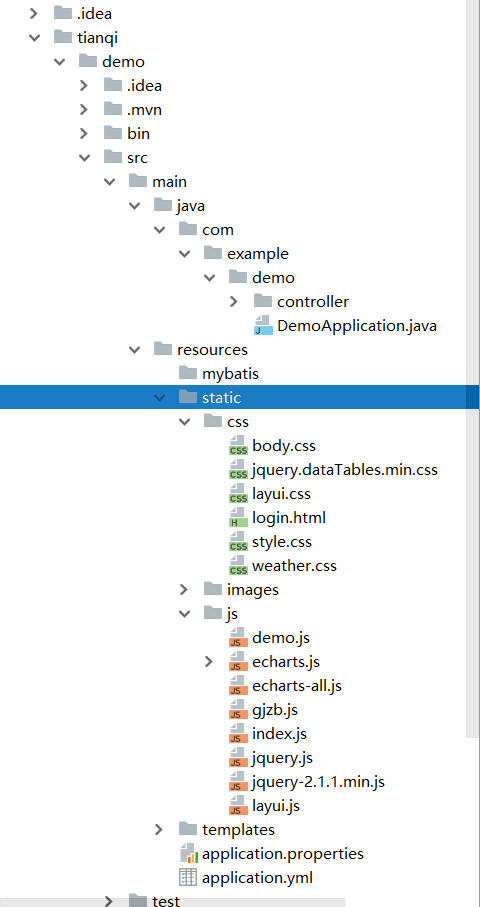
（1）操作系统：Windows 10

（2）软件环境 IEDA

（3）硬件环境 电脑一台

# 3 系统实现

## 3.1 项目结构



## 3.2 jsonp

**调用jsonp函数请求当前所在城市**

document.write("<script language=javascript src='/jquery.js'></script>");

jsonp('https://api.map.baidu.com/api?v=2.0&ak=p216eDob1pGzTFpzf5k4xYjX6EyjuiKA&s=1&callback=getCity');

window.onload = **function** () {

btn.addEventListener('click',**function** () {

jsonp(createUrl()[0]);

jsonp(createUrl()[1]);

});

text.addEventListener('keydown', **function** (e){

if (e.keyCode == 13) {

jsonp(createUrl()[0]);

jsonp(createUrl()[1]);

}

});

}

**3.3 getweather（）**

**数据请求成功回调函数，用于将获取到的数据放入页面相应位置**

**function** getWeather(response) {

**var** oSpan = document.getElementsByClassName('info');

**var** dom = document.getElementById("container");

**var** data = response.result;

oSpan[0].innerHTML = data[0].citynm;

oSpan[1].innerHTML = data[0].days;

oSpan[2].innerHTML = data[0].week;

oSpan[3].innerHTML = data[0].weather;

oSpan[4].innerHTML = data[0].temperature;

oSpan[5].innerHTML = data[0].winp;

oSpan[6].innerHTML = data[0].wind;

**var** aDiv = document.getElementsByClassName('future\_box');

for (**var** i = 0; i < aDiv.length; i++) {

**var** aSpan = aDiv[i].getElementsByClassName('future\_info');

aSpan[0].innerHTML = data[i + 1].days;

aSpan[1].innerHTML = data[i + 1].week;

aSpan[2].innerHTML = data[i + 1].weather;

aSpan[3].innerHTML = data[i + 1].temperature;

**var** myChart = echarts.init(dom);

**var** app = {};

option = null;

option = {

title: {

text: '天气折线图'

},

tooltip: {

trigger: 'axis'

},

legend: {

data: ['早', '晚']

},

grid: {

left: '3%',

right: '4%',

bottom: '3%',

containLabel: true

},

toolbox: {

feature: {

saveAsImage: {}

}

},

xAxis: {

type: 'category',

boundaryGap: false,

data: [data[0].week, data[1].week, data[2].week, data[3].week, data[4].week, data[5].week]

},

yAxis: {

type: 'value'

},

series: [

{

name: '最高',

type: 'line',

stack: '温度',

data: [(data[0].temperature).split("℃/")[0], (data[1].temperature).split("℃/")[0], (data[2].temperature).split("℃/")[0], (data[3].temperature).split("℃/")[0], (data[4].temperature).split("℃/")[0], (data[5].temperature).split("℃/")[0]]

},

{

name: '最低',

type: 'line',

stack: '温度',

data: [(data[0].temperature.split("/")[1]).split("℃")[0], (data[1].temperature.split("/")[1]).split("℃")[0], (data[2].temperature.split("/")[1]).split("℃")[0], (data[3].temperature.split("/")[1]).split("℃")[0],(data[4].temperature.split("/")[1]).split("℃")[0], (data[5].temperature.split("/")[1]).split("℃")[0]]

}

]

};

;

if (option && typeof option === "object") {

myChart.setOption(option, true);

}

}

//根据返回数据，替换不同天气图片

changeImg(response);

}

**3.4 changeImg()**

**根据获取到的数据更改页面中相应的图片**

**function** changeImg(data) {

**var** firstImg = document.getElementsByTagName("img")[0];

**var** firstWeatherId = data.result[0].weatid;

chooseImg(firstWeatherId, firstImg);

**var** aImg = document.getElementById('future\_container').getElementsByTagName('img');

for (**var** j = 0; j < aImg.length; j++) {

**var** weatherId = data.result[j + 1].weatid;

chooseImg(weatherId, aImg[j]);

