

微信小程序开发文档

1801210680

吴珏

写在前面：

本文参考了刘玄同学的文档，使用了百度地图 API 用于定位，使用了和风天气 API 用于获取天气信息。在他工作的基础上，增加了以下内容：7 天天气概况、今日天气的逐小时预报（绘制动态曲线）、今日天气的生活指数（借鉴了王鲁月同学的想法）。

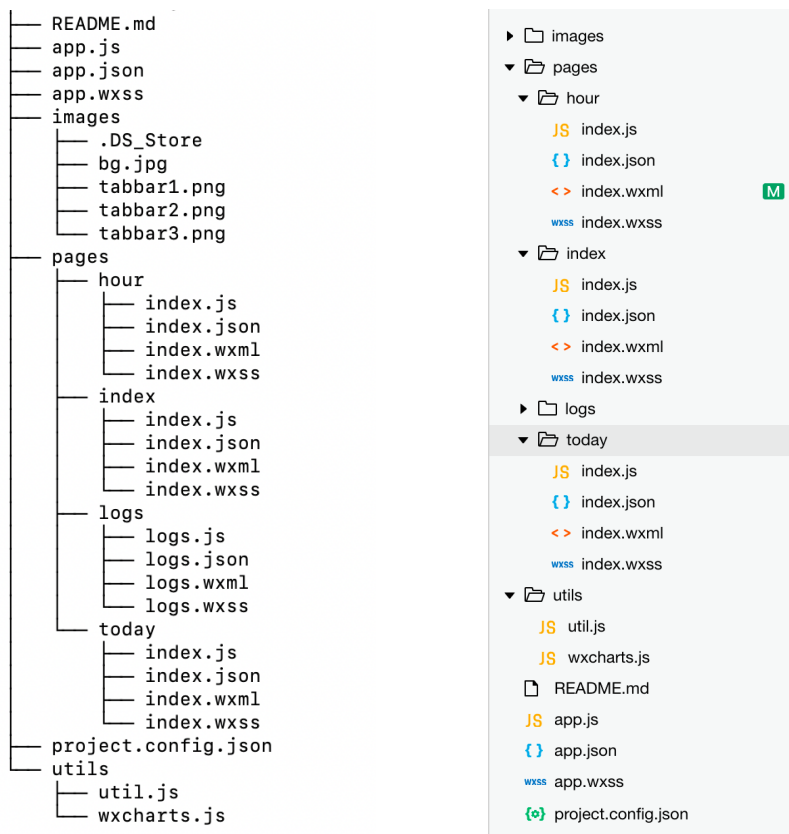
结果展示：



实验内容：

1. 文件目录：

images 里放置使用的图片资源，pages 是小程序的页面，目录中的三个页面通过 tabbar 进行切换，pages/index 目录是近日天气概况的源码，pages/hour 目录是逐小时天气预报的源码，pages/today 是今日生活指数的源码。



在 app.json 文件中对 pages 和 tabbar 进行配置，代码如下：

```
{  
  "pages": [  
    "pages/index/index",  
    "pages/logs/logs",  
    "pages/today/index",  
    "pages/hour/index"  
  ],  
  "window": {  
    "backgroundTextStyle": "light",  
    "navigationBarBackgroundColor": "#fff",  
    "navigationBarTitleText": "WeChat",  
    "navigationBarTextStyle": "black"  
  },  
  "tabBar": {  
    "color": "#a0a0a0",  
    "selectedColor": "#FFFFFF",  
    "backgroundColor": "#FDEEF4",
```

```
"borderStyle": "white",
"list": [
  {
    "pagePath": "pages/index/index",
    "iconPath": "images/tabbar1.png",
    "selectedIconPath": "images/tabbar1.png",
    "text": "近日天气概况"
  },
  {
    "pagePath": "pages/hour/index",
    "iconPath": "images/tabbar3.png",
    "selectedIconPath": "images/tabbar3.png",
    "text": "逐小时天气预报"
  },
  {
    "pagePath": "pages/today/index",
    "iconPath": "images/tabbar2.png",
    "selectedIconPath": "images/tabbar2.png",
    "text": "今日生活指数"
  }
]
}
```