}

}

**3.5 createUrl()**

**根据城市名创建请求数据及url**

**function** createUrl() {

**var** cityName = '';

if (arguments.length == 0) {

cityName = document.getElementById('text').value;

} else {

cityName = arguments[0];

}

**var** urls = [];

urls[0] = 'https://sapi.k780.com/?app=weather.future&appkey=10003&sign=b59bc3ef6191eb9f747dd4e83c99f2a4&format=json&jsoncallback=getWeather&weaid=' + encodeURI(cityName);

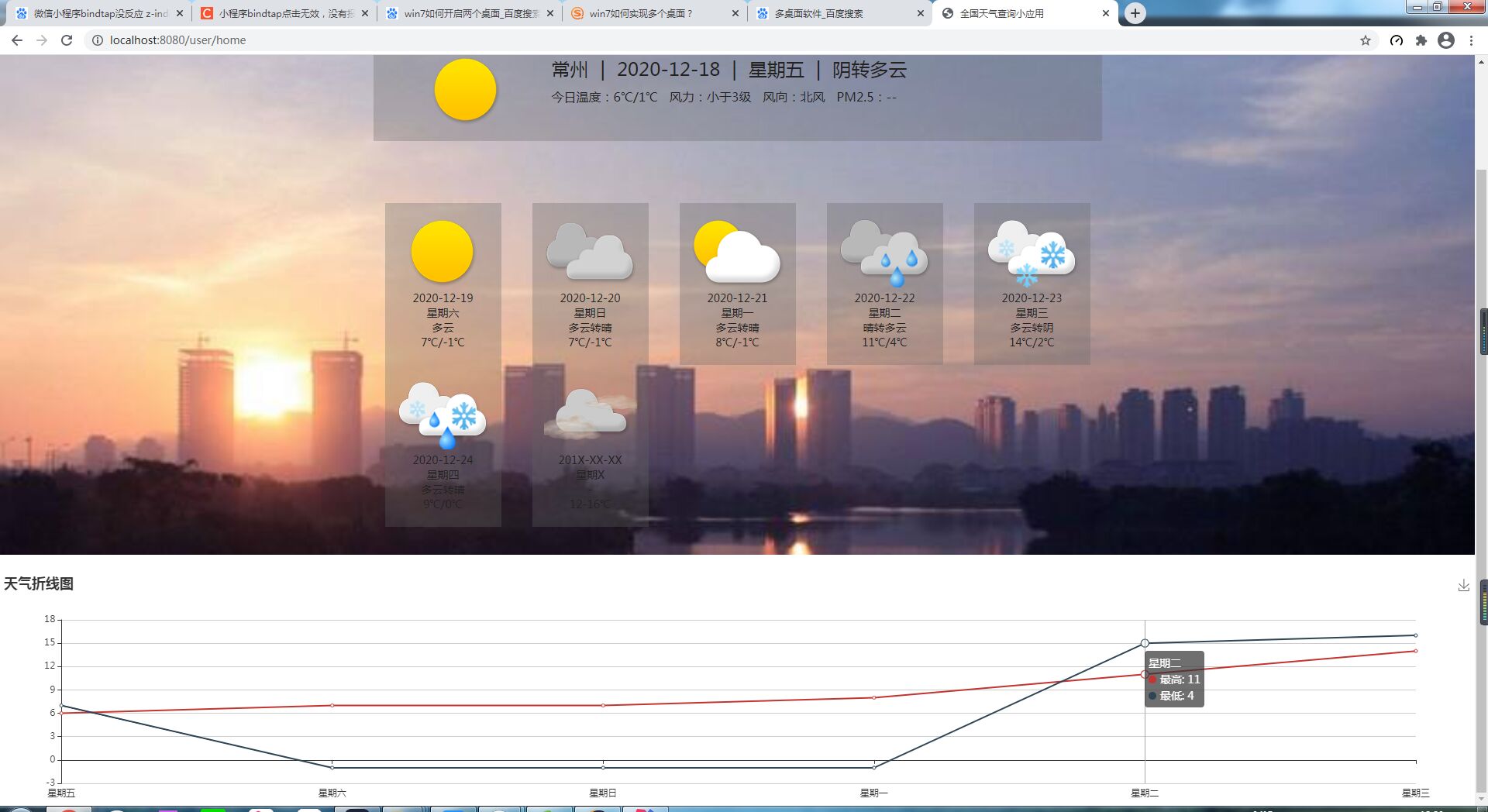
urls[1] = 'https://api.map.baidu.com/telematics/v3/weather?output=json&ak=FK9mkfdQsloEngodbFl4FeY3&callback=getTodayWeather&location=' + encodeURI(cityName);

return urls;

}

# 4 系统测试

运行成功



# 5 系统总结

**经过此次课程设计，将前后端完整的做了一遍，从界面设计到后端环境的搭建以及代码的书写。感觉自己对于上学期学习的java知识更加的熟悉，用到了经典的mvc设计思想。虽然系统的基本功能已经实现，但还是存在系统不稳定、数据解析太慢，查询响应时间太长等一些缺点。这些问题虽然对于目前这种小型的系统影响并不是很大，但是当访问量和查询过大的时候，这些问题所带来的影响可能会被无限的放大。这也是作为一个程序设计者必须要考虑到的因素。总之，对于本系统的实现，也感觉到了很多的不足，对于原生的js代码比较陌生，对于接口的调用程序的介绍不习惯于去查看官方的API文档等等。但是总体来说，进一步提高了自己的编码规范和水平，不管是在以后的学习还是工作中都应该不断的学习，将课本知识与实践结合起来，才能更好的理解。**