2. 近日天气概况（pages/index 目录）：

如果想要获取未来七天天气和逐小时的天气预报，需要在和风天气网页（<https://www.heweather.com>）进行个人开发者的认证，认证通过之后，就可以调用 API 了。



在主页点击注册按钮，选择个人开发者进行注册，按照网页提供的信息一步一步进行就可以了。注册完需要审核一段时间，审核通过之后，点击主页上的控制台按钮，可以看到账号类型为认证开发者了。



显示 7 天天气的方法与显示 3 天天气的类似（可参考刘玄同学文档），这里不再赘述，仅对补充内容进行说明。由于调用和风天气 api 得到的 json 数据中，只有日期信息，并无星期几的信息。这里使用了微信封装好的计算日期的工具。

在需要计算日期的页面引入 util.js, 引入的代码如下：

```
var util = require('../utils/util.js');
```

执行代码，可以看到 utils 目录下自动生成了 util.js 文件，util.js 的代码如下：

```
const formatTime = date => {  
  const year = date.getFullYear()
```

```

const month = date.getMonth() + 1
const day = date.getDate()
const hour = date.getHours()
const minute = date.getMinutes()
const second = date.getSeconds()

return [year, month, day].map(formatNumber).join('/') + ' ' + [hour, minute,
second].map(formatNumber).join(':')
}

const formatNumber = n => {
  n = n.toString()
  return n[1] ? n : '0' + n
}

const formatDate = date => {
  const year = date.getFullYear()
  const month = date.getMonth() + 1
  const day = date.getDate()
  return [year, month, day].map(formatNumber).join('-')
}

```

在文件中，加入函数 `getDates()`和 `dateLater()`：用于获取当前时间，并计算当前时间往后推的日期，计算星期几。代码如下：

```

function getDates(days, todate) {
  var dateArray = [];
  for (var i = 0; i < days; i++) {
    var dateObj = dateLater(todate, i);
    dateArray.push(dateObj)
  }
  return dateArray;
}

function dateLater(dates, later) {

```

```

let dateObj = {};
let show_day = new Array('周日', '周一', '周二', '周三', '周四', '周五', '周六');
let date = new Date(dates);
date.setDate(date.getDate() + later);
let day = date.getDay();
let yearDate = date.getFullYear();
let month = ((date.getMonth() + 1) < 10 ? ("0" + (date.getMonth() + 1)) :
date.getMonth() + 1);
let dayFormate = (date.getDate() < 10 ? ("0" + date.getDate()) : date.getDate());
dateObj.time = yearDate + '-' + month + '-' + dayFormate;
dateObj.week = show_day[day];
return dateObj;
}

```

需要注意的是：所有的函数都需要 exports 之后才生效，在 util.js 文件里写入：

```

module.exports = {
  formatDate: formatDate,
  getDates: getDates
}

```

在 index.js 文件中，修改函数 getWeather，通过 formatDate 函数获取当前日期，再使用 getDates 函数计算出周几，并且在保存七天天气信息的 daily_forecast 对象中加入属性 dayofweek：

```

let time = util.formatDate(new Date());
let date = util.getDates(7, time);
for(var i=0;i<7;i++){
  daily_forecast[i].dayofweek = date[i].week;
};

```

3. 逐小时天气预报（pages/hour 目录）：

首先在 index.wxml 和 index.wxss 中，对页面前端进行设计。页面中展示了一个动态的折线图和每个时段天气的具体信息。其中，绘制折线图需要使用组件 canvas。

在 index.wxml 代码中加入以下内容：

```

<view class="chart" style='width:100%;text-align:center;'>
  <canvas canvas-id="lineCanvas" disable-scroll="true" class="canvas"
bindtouchstart="touchHandler"></canvas>
</view>

```

在 index.wxss 中加入：

```

.canvas{
  width:600px;
  height: 300px;
  transform: scale(0.9);
}
.chart{
  margin:0 auto;
  display: flex;
  width: 100%;
  height: 300px;
  justify-content: center;
  padding-bottom: 60rpx;
}

```

逐小时天气的文字信息内容的展示与七天天气信息的显示类似，在 index.wxml 中加入以下代码：

```

<view wx:for="{{hour}}" wx:for-index="i" wx:for-item="item">
  <view class="hor forecast">
    <view class="time">
      <view class="center">{{item.time}}</view>
    </view>
    <view class="tmp">
      <view class="center">
        <text
decode="{{true}}">{{item.tmp}}°&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;{{item.cond_txt}}&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;
;降水:{{item.pop}}%</text>
      </view>
    </view>
  </view>

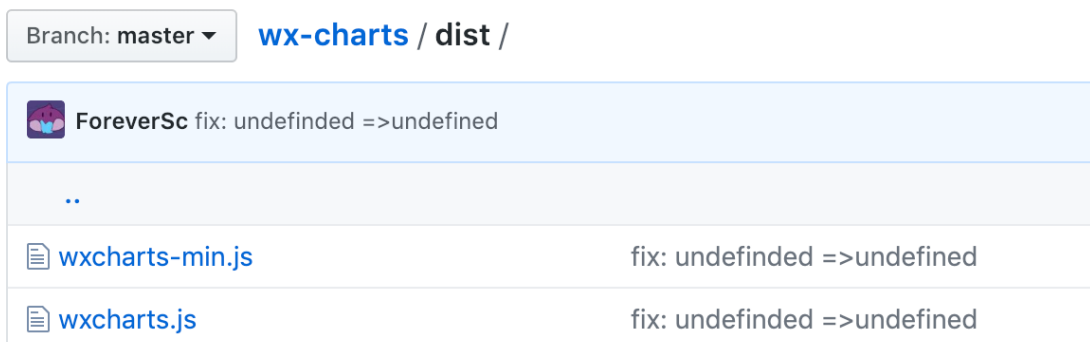
```

```
<view>
  <view class="center">{{item.wind_dir}}:{{item.wind_spd}}级</view>
</view>
</view>
</view>
```

在 index.wxss 中加入:

```
.forecast{
padding: 15px 0px 0px 0px;
padding-left: 30rpx;
padding-right: 30rpx;
border-bottom:0.5rpx solid rgb(241,229,229);
color:white;
flex-direction: row;
justify-content:space-between;
font-size: 32rpx;
}
```

由于微信小程序本身框架的限制，很难集成目前已有的图表工具，这里我们使用微信小程序图表插件(wx-charts)。它是基于 canvas 绘制的，体积小巧，支持多种图表类型，包括：饼图、线图、柱状图等，需要自行下载 wxcharts.js（下载地址：<https://github.com/xiaolin3303/wx-charts.git>）。



将文件 wx-charts/dist/wxcharts.js 拷贝到 utils 目录下，就可以可使用 wx-charts 插件了。



在 index.js 中导入 wxcharts.js，并定义需要绘制的图：

```
var chart = null;  
var wxCharts = require('../utils/wxcharts.js');
```

通过 getWeather 函数，获取到每个时段的天气信息：

```
getWeather: function (city) {  
    var that = this;  
    var url = "https://free-api.heweather.com/s6/weather";  
    var parameters = {  
        location: city,  
        key: "f5a5b6e1918e449e8e78162450cbb813"  
    }  
    wx.request({  
        url: url,  
        data: parameters,  
        success: function (res) {  
            var hour = res.data.HeWeather6[0].hourly;  
            var tmp = res.data.HeWeather6[0].now.tmp;  
            console.log(hour);  
            console.log(hour[0].time);  
            for(var i=0;i<8;i++){  
                var n = hour[i].time.split(" ");  
                hour[i].time = n[1];  
            }  
            that.setData({  
                hour:hour,  
                tmp: tmp,  
            })  
            that.getChart(hour);  
        },  
        fail: function (res) { },  
        complete: function (res) { },  
    })  
}
```

通过 `getChart` 函数绘制表格，值得注意的是：`wxCharts` 插件的自适应能力较差，需要获取当前设备的屏幕数据进行配置。`wxCharts` 的具体参数说明，可以参考：<https://github.com/xiaolin3303/wx-charts/issues/56>。

```
getChart:function(hour){
    var windowWidth = "", windowHeight = "";    //定义宽高
    try {
        var res = wx.getSystemInfoSync();    //获取屏幕宽高数据
        windowWidth = res.windowWidth;
        windowHeight = res.windowWidth;
    } catch (e) {
        console.error('getSystemInfoSync failed!');
    }
    console.log(hour);
    chart = new wxCharts({
        canvasId: 'lineCanvas',
        type: 'line',
        categories: [hour[0].time, hour[1].time, hour[2].time, hour[3].time,
hour[4].time, hour[5].time, hour[6].time, hour[7].time],
        series: [{
            name: '温度',
            fontColor: "#FFFFFF",
            data: [hour[0].tmp, hour[1].tmp, hour[2].tmp, hour[3].tmp, hour[4].tmp,
hour[5].tmp, hour[6].tmp, hour[7].tmp],
            format: function (val) {
                return val + "°";
            }
        }],
        xAxis:{
            title: "时间",
            fontColor: "#FFFFFF",
            titleFontColor: "#FFFFFF",
        },
        yAxis: {
```

```

        fontColor: "#FFFFFF",
        titleFontColor: "#FFFFFF",
        format: function (val) {
            return val.toFixed(0);
        }
    },
    width: windowWidth,
    height: 300,
    dataLabel:false,
    dataPointShape:true,
    extra: {
        legendTextColor: "#FFFFFF"
    }
});
}

```

4. 今日生活指数（pages/today 目录）：

通过天气 api 获取数据后，将数据展示到前端页面中，具体的方法与前面类似。获取天气数据的函数代码如下：

```

getWeather: function (city) {
    var that = this;
    var url = "https://free-api.heweather.com/s6/weather";
    var parameters = {
        location: city,
        key: "f5a5b6e1918e449e8e78162450cbb813"
    }
    wx.request({
        url: url,
        data: parameters,
        success: function (res) {
            var lifestyle = res.data.HeWeather6[0].lifestyle;
            var comf = res.data.HeWeather6[0].lifestyle[0].brf;
            var comf_txt = res.data.HeWeather6[0].lifestyle[0].txt;

```

```
var drsg = res.data.HeWeather6[0].lifestyle[1].brf;
var drsg_txt = res.data.HeWeather6[0].lifestyle[1].txt;
var flu = res.data.HeWeather6[0].lifestyle[2].brf;
var flu_txt = res.data.HeWeather6[0].lifestyle[2].txt;
var sport = res.data.HeWeather6[0].lifestyle[3].brf;
var sport_txt = res.data.HeWeather6[0].lifestyle[3].txt;
var uv = res.data.HeWeather6[0].lifestyle[5].brf;
var uv_txt = res.data.HeWeather6[0].lifestyle[5].txt;
var cw = res.data.HeWeather6[0].lifestyle[6].brf;
var cw_txt = res.data.HeWeather6[0].lifestyle[6].txt;
var air = res.data.HeWeather6[0].lifestyle[7].brf;
var air_txt = res.data.HeWeather6[0].lifestyle[7].txt;
console.log(lifestyle);
that.setData({
  comf:comf,
  comf_txt: comf_txt,
  drsg: drsg,
  drsg_txt: drsg_txt,
  uv:uv,
  uv_txt:uv_txt,
  cw:cw,
  cw_txt:cw_txt,
  air:air,
  air_txt:air_txt,
  sport: sport,
  sport_txt: sport_txt,
  flu:flu,
  flu_txt:flu_txt,
})
},
fail: function (res) { },
complete: function (res) { },
})
```

```
wx.request({  
  
  })  
}
```

写在最后：

本文的所有代码：

<https://github.com/imwujue/weixin-miniprogram-WuJueWeather>

欢迎扫码：